Learning Nested Differences in the Presence of Malicious Noise

Peter Auer

Graz University of Technology, Klosterwiesgasse 32/2, A-8010 Graz (Austria)

pauer@igi.tu-graz.ac.at

July 6, 1996

Abstract

We present a PAC-learning algorithm and an on-line learning algorithm for nested differences of intersection-closed classes. Examples of intersection-closed classes include axis-parallel rectangles, monomials, and linear sub-spaces. Our PAC-learning algorithm uses a pruning technique that we rigorously proof correct. As a result we show that the tolerable noise rate for this algorithm does not depend on the complexity (VC-dimension) of the target class but only on the VC-dimension of the underlying intersection-closed class. For our on-line algorithm we show an optimal mistake bound in the sense that there are concept classes for which each on-line learning algorithm (using nested differences as hypotheses) can be forced to make at least that many mistakes.

1 Introduction and preliminaries

We are interested in the implications of noise when learning nested differences of intersection-closed classes. For the noise-free case the learnability of nested differences was analyzed by Helmbold, Sloan, and Warmuth [7]. The main focus of our work is the tolerable amount of noise such that learning is still possible. The learning models we will consider are the PAC-learning model with malicious noise [12, 8] and the on-line learning model [1, 10] with noise. In both learning models the learner has to discover some fixed target concept $C \subseteq X$ over the domain X, where it is only known that $C \in \mathcal{C}$ for some given concept class $\mathcal{C}$ of subsets of X. We will not distinguish between a concept C and the corresponding function $C(x) = \begin{cases} + & \text{if} \quad x \in C \\ - & \text{if} \quad x \notin C \end{cases}$.

1.1 Learning models

In the original PAC-learning model of Valiant [12] the learner receives a sample $(x_1, C(x_1)), \ldots, (x_m, C(x_m))$ labeled by the target concept C where the x_i are independently

ISSN 1433-8092

drawn from a probability distribution $\mathcal{D}$ on X. The size m of the sample can be chosen by the learner depending on the required precision ϵ and confidence δ. The learner successfully learns C if with high probability (measured by the confidence parameter δ) a random sample is draw such that based on this random sample the learner produces a hypothesis H which is ϵ-close to C, i.e. $\mathcal{D}\{x : H(x) \neq C(x)\} < \epsilon$, where ϵ is the precision parameter. In the malicious PAC model of Kearns and Li [8] a certain fraction (measured by the noise rate η) of the examples is noisy. Formally, for each example $(x_i, C(x_i))$ of the sample an independent Bernoulli experiment with success probability η determines if the example is affected by noise. On failure the original example $(x_i, C(x_i))$ is passed to the learner, on success an arbitrary example (x'_i, l'_i) chosen by an adversary is passed to the learner. As in the original PAC model, with high probability the learner has to produce a hypothesis H which is ϵ-close to the target concept C in respect to the original distribution $\mathcal{D}$.

Definition 1.1 *Let $\mathcal{C}$ be a concept class over domain X. Algorithm A (ϵ, δ)-learns $\mathcal{C}$ with malicious noise rate η if there is an $m(\epsilon, \delta, \eta)$ such that the following condition is fulfilled: for any concept $C \in \mathcal{C}$ and for any probability distribution $\mathcal{D}$ on X, the probability that a sample of size $m(\epsilon, \delta, \eta)$ is given to algorithm A such that the algorithm's hypothesis H is not ϵ-close to the target C (in respect to $\mathcal{D}$) is at most δ. The sample is drawn accordingly to $\mathcal{D}$ and C and it is affected by a noise rate of at most η .*

For the on-line learning model we use the formalization of Angluin [1], where in each trial $t \geq 1$ the learner has to produce a hypothesis H_t, and if H_t is considered to be different from the target concept C, then the learner receives a counterexample (x_t, l_t), $l_t \in \{+, -\}$, such that $H_t(x_t) \neq l_t$. If $l_t = C(x_t)$ then (x_t, l_t) is a correct counterexample, if $l_t \neq C(x_t)$ then the counterexample is noisy. Furthermore we call an example (x, l) positive if $l = +$ and we call it negative if $l = -$. The performance of the learner is measured by its number of mistakes, i.e. by the number of counterexamples it receives until it has learned the target concept. We denote by $\mathrm{MB}(A, \mathcal{C}, N)$ the maximal number of mistakes which algorithm A makes while learning a concept from $\mathcal{C}$, if at most N of the counterexamples are noisy. For $N = 0$ we abbreviate $\mathrm{MB}(A, \mathcal{C}) := \mathrm{MB}(A, \mathcal{C}, 0)$. Furthermore we denote by $\mathrm{MB}(A, C)$ the number of mistakes algorithm A makes when learning a fixed concept C. It must be observed that in the on-line model we do not explicitely introduce a noise rate as was done in [3, 4]. Nevertheless, our results could also be stated in terms of the tolerable noise rate. Assume a bound like $\mathrm{MB}(A, \mathcal{C}, N) \leq RN + M_0$. Here R is essentially the number of additional wrong hypotheses that can result from a single noisy counterexample and M_0 is the number of wrong hypotheses when there are no noisy counterexamples at all. Then learnability can be proven for all noise rates less than $\frac{1}{R}$, [4].

1.2 Intersection-closed classes

A class $\mathcal{C}$ is intersection-closed if $\bigcap_{C \in \mathcal{C}'} C \in \mathcal{C}$ for any subclass $\mathcal{C}' \subseteq \mathcal{C}$, and if $\emptyset \in \mathcal{C}$. Intersection-closed classes can be learned using the Closure Algorithm (ClosAlg) [5, 7, 11, 6], which uses as hypothesis the closure of all positive (counter)examples seen so far. For any intersection-closed concept class $\mathcal{C}$ the closure operator $\mathrm{CL}_{\mathcal{C}} : 2^X \rightarrow 2^X$ is defined as $\mathrm{CL}_{\mathcal{C}}(S) = \bigcap_{C \in \mathcal{C}, S \subseteq C} C$. Thus the closure of a set $S \subseteq X$ is the smallest concept in $\mathcal{C}$ which

```
1. $B := S$.
2. WHILE $\exists x \in B : \mathrm{CL}(B \setminus \{x\}) = \mathrm{CL}(B)$ DO $B := B \setminus \{x\}$.
3. OUTPUT $B$.
```

Figure 1: Construction of a basis B for set S.

contains S. Since the Closure Algorithm always produces the smallest hypothesis consistent with all positive examples, in the noise-free case this hypothesis is also consistent with the negative examples. For the noisy case the Closure Algorithm was extended in [4]. If not stated otherwise we assume from now on that $\mathcal{C}$ is an intersection-closed concept class and for convenience we write CL instead of $\mathrm{CL}_{\mathcal{C}}$. If S is a set of labeled examples we write $\mathrm{CL}(S, l) := \mathrm{CL}(\{x : (x, l) \in S\})$, $l \in \{+, -\}$, for the closure of the positive or negative examples in S and we write $\mathrm{CL}(S)$ if we disregard the labels in S.

Intersection-closed classes have the following important property which we will use to construct our algorithms: for any finite set $S \subseteq X$ there is a minimal basis $\mathrm{Bas}(S) \subseteq S$, such that $\mathrm{CL}(S) = \mathrm{CL}(\mathrm{Bas}(S))$ and $B' \subset \mathrm{Bas}(S)$ implies $\mathrm{CL}(B') \subset \mathrm{CL}(S)$. The basis $\mathrm{Bas}(S)$ can be constructed by removing elements from S as long as the closure of the remaining elements equals the closure of the original set S, see Figure 1.[1] Observe that there might be more than one basis for a set S. In this case we assume that among these bases one is chosen arbitrarily. For a labeled sample S we write $\mathrm{Bas}(S, l) = \{(x, l) : x \in \mathrm{Bas}(\{x : (x, l) \in S\})\}$ for the labeled basis of the positive or negative examples in S. The size of any basis is bounded by the VC-dimension [13] of $\mathcal{C}$.

Lemma 1.2 ([7]) *For any intersection-closed concept class $\mathcal{C}$ over X and any finite set $S \subseteq X$,*

$$|\mathrm{Bas}(S)| \leq \text{VC-dim}(\mathcal{C}).$$

1.3 Nested differences

The nested difference C of concepts $C_1, \ldots, C_K \in \mathcal{C}$ is defined as

$$C = C_1 \setminus (C_2 \setminus (C_3 \setminus \ldots (C_{K-1} \setminus C_K))). \tag{1}$$

We call each C_i in (1) a *shell* of C. To simplify notation we define

$$\langle C_1, \ldots, C_K \rangle := C_1 \setminus (C_2 \setminus \ldots (C_{K-1} \setminus C_K)).$$

If $C = \langle C_1, \ldots, C_K \rangle$ then $C(x) = \ell_i$ where $i = \max\{j \geq 0 : x \in \bigcap_{j'=0}^{j} C_{j'}\}$ (we assume $C_0 = X$) and

$$\ell_i = \begin{cases} + & \text{if } i \text{ odd} \\ - & \text{if } i \text{ even} \end{cases} .$$

[1] For many concept classes this is not a very efficient algorithm, but it shows that a basis can be constructed effectively.

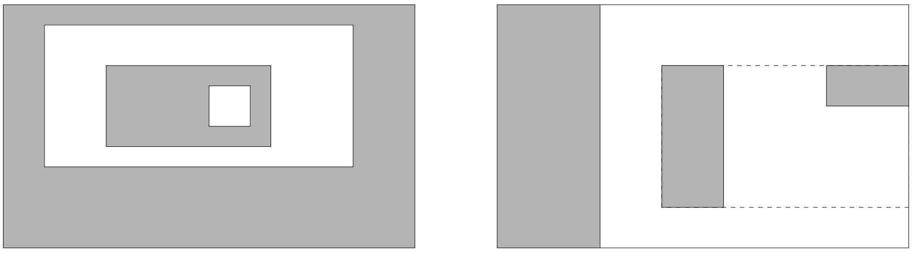

Figure 2: Examples of nested differences of rectangles with 4 and 5 shells.

```
1. "All positive examples build the first shell."
   $n := 1$, $S_1 := \{(x, +) \in S\}$.
2. REPEAT
       "All examples misclassified by the last shell build the next shell."
       $n := n + 1$, $S_n := \{(x, \ell_n) \in S : x \in \mathrm{CL}(S_{n-1})\}$.
   UNTIL $S_n = \emptyset$.
3. OUTPUT $\langle \mathrm{CL}(S_1), \ldots, \mathrm{CL}(S_{n-1}) \rangle$.
```

Figure 3: The Inclusion-Exclusion-Algorithm computes a hypothesis consistent with the noise-free sample S.

(We will use the notation ℓ_i throughout the paper for the classification associated with the i-th shell of a nested difference.) Two examples of nested differences of rectangles are show in Figure 2. The concept class of nested differences with K shells and underlying class $\mathcal{C}$ is defined as

$$\mathcal{C}^{(K)} = \{\langle C_1, \ldots, C_K \rangle : C_i \in \mathcal{C}\}$$

and the class of nested differences with an unbounded number of shells is $\mathcal{C}^{(*)} = \bigcup_{K \geq 1} \mathcal{C}^{(K)}$.

For intersection-closed classes $\mathcal{C}$ we can always obtain a normal form of a nested difference $C \in \mathcal{C}^{(K)}$.

Fact 1.3 *Let $\mathcal{C}$ be an intersection-closed concept class. Then for any $C \in \mathcal{C}^{(K)}$ there are $C_1 \supset C_2 \supset \cdots \supset C_k \neq \emptyset$, $C_i \in \mathcal{C}$, $k \leq K$, with $C = \langle C_1, \ldots, C_k \rangle$.*

Proof. Assume that $C = \langle C'_1, \ldots, C'_K \rangle$. Then also $C = \langle C''_1, \ldots, C''_K \rangle$ where $C''_i = \bigcap_{1 \leq j \leq i} C'_j$. Clearly $C''_1 \supseteq C''_2 \supseteq \cdots \supseteq C''_K$. If $C''_i = C''_{i+1}$ for some $i = 1, \ldots, K-1$ then $\langle C''_1, \ldots, C''_K \rangle = \langle C''_1, \ldots, C''_{i-1}, C''_{i+2}, \ldots, C''_K \rangle$. Thus we can remove all duplicates among the $C''_1, \ldots, C''_K$ and get $C = \langle C_1, \ldots, C_{k'} \rangle$ with $C_1 \supset \cdots \supset C_{k'}$. Finally, if $C_{k'} = \emptyset$ then $\langle C_1, \ldots, C_{k'} \rangle = \langle C_1, \ldots, C_{k'-1} \rangle$, which completes the proof. □

Helmbold, Sloan, and Warmuth [7] developed the Inclusion-Exclusion-Algorithm, Figure 3, to learn nested differences of intersection-closed classes. This algorithm first computes the closure of all positive examples, obtaining the first shell of the hypothesis. In general, this shell contains some negative examples so that the closure of these negative examples

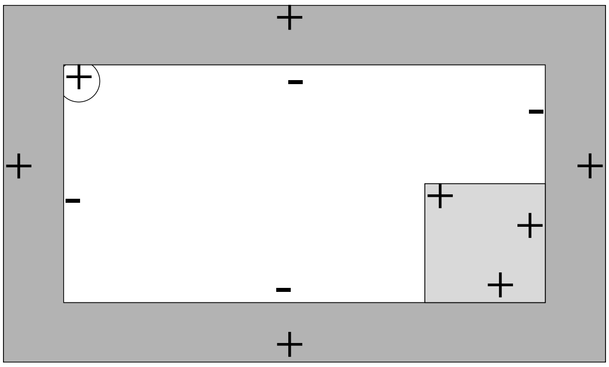

Figure 4: The noisy example ⊕ causes the Inclusion-Exclusion-Algorithm to loop forever.

must be subtracted from the first shell. This closure of the negative examples (only the negative examples in the first shell are considered) forms the second shell. Of course, in this shell there again might be positive examples, and a third, positive shell must be subtracted from the second, negative shell. This continues until a nested difference consistent with all examples is found. It can be proven that this algorithm works well in the noise-free case, but there is a problem in the noisy case. Consider Figure 4 where in the second shell there is a noisy positive example. Given by the closure of all positive examples in the second shell the third shell equals the second shell. The fourth shell, given by the closure of all negative examples in the third shell, again equals the second shell. Thus for this set of examples the Inclusion-Exclusion-Algorithm will not make any progress and cannot compute a consistent hypothesis. This is not surprising at all, since for this set of examples there is no consistent hypothesis in $\mathcal{C}^{(*)}$.

Lemma 1.4 *Let $\mathcal{C}$ be any intersection-closed concept class. Then there is no hypothesis in $\mathcal{C}^{(*)}$ consistent with examples $(x_1,+),\ldots,(x_n,+)$, $(y_1,-),\ldots,(y_{n'},-)$ if* $\mathrm{CL}(\{x_1,\ldots,x_n\}) = \mathrm{CL}(\{y_1,\ldots,y_{n'}\})$.

Proof. Assume that $H = \langle H_1,\ldots,H_k\rangle$ is consistent with the examples above and normalized such that $H_1 \supset \cdots \supset H_k \neq \emptyset$. Since $\mathrm{CL}(\{x_1,\ldots,x_n\}) = \mathrm{CL}(\{y_1,\ldots,y_{n'}\})$ we have for any $i = 1,\ldots,k$: $x_j \in H_i$ for all $j = 1,\ldots,n$ iff $y_j \in H_i$ for all $j = 1,\ldots,n'$. Since H is consistent with the examples, all $x_j \in H_1$. Thus also all $y_j \in H_1$. Again, since H is consistent, all $y_j \in H_2$. This implies that all $x_j \in H_2$. Continuing with this argument we finally find that all $x_j \in H_k$ and all $y_j \in H_k$. Hence H classifies all x_j and y_j with ℓ_k which contradicts that H is consistent with the examples. □

In Section 2 we present a PAC-learning algorithm which removes a few examples from the sample to obtain a hypothesis which is consistent with the remaining examples. Some pruning of this consistent hypothesis finally gives a hypothesis which is ϵ-close to the target concept. In Section 3 we give an on-line algorithm which does not explicitly discard previous counterexamples but which maintains its hypothesis in a way such that this hypothesis misclassifies some of the previous counterexamples but is consistent with all the other counterexamples seen so far.

We conclude this section by proving that for any intersection-closed class $\mathcal{C}$ the number of shells in the normal form of any nested difference is bounded by the mistake bound of the Closure Algorithm, $\mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C})$. To prove this we use the following lemma.

Lemma 1.5 *If* $C_1 \supset C_2$ *then* $\mathrm{MB}(\mathrm{ClosAlg}, C_1) \geq \mathrm{MB}(\mathrm{ClosAlg}, C_2) + 1$.

Proof. Consider a sequence of counterexamples to ClosAlg when learning C_2. Since in the noisy-free case ClosAlg receives only positive counterexamples all these counterexamples are in C_2. Thus all these counterexamples are also counterexamples to ClosAlg when learning C_1. After this sequence of counterexamples the hypothesis of ClosAlg is a subset of C_2. Hence any $x \in C_1 \setminus C_2$ is an additional counterexample to ClosAlg when learning C_1. Therefore ClosAlg makes at least one mistake more when learning C_1 than when learning C_2. □

Thus $C = \langle C_1, \ldots, C_{m+1} \rangle$ with $C_i \supset C_{i+1}$ and $m = \mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C})$ implies $\mathrm{MB}(\mathrm{ClosAlg}, C_{m+1}) = 0$ and hence $C_{m+1} = \emptyset$. Therefore any normal form has at most m shells and we have the following corollary.

Corollary 1.6 *Let $\mathcal{C}$ be any intersection-closed concept class. Then for any $k \geq m =$ $\mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C})$ we have $\mathcal{C}^{(k)} = \mathcal{C}^{(m)}$.*

2 Learning of nested differences in the malicious PAC-model

In this section we present an extension of the Inclusion-Exclusion-Algorithm which is robust against noise. Algorithm RobustInclusionExclusion (Figure 5) performs in two phases. In the first phase it removes examples from the sample until there is a hypothesis in $\mathcal{C}^{(*)}$ which is consistent with the remaining sub-sample. In general this sub-sample will still contain noisy examples. We will see (Lemma 2.5) that these noisy examples might force the consistent hypothesis to be much more complex than the target concept. This can be seen as a case of overfitting in the attempt to be consistent also with the noisy examples. In general this complex consistent hypothesis will not be ϵ-close to the target concept. Therefore, in the second phase algorithm RobustInclusionExclusion prunes the complex consistent hypothesis to obtain a hypothesis which is only moderately more complex than the target concept. This pruned hypothesis is consistent with less examples from the sample than the complex hypothesis, but nevertheless we are able to show that the pruned hypothesis is ϵ-close to the target concept.

In the first phase algorithm RobustInclusionExclusion has to detect noisy examples which cause any hypothesis from $\mathcal{C}^{(*)}$ to be inconsistent with the sample. Recall that the Inclusion-Exclusion-Algorithm does not make progress only if two consecutive shells of its "hypothesis" are equal. Hence, in this case, algorithm RobustInclusionExclusion removes the bases of these shells. Since the closures of both bases are equal and the examples in one basis are labeled $+$ and the the examples in the other basis are labeled $-$ at least one of these examples is noisy by Lemma 1.4. Thus in one step the algorithm removes at least 1 noisy example and at most $2d-1$ correct examples when d is the VC-dimension of $\mathcal{C}$, since d upper bounds

```
Input: Sample $S$ and upper bound $K$ on the number of shells of the target concept.
Phase 1:
1. $n := 0$, $S_0 = S$.
2. REPEAT
        $n := n + 1$, $S_n := \{(x, \ell_n) \in S_0 : x \in \mathrm{CL}(S_{n-1})\}$.
        IF $\mathrm{CL}(S_n, \ell_n) = \mathrm{CL}(S_{n-1}, \ell_{n-1})$
        THEN $n := 0$, $S_0 := S_0 \setminus (\mathrm{Bas}(S_n, \ell_n) \cup \mathrm{Bas}(S_{n-1}, \ell_{n-1}))$.
   UNTIL $S_n = \emptyset$.
Phase 2:
1. $n := 0$.
2. REPEAT
        $n := n + 1$, $S_n := \{(x, \ell_n) \in S_0 : x \in \mathrm{CL}(S_{n-1})\} \setminus \bigcup_{i=1}^{n-1} \mathrm{Bas}(S_i, \ell_i)$.
        IF $n > 2K + 1$
        THEN $n := 0$, $S_0 := S_0 \setminus \bigcup_{i=1}^{n} \mathrm{Bas}(S_i, \ell_i)$.
   UNTIL $S_n = \emptyset$.
3. OUTPUT $\langle \mathrm{CL}(S_1), \ldots, \mathrm{CL}(S_{n-1}) \rangle$.
Remark:
   All sets $S_n$ have to be implemented as multi-sets. For example, if $(x, +)$ appears
   twice in the sample then initially it will appear twice in $S_0$.
```

Figure 5: Algorithm RobustInclusionExclusion constructs a hypothesis in $\mathcal{C}^{(2K)}$.

the number of examples in any basis. Phase 1 stops as soon as there are only examples left which are consistent with some hypothesis in $\mathcal{C}^{(*)}$.

In Phase 2 examples are removed until there is an (almost) consistent hypothesis with at most twice as many shells as the target concept. Let us assume that the target concept has K shells and that during Phase 2 algorithm RobustInclusionExclusion constructs a hypothesis with $n = K + 2N$ shells. We will show (Lemma 2.5) that in this case at least N shells have noisy examples in their bases. Thus removing the bases of all shells removes at most $d(K + 2N)$ examples and at least N noisy examples. Since the number of noisy examples is reduced there is now a hypothesis with fewer shells. Repeating this process finally yields a hypothesis in $\mathcal{C}^{(2K)}$. Observe that there is a subtle point in Step 2 of Phase 2 of algorithm RobustInclusionExclusion: the bases of previously constructed shells are removed before a new shell is constructed. This guarantees that each example is an element of at most one basis. On the other hand the examples in these bases might be misclassified by the final hypothesis since they were not considered when subsequent shells were constructed. Nevertheless we can show that the final hypothesis is ϵ-close to the target concept.

Theorem 2.1 *Let $\mathcal{C}$ be any intersection-closed concept class with* VC-dim$(\mathcal{C}) = d < \infty$ *and let $K \geq 1$. Then for any $\epsilon, \delta > 0$ and any $\eta = \frac{\epsilon}{4d} - \alpha$, $0 < \alpha \leq \frac{\epsilon}{4d}$, algorithm* RobustInclusionExclusion *(ϵ, δ)-learns $\mathcal{C}^{(K)}$ in the malicious PAC model with noise rate η when provided with a sample S of size $m \geq \max\left\{\frac{16\epsilon}{\alpha^2}\left(2dK \ln \frac{48\epsilon}{\alpha^2} + \ln \frac{8}{\delta}\right), \frac{8}{\alpha^2} \log \frac{2}{\delta}\right\}$. Algorithm* RobustInclusionExclusion *outputs a hypothesis in $\mathcal{C}^{(2K)}$ and runs in time polynomial* *in the sample size m and the time needed to compute the closure and a basis of a set of size m.*

Remark 2.2 *Algorithm* RobustInclusionExclusion *can be modified so that it tolerates a noise rate up to $\frac{\epsilon}{2d}$. This can be done by changing the bound on n in Phase 2 of the algorithm. Instead of producing a hypothesis in $\mathcal{C}^{(2K)}$ the modified algorithm produces a hypothesis whose number of shells depends on $\frac{\epsilon}{2d} - \eta$. Another modification of* RobustInclusionExclusion *gives an algorithm which outputs a hypothesis in $\mathcal{C}^{(K)}$. The drawback of this algorithm is that it tolerates only a noise rate of $O(\frac{\epsilon}{Kd})$.*

Remark 2.3 *We made no attempt to optimize the bound on the sample size.*

2.1 Proof of Theorem 2.1

Assume that $S' \subseteq S$ is the sub-sample which was not effected by noise. We set $m' = |S'|$ so that $N = m - m'$ is the number of noisy examples. Since the examples which were effected by noise were chosen at random, S' is a noise-free sample in the sense of the original PAC-model. Thus we will bound the number of examples in S' which are misclassified by the algorithm's hypothesis and then apply the following result from PAC-learning theory. Essentially the lemma states that with high probability a hypothesis which makes few mistakes on a noise-free sample is close to the target concept.

Lemma 2.4 (Adapted from [2]) *Let C be any target concept and $\mathcal{H}$ any hypothesis class on a domain X, $d = \text{VC-dim}(\mathcal{H})$. Furthermore let $\mathcal{D}$ be any distribution on X, and choose $\epsilon, \delta, \alpha > 0$. Then with probability at most δ a sample of size $m' \geq \frac{8\epsilon}{\alpha^2}\left(d \ln \frac{48\epsilon}{\alpha^2} + \ln \frac{4}{\delta}\right)$ is drawn accordingly to $\mathcal{D}$ and labeled by C, such that there is an $H \in \mathcal{H}$ which is not ϵ-close to C but makes at most $(\epsilon - \alpha)m'$ mistakes on the sample.*

To bound the number of misclassified examples in S' observe that there are two kinds of misclassified examples. Obviously examples which were removed from S_0 in Phase 1 or 2 might be misclassified by the final hypothesis. Furthermore, some examples in the bases of the shells of the final hypothesis might be misclassified. But observe that there are at most $2dK$ examples in these bases. Thus we only have to bound the number of examples which are removed from S' by the algorithm. Let s_1 and s_2 be the number of examples removed from S' during Phase 1 and Phase 2, respectively, and let N_1 and N_2 be the number of noisy examples removed during Phase 1 and Phase 2, respectively.

In Phase 1 the examples in $\text{Bas}(S_{n+1})$ and $\text{Bas}(S_n)$ are removed from S_0 iff $\text{CL}(\text{Bas}(S_{n+1})) = \text{CL}(\text{Bas}(S_n))$. By Lemma 1.4 at least one of these examples is noisy. Since $|\text{Bas}(S_{n+1}) \cup \text{Bas}(S_n)| \leq 2d$, with each noisy example at most $2d - 1$ correct examples are removed from S'. Thus we find $s_1 \leq (2d-1)N_1$.

To analyze Phase 2 we have to calculate the number of shells which are created by noisy examples: besides the K shells which correspond to the shells of the target concept each noisy counterexample generates at most 2 additional shells. See Figure 6: roughly speaking, each noisy counterexample can be "covered" by 2 additional shells. The following lemma gives a little stronger statement which we will need in Section 3.

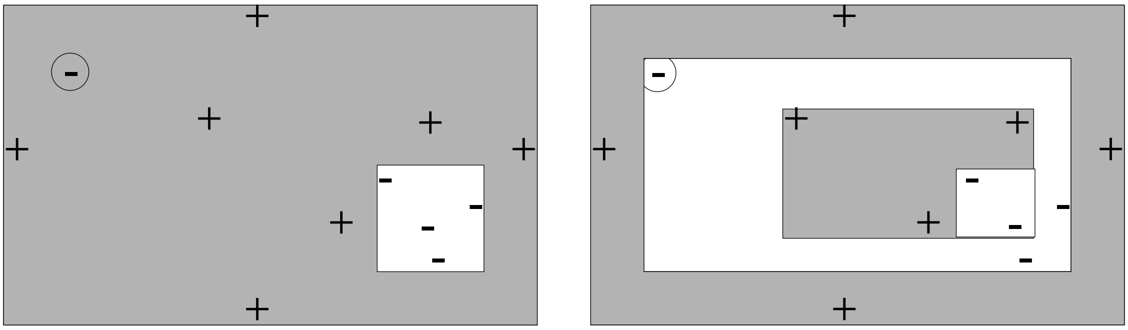

Figure 6: The noisy example $\ominus$ is covered by an additional shell. Another shell is used to cover the positive examples which would be misclassified otherwise.

Lemma 2.5 *Let $\mathcal{C}$ be any intersection-closed concept class and $C = \langle C_1, \ldots, C_K \rangle \in \mathcal{C}^{(K)}$ some target concept. Furthermore let $S_1, \ldots, S_n \subseteq X$ be a sequence of sets of examples such that the label of an example $x \in S_i$ is ℓ_i. If $\mathrm{CL}(S_1) \supseteq \mathrm{CL}(S_2) \supseteq \cdots \supseteq \mathrm{CL}(S_n) \neq \emptyset$ and at most N examples in the sets $S_1, \ldots, S_n$ are noisy (in respect to C) then $n \leq k + 2N$ for some $0 \leq k \leq K$ and there are indices $1 \leq i_1 < \cdots < i_k \leq n$ with $S_{i_j} \subseteq C_j$ for $j = 1, \ldots, k$.*

Proof. We start by constructing the indices i_j. We set $i_0 = 0$, $C_0 = S_0 = X$, $C_{K+1} = \emptyset$, and $i_{j+1} = \min\{i_j + 1 + 2s : s \geq 0, S_{i_j+1+2s} \subseteq C_{j+1}\}$ for $j = 0, \ldots, K$ (we assume $S_i = \emptyset$ for $i > n$). Observe that $\ell_{i_j} = \ell_j$. The indices i_j are chosen such that S_{i_j} is the "largest" set which is included in C_j and whose examples are labeled by ℓ_j. Let k be the number of indices i_j with $1 \leq i_j \leq n$. Obviously this choice of indices satisfies $S_{i_j} \subseteq C_j$ for $j = 0, \ldots, k$. By the construction of the i_j and the prerequisite of the lemma we have $C_j \supseteq \mathrm{CL}(S_{i_j}) \supseteq S_{i_j+1+2s}$ for all $s \geq 0$ and all $j = 0, \ldots, k$. Since all examples in S_{i_j+1+2s} have label ℓ_{j+1}, all correct examples in $S_{i_j+1+2s} \subseteq C_j$ are in C_{j+1}. Thus $S_{i_j+1+2s} \not\subseteq C_{j+1}$ only if S_{i_j+1+2s} contains at least one noisy example. Hence for $j = 0, \ldots, k$ each $S_{i_j+1}, S_{i_j+3}, \ldots, S_{i_{j+1}-2}$ contains at least one noisy example since $S_{i_{j+1}}$ is the first set with $S_{i_j+1+2s} \subseteq C_{j+1}$. Counting these sets for all j gives

$$N \geq \sum_{j=0}^{k} \frac{i_{j+1} - i_j - 1}{2} = \frac{i_{k+1} - i_0 - (k+1)}{2} \geq \frac{n-k}{2}$$

and the lemma. □

In Phase 2 the bases of all sets S_i are removed from S_0. Since the bases $\mathrm{Bas}(S_i)$, $i = 1, \ldots, n$, fulfill the prerequisite of Lemma 2.5 at least $\frac{n-K}{2}$ bases contain a noisy example. Thus by removing all bases at least $\frac{n-K}{2}$ noisy examples are removed. Since at most a total number of dn examples is removed at most $dn - \frac{n-K}{2}$ examples are removed from S'. Thus per noisy example at most

$$\frac{dn - \frac{n-K}{2}}{\frac{n-K}{2}} = \frac{2dn}{n-K} - 1 \leq \frac{4dK}{K} - 1 = 4d - 1$$

examples from S' are removed. Hence we get $s_2 \leq (4d-1)N_2$.

Summing over Phase 1 and Phase 2 we find that $s_1 + s_2 \leq (4d-1)N$. Now we have to bound the number of noisy examples N. The number of noisy examples is the sum of m independent Bernoulli trials whose probability of success is at most η. Thus we get by standard Höffding bounds that $N \leq m(\eta + \frac{\alpha}{2})$ with probability at least $1 - \frac{\delta}{2}$ if $m \geq \frac{8}{\alpha^2} \log \frac{2}{\delta}$. Recalling that there are at most $2dK$ examples in the bases of the final hypothesis we find that with probability at least $1 - \frac{\delta}{2}$ the algorithm's hypothesis misclassifies at most a fraction of

$$\begin{aligned} \frac{(4d-1)N + 2dK}{m'} &= \frac{(4d-1)N + 2dK}{m - N} \\ &\leq \frac{(4d-1)(\eta + \alpha/2)}{1 - \eta - \alpha/2} + \frac{4dK}{m} \\ &\leq \frac{(4d-1)(\frac{\epsilon}{4d} - \alpha/2)}{1 - \frac{\epsilon}{4d} + \alpha/2} + \frac{\alpha^2}{8\epsilon} \qquad (2) \end{aligned}$$

examples in S'. Some algebra shows that (2) $\leq \epsilon - \alpha$ for $0 < \epsilon < 1$, $0 < \alpha \leq \frac{\epsilon}{4d}$. By applying Lemma 2.4 with $\mathcal{H} = \mathcal{C}^{(2K)}$, ϵ, $\delta/2$, and α, we get that with probability $1 - \frac{\delta}{2}$ the algorithm's hypothesis is ϵ-close to the target if

$$m' \geq \frac{8\epsilon}{\alpha^2} \left(\text{VC-dim}(\mathcal{C}^{(2K)}) \ln \frac{48\epsilon}{\alpha^2} + \ln \frac{8}{\delta} \right), \qquad (3)$$

provided that $N \leq m(\eta + \frac{\alpha}{2})$. The VC-dimension of $\mathcal{C}^{(2K)}$ is bounded by the following lemma.

Lemma 2.6 ([7]) *For any intersection-closed class $\mathcal{C}$*

$$\text{VC-dim}(\mathcal{C}^{(K)}) \leq K \cdot \text{VC-dim}(\mathcal{C}).$$

At last we have with probability $1 - \frac{\delta}{2}$ that $m' = m - N \geq m(1 - \eta - \frac{\alpha}{2}) \geq \frac{m}{2}$ (for $0 < \epsilon < 1$) which implies (3). Summing up we find that with probability $1 - \delta$ the algorithm's hypothesis is ϵ-close to the target, which gives the theorem.

2.2 Discussion and related results

Kearns and Li [8] presented a general technique to make a PAC-learning algorithm noise robust. They show that any time-efficient learning algorithm for the noise-free PAC model which uses a sample of size m, can be turned into a time-efficient PAC learning algorithm tolerating a malicious noise rate up to $\Omega\left(\frac{\log m}{m}\right)$. In general, $\mathcal{C}^{(K)}$ can be (ϵ, δ)-learned in the noise-free case using a sample of size $m = O\left(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{dK}{\epsilon} \log \frac{1}{\epsilon}\right)$ [7], where $d = \text{VC-dim}(\mathcal{C})$. Applying the result of Kearns and Li gives for small ϵ a tolerable noise rate of $\Omega\left(\frac{\epsilon}{dK}\right)$. While this bound on the tolerable noise rate depends on the number of shells, our result gives an error tolerance of $\Omega\left(\frac{\epsilon}{d}\right)$ which is independent of the number of shells. The error tolerance of our algorithm is the best known for example for the class of nested differences of axis-parallel rectangles.

Talking about achievable noise tolerance one has to be aware of the fact that we are considering only time-efficient algorithms. Time-efficiency essentially means that the algorithms run in time polynomial in the sample size. If the run time of the learning algorithm

Algorithm XInclusionExclusion maintains a sequence of sets of counterexamples $\mathcal{S} = \langle S_1, \ldots, S_n \rangle$, $n \geq 0$, such that $\mathrm{CL}(S_i) \supseteq \mathrm{CL}(S_{i+1})$ for all $i = 1, \ldots, n-1$.

Initialization:
Initialize $\mathcal{S}$ to the empty sequence, $\mathcal{S} := \langle\rangle$, $n := 0$.

Construction of the hypothesis:
In each trial $t \geq 1$ set $H_t := \langle \mathrm{CL}(S_1), \ldots, \mathrm{CL}(S_n) \rangle$. ($H_1 = \langle\rangle = \emptyset$.)

Update:
Let (x_t, l_t) be the counterexample to H_t.
Set $i := \min\{1 \leq j \leq n : x_t \notin \mathrm{CL}(S_j)\}$. If $x_t \in \mathrm{CL}(S_j)$ for all j then $i := n+1$.
Update $\mathcal{S}$ by setting $S_i := S_i \cup \{x_t\}$. If $i = n+1$ set $S_{n+1} := \{x_t\}$ and $n := n+1$.

Figure 7: Algorithm XInclusionExclusion for the on-line learning of nested differences with noise.

is not constrained then the optimal noise tolerance of $\frac{\epsilon}{1+\epsilon}$ can be achieved. This is done by searching for a hypothesis which misclassifies a minimal number of examples in the sample. The optimality of $\frac{\epsilon}{1+\epsilon}$ was proven in [8].

3 On-line learning of nested differences in the presence of noise

In this section we present algorithm XInclusionExclusion, Figure 7, which is a variation of the Inclusion-Exclusion-Algorithm and has the advantage that in the presence of noise it still produces a hypothesis consistent with most of the counterexamples seen so far. We start with an informal description of the algorithm. Like the Inclusion-Exclusion-Algorithm its hypothesis H_t is the nested difference of the closures of some sets of counterexamples $S_1, S_2, \ldots$ For each trial the Inclusion-Exclusion-Algorithm calculates these sets from scratch, such that S_1 is the set of all positive counterexamples seen so far, S_2 is the set of all negative counterexamples in the closure of S_1, and so on. In contrast, algorithm XInclusionExclusion updates the sets $S_1, S_2, \ldots$ incrementally: the counterexample x_t is added to the set S_i with the smallest index i such that $x_t \notin \mathrm{CL}(S_i)$. Since before the update $x_t \in \mathrm{CL}(S_{i-1}) \setminus \mathrm{CL}(S_i)$ and x_t was a counterexample to H_t, the label of x_t is ℓ_i. While the hypothesis of the Inclusion-Exclusion-Algorithm is always consistent with all counterexamples seen so far (therefore the Inclusion-Exclusion-Algorithm cannot tolerate noisy counterexamples), the hypothesis of algorithm XInclusionExclusion is in general not consistent with all the counterexamples (which enables the algorithm to deal with noise), see Figure 8. We get the following mistake bound for algorithm XInclusionExclusion and we will show that this bound is optimal.

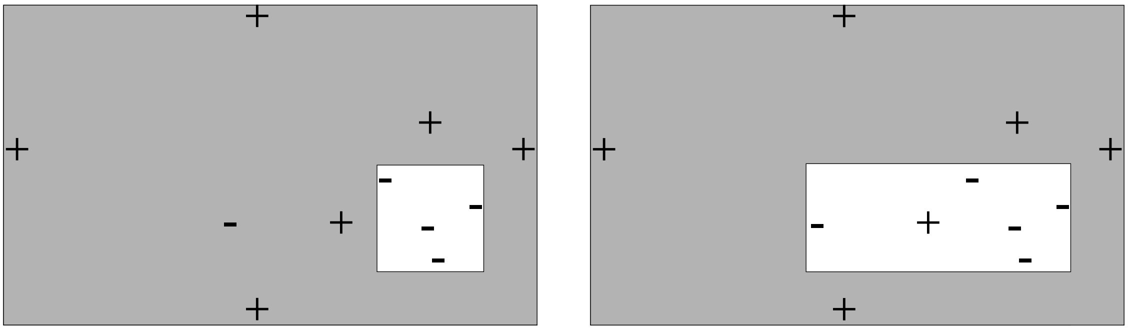

Figure 8: The new negative counterexample, added to S_2, enlarges the second shell of the hypothesis. Since all positive counterexamples remain in S_1, one of them (not necessarily a noisy one if the target is from $\mathcal{C}^{(3)}$) is misclassified by the hypothesis.

Theorem 3.1 *For any intersection-closed concept class $\mathcal{C}$ and for any $2 \leq K \leq \mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C})$ and any $N \geq 0$*

$$\mathrm{MB}(\mathrm{XInclusionExclusion}, \mathcal{C}^{(K)}, N) \leq (2N + K) \cdot \mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C}) - \frac{K(K-1)}{2}$$

where algorithm XInclusionExclusion *uses hypotheses from $\mathcal{C}^{(K+2N)}$ and runs in time polynomial in K, N, and the maximal time taken by* ClosAlg *to learn a concept from $\mathcal{C}$.*

Theorem 3.2 *For any $m \geq 2$ there is an intersection-closed concept class $\mathcal{C}$ with $\mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C}) = m$ such that*

$$\mathrm{MB}(A, \mathcal{C}^{(K)}, N) \geq (2N + K) \cdot m - \frac{K(K-1)}{2}$$

for any $N \geq 0$, any $2 \leq K \leq m$, and for any on-line learning algorithm A which uses hypothesis from $\mathcal{C}^{(*)}$.

Proof of Theorem 3.1. To prove the mistake bound on algorithm XInclusionExclusion we show that there are at most $K + 2N$ sets S_i and we bound the number of examples in each of these sets. We use the notation of Figure 7. Let $C = \langle C_1, \ldots, C_{K'} \rangle$, $K' \leq K$, be the normalized target concept. Since the sets S_i fulfill the prerequisite of Lemma 2.5 we have $n \leq k + 2N$ for some $0 \leq k \leq K'$ and there are indices $1 \leq i_1 < \cdots < i_k \leq n$ with $S_{i_j} \subseteq C_j$. Furthermore observe that counterexample x_t is added to set S_i only if $x_t \notin \mathrm{CL}(S_i)$. Thus the sequence of counterexamples $x_{t_1}, x_{t_2}, \ldots$ added to set S_i is also a sequence of counterexamples to the Closure Algorithm when learning the concept class $\mathcal{C}$. Moreover, the sequence $x_{t_1}, x_{t_2}, \ldots$ added to a set S_{i_j} is a sequence of counterexamples to the Closure Algorithm when learning C_j. Hence $|S_i| \leq \mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C})$ for all $1 \leq i \leq n$. Using the fact that $C_1 \supset \cdots \supset C_{K'}$ and Lemma 1.5 we find $|S_{i_j}| \leq \mathrm{MB}(\mathrm{ClosAlg}, C_j) \leq \mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C}) - j + 1$. Since $n \leq 2N + k$ summing over all sets S_i gives the theorem. □

3.1 Proof of Theorem 3.2

For the proof of the lower bound we will show that nested differences of linear sub-spaces are hard to learn. Let $\mathcal{C}$ be the concept class of all linear sub-spaces of the vector space $\mathbf{Z}_p^m$ over the field $\mathbf{Z}_p$ where p is an arbitrary prime $p > m$. (The field $\mathbf{Z}_p$ is given by the set $\{0, \ldots, p-1\}$ and the operations addition and multiplication modulo p [9].) For this concept class $\mathrm{MB}(\mathrm{ClosAlg}, \mathcal{C}) = m$ [4]. We will show that there are $2m$ vectors in $\mathbf{Z}_p^m$ such that to any on-line algorithm these vectors can be given as counterexamples several times resulting in a total number of $2mN + 2m - 1$ counterexamples. Among these counterexamples at most N will be noisy. After these $2mN + 2m - 1$ counterexamples the learning algorithm will not have gained sufficient information about the target concept so that it receives another $m(K-2) - \frac{K(K-1)}{2} + 1$ counterexamples before finally learning the target.

We start proving the theorem for the case $m = K = 2$ which gives an idea how the proof in the general case works. In this case $m(K-2) - \frac{K(K-1)}{2} + 1 = 0$ such that the information from the first $2mN + 2m - 1$ counterexamples is sufficient for learning. For the general case we will have to deal also with the $m(K-2) - \frac{K(K-1)}{2} + 1$ additional counterexamples.

Observe that for vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbf{Z}_p^m$ the closure $\mathrm{CL}(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\})$ is the linear sub-space spanned by these vectors. We set $\boldsymbol{b}_1 = (1, 0)$, $\boldsymbol{b}_2 = (0, 1)$, $\boldsymbol{u}_1 = (1, 1)$, and $\boldsymbol{u}_2 = (1, -1)$. Since $\mathrm{CL}(\{\boldsymbol{b}_1, \boldsymbol{b}_2\}) = \mathrm{CL}(\{\boldsymbol{u}_1, \boldsymbol{u}_2\}) = \mathbf{Z}_p^2$ there is no concept in $\mathcal{C}^{(*)}$ consistent with the labels $(\boldsymbol{b}_1, +), (\boldsymbol{b}_2, +), (\boldsymbol{u}_1, -), (\boldsymbol{u}_2, -)$ by Lemma 1.4. Thus among the $(\boldsymbol{b}_1, +), (\boldsymbol{b}_2, +), (\boldsymbol{u}_1, -), (\boldsymbol{u}_2, -)$ there is a counterexample to any hypothesis from $\mathcal{C}^{(*)}$. After $4N + 3$ trials one of these counterexamples has been given at most N times so that we consider this counterexample to be noisy. We can do that because a learning algorithm has to work for any target concept and for any selection of noisy examples. Knowing the learner's hypotheses in advance (since we are considering deterministic algorithms) we can pick a target concept which is consistent with all but the noisy examples. If $(\boldsymbol{b}_1, +)$ is this noisy counterexample than the remaining counterexamples are consistent with $\mathrm{CL}(\{\boldsymbol{b}_2\})$. If $(\boldsymbol{b}_2, +)$ is the noisy counterexample than the remaining counterexamples are consistent with $\mathrm{CL}(\{\boldsymbol{b}_1\})$. If $(\boldsymbol{u}_1, -)$ is the noisy counterexample than the remaining counterexamples are consistent with $\mathrm{CL}(\{\boldsymbol{b}_1, \boldsymbol{b}_2\}) \setminus \mathrm{CL}(\{\boldsymbol{u}_2\})$, and if $(\boldsymbol{u}_2, -)$ is the noisy counterexample than the remaining counterexamples are consistent with $\mathrm{CL}(\{\boldsymbol{b}_1, \boldsymbol{b}_2\}) \setminus \mathrm{CL}(\{\boldsymbol{u}_1\})$. Thus for each algorithm there is a hypothesis from $\mathcal{C}^{(2)}$ which can force $4N + 3$ mistakes of the algorithm if N of the counterexamples might be noisy.

For the case $m \geq 3$ we use a somewhat more sophisticated argument and some tools from linear algebra. Let $\boldsymbol{x} \cdot \boldsymbol{y}$ denote the inner product of the vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{Z}_p^m$. Furthermore we call vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbf{Z}_p^m$ orthonormal if $\boldsymbol{x}_i \cdot \boldsymbol{x}_i \equiv 1(p)$ for all $i = 1, \ldots, n$, and $\boldsymbol{x}_i \cdot \boldsymbol{x}_j \equiv 0(p)$ for $i \neq j$. We will make use of the following lemma.

Lemma 3.3 *If for $n \geq 3$ $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ are orthonormal vectors from $\mathbf{Z}_p^m$ then there are also orthonormal vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n \in \mathbf{Z}_p^m$ such that*

1. $\mathrm{CL}(\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}) = \mathrm{CL}(\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\})$,

2. *$\boldsymbol{b}_i \notin \mathrm{CL}(U)$ for all $\boldsymbol{b}_i$ and any subset U of at most $n-1$ vectors from $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$,*

3. *$\boldsymbol{u}_i \notin \mathrm{CL}(B)$ for all $\boldsymbol{u}_i$ and any subset B of at most $n-1$ vectors from $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.*

Proof. Basicly we get the $\boldsymbol{u}_i$ by rotating the $\boldsymbol{b}_i$. We set

$$\boldsymbol{u}_i = \left(\sum_{j \neq i} \boldsymbol{b}_j + (1 - 2^{-1}n)\boldsymbol{b}_i \right) 2n^{-1}$$

where x^{-1} denotes the multiplicative inverse in the field $\mathbf{Z}_p$. Since the $\boldsymbol{b}_i$ are orthonormal we find

$$\boldsymbol{u}_i \cdot \boldsymbol{u}_i \equiv \left(n - 1 + (1 - 2^{-1}n)^2\right) 4n^{-2} \equiv 1(p)$$

and

$$\boldsymbol{u}_i \cdot \boldsymbol{u}_j \equiv \left(n - 2 + 2(1 - 2^{-1}n)\right) 4n^{-2} \equiv 0(p)$$

for $i \neq j$. Thus the vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ are orthonormal and therefore linear independent. Since $\mathrm{CL}(\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}) \subseteq \mathrm{CL}(\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\})$ and the linear sub-space spanned by $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ has dimension n as the linear sub-space spanned by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, we have $\mathrm{CL}(\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}) = \mathrm{CL}(\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\})$. Furthermore $\boldsymbol{b}_i \cdot \boldsymbol{u}_j \not\equiv 0(p)$ for all i and j implies that no $\boldsymbol{b}_i$ can be expressed as the linear combination of $n - 1$ of the vectors $\boldsymbol{u}_j$ and vice versa. □

We are ready now to prove Theorem 3.2 for $m \geq 3$. Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ be the unit vectors of $\mathbf{Z}_p^m$ (which obviously are orthonormal) and let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ be the alternate orthonormal vectors given by Lemma 3.3. We label all $\boldsymbol{b}_i$ with $+$ and all $\boldsymbol{u}_i$ with $-$. Since $\mathrm{CL}(\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}) = \mathrm{CL}(\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\})$ there is no hypothesis in $\mathcal{C}^{(*)}$ consistent with all these labels and one of the labeled vectors can be given as counterexample to any hypothesis. After $2mN + 2m - 1$ trials one of the vectors was given as a counterexample at most N times and we consider this labeled vector to be the noisy counterexample. If $\boldsymbol{b}_i$ is the noisy vector then the labels of all other vectors are consistent with $\mathrm{CL}(\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}, \boldsymbol{b}_{i+1}, \ldots, \boldsymbol{b}_m\})$ since no $\boldsymbol{u}_i$ is in this closure by Lemma 3.3. If $\boldsymbol{u}_i$ is the noisy vector then $\mathbf{Z}_p^m \setminus \mathrm{CL}(\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{i-1}, \boldsymbol{u}_{i+1}, \ldots, \boldsymbol{u}_m\})$ is consistent with the labels of the remaining vectors. Therefore we can pick a concept $\langle C_1 \rangle$ or $\langle C_1, C_2 \rangle$ such that after $2mN + 2m - 1$ counterexamples this concept is consistent with all but N of the counterexamples. Furthermore C_1 resp. C_2 is spanned by $m - 1$ orthonormal vectors.

We proceed by proving by induction that for any $2 \leq K \leq m - 1$ we can force any learner to make at least $2mN + \sum_{l=0}^{K-1}(m - l) = \left[2mN + 2m - 1\right] + \left[m(K - 2) - \frac{K(K-1)}{2} + 1\right]$ mistakes while there is still a concept in $\mathcal{C}^{(K)}$ consistent with all but the N noisy counterexamples. Obviously this gives the theorem for $2 \leq K \leq m - 1$. Above we have already proven this for $K = 2$. To get the remaining number of mistakes we basicly use the fact that in the noise-free case any algorithm can be forced to make at least $m - l$ mistakes when learning a linear sub-space of dimension $m - l$. Since the shells $C_1, C_2, \ldots, C_K$ have essentially dimensions $m, m - 1, \ldots, m - K + 1$ we get the result. But of course we have to take care that a counterexample given while learning C_i does not help the learner when learning C_j for some $j > i$.

For $K \geq 3$ let $\langle C_1, \ldots, C_l \rangle$, $l \leq K - 1$, be the concept consistent with previous correct counterexamples such that the following holds: C_l is spanned by orthonormal vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{m-K+2}$, no previous counterexamples besides the $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{m-K+2}$ are elements of C_l, and the learner has already made $2mN + \sum_{l=0}^{K-2}(m - l)$ mistakes. By the constructions in the above paragraphs these conditions are fulfilled for $K = 3$. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{m-K+2}$

be the alternate orthonormal vectors given by Lemma 3.3 such that $\mathrm{CL}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{m-K+2}) = \mathrm{CL}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{m-K+2})$. The labels of all $\boldsymbol{b}_i$ are ℓ_l and we label all $\boldsymbol{u}_i$ with ℓ_{l+1}. Thus one of these labeled vectors can be given as counterexample to any hypothesis from $\mathcal{C}^{(*)}$. After $m-K+1$ trials at least one of the $\boldsymbol{u}_i$ was not given as counterexample. We denote this vector by $\boldsymbol{u}_{i^*}$. Then the concept $\langle C_1, \ldots, C_{l+1}\rangle \in \mathcal{C}^{(K)}$, $C_{l+1} = \mathrm{CL}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{i^*-1}, \boldsymbol{u}_{i^*+1}, \ldots, \boldsymbol{u}_{m-K+2})$, is consistent with all correct counterexamples (note that none of the $\boldsymbol{b}_i$ or $\boldsymbol{u}_i$ was noisy): by Lemma 3.3 no $\boldsymbol{b}_i$ is contained in C_{l+1}, and since $C_{l+1} \subseteq C_l$ also no of the other previous counterexamples is contained in C_{l+1}. Thus all requirements for the next induction step are met.

At last we have to deal with the case $K = m \geq 3$. From the above considerations we know that there is a concept $\langle C_1, \ldots, C_l\rangle$, $l \leq m-1$, consistent with all previous correct counterexamples such that C_l is spanned by orthonormal vectors $\boldsymbol{b}_1, \boldsymbol{b}_2$, no other counterexample is element of C_l, and the learner has already made $2mN + \sum_{l=0}^{m-2}(m-l)$ mistakes. Let $\boldsymbol{u}_1 = \boldsymbol{b}_1 + \boldsymbol{b}_2$ and $\boldsymbol{u}_2 = \boldsymbol{b}_1 - \boldsymbol{b}_2$. We assign the label ℓ_{l+1} to $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. If the learner's hypothesis is consistent with these labels then it is not consistent with the labels of $\boldsymbol{b}_1$ or $\boldsymbol{b}_2$ by Lemma 1.4. Thus in this case one of the $\boldsymbol{b}_i$ can be given as counterexample to the learner's hypothesis. If the learner's hypothesis is not consistent with the label of $\boldsymbol{u}_1$ then $\boldsymbol{u}_1$ is given as counterexample (analogously for $\boldsymbol{u}_2$): this forces an additional mistake of the learner and $\langle C_1, \ldots, C_l, C_{l+1}\rangle$ with $C_{l+1} = \mathrm{CL}(\{\boldsymbol{u}_1\})$ is consistent with this counterexample. This concludes the proof.

4 Conclusion

We investigated the learnability of nested differences in the presence of noise, both in the malicious PAC-learning model and in the on-line learning model. For both models we presented general algorithms which were based on the Closure Algorithm and the Inclusion-Exclusion-Algorithm. We analyzed a pruning technique used by the algorithm for the malicious PAC-learning model and we showed that this algorithm achieves a noise tolerance which is superior to previously known results. Our on-line learning algorithm was proven to obtain the best possible general mistake bound.

Acknowledgments

I want to thank Nicolò Cesa-Bianchi, David Haussler, Christiane Michel, and Manfred Warmuth for very fruitful discussions. I also gratefully acknowledge support by the Fonds zur Förderung der wissenschaftlichen Forschung, Austria, through grant J01028-MAT. This research was done while visiting the University of California at Santa Cruz.

References

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, April 1988.

[2] M. Anthony and J. Shawe-Taylor. A result of Vapnik with applications. *Discrete Applied Mathematics*, 47:207–217, 1993.

[3] Peter Auer. On-line learning of rectangles in noisy environments. In *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory*, pages 253–261. ACM Press, 1993.

[4] Peter Auer and Nicolò Cesa-Bianchi. On-line learning with malicious noise and the closure algorithm. In Setsuo Arikawa and Klaus P. Jantke, editors, *Algorithmic Learnung Theory, AII'94, ALT'94*, pages 229–247. Lecture Notes in Artificial Intelligence 872, Springer, 1994. The journal version was accepted for publication in *Annals of Mathematics and Artificial Intelligence.*

[5] D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting {0,1} functions on randomly drawn points. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 100–109. IEEE Computer Society Press, 1988.

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

[7] David Helmbold, Robert Sloan, and Manfred K. Warmuth. Lerning nested differences of intersection-closed concept classes. *Machine Learning*, 5:165–196, 1990.

[8] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM J. Comput.*, 22:807–837, 1993.

[9] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and Their Applications.* Cambridge University Press, revised edition, 1994.

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

[11] B. K. Natarajan. *Machine Learning: A Theoretical Approach.* Morgan Kaufmann, San Mateo, CA, 1991.

[12] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.

[13] 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.

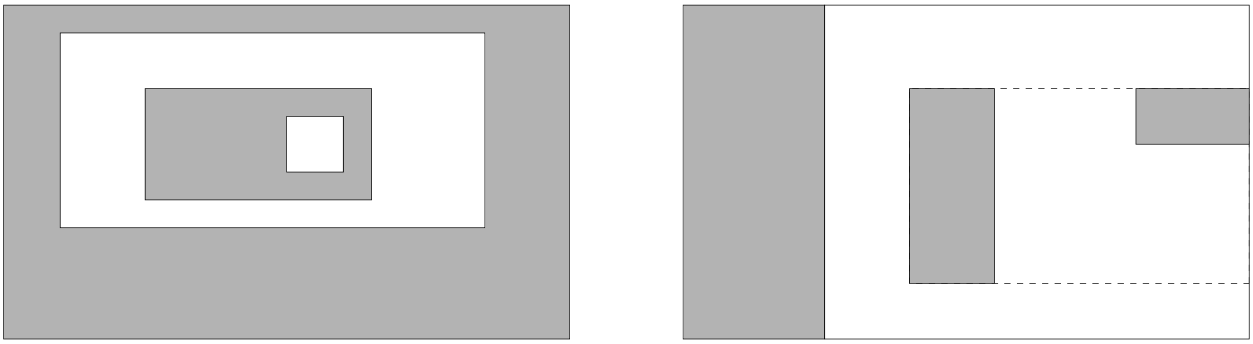

Figure 2: Examples of nested differences of rectangles with 4 and 5 shells.

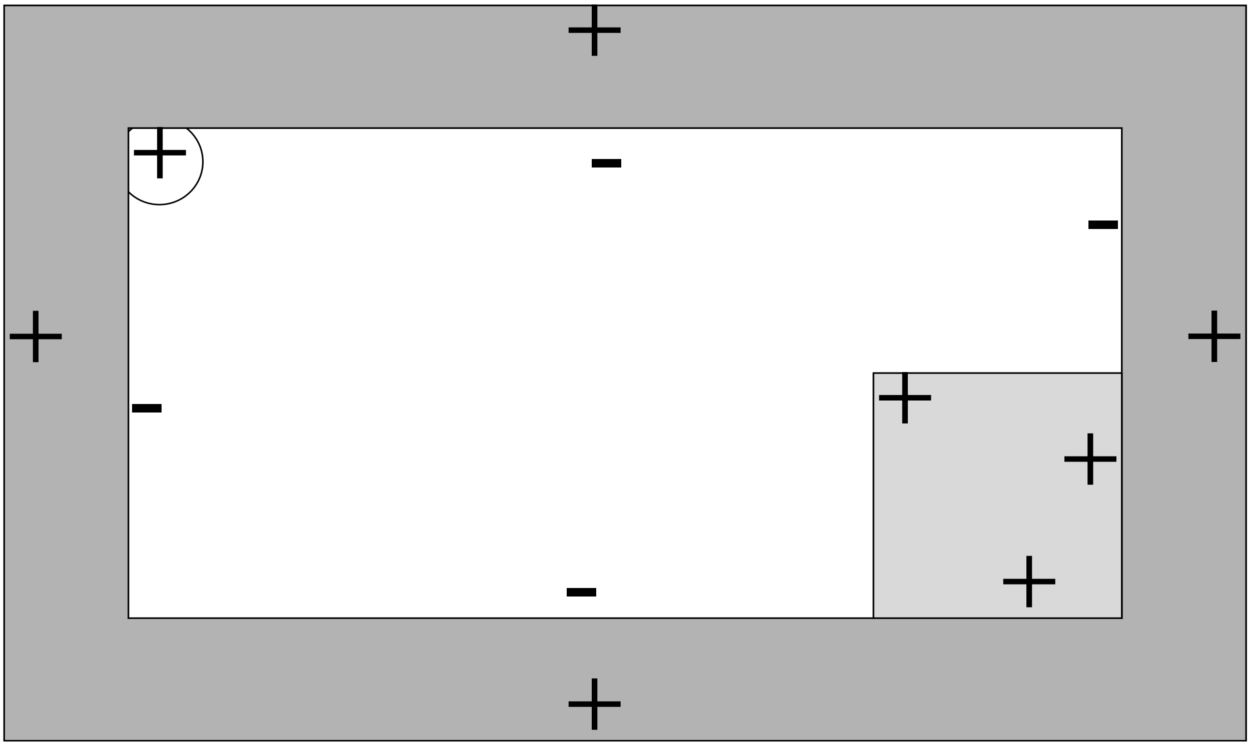

Figure 4: The noisy example $\oplus$ causes the Inclusion-Exclusion-Algorithm to loop forever.

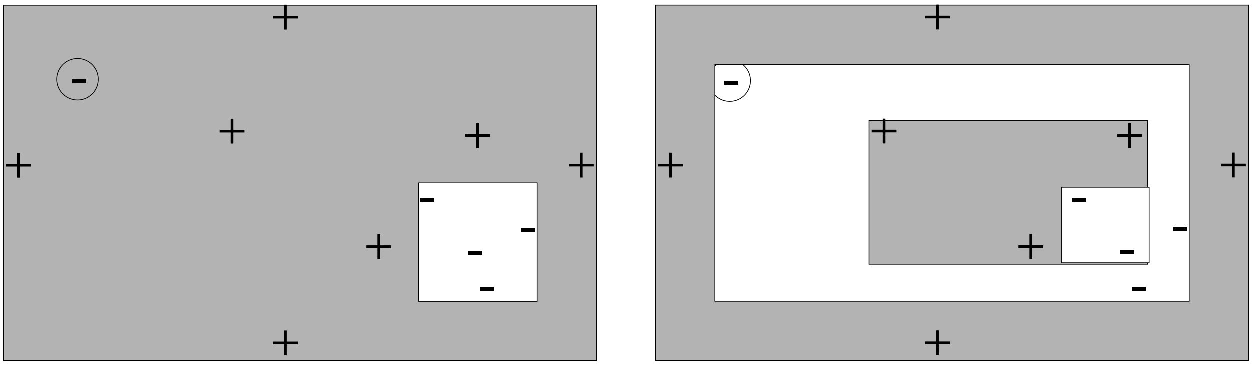

Figure 6: The noisy example $\ominus$ is covered by an additional shell. Another shell is used to cover the positive examples which would be misclassified otherwise.

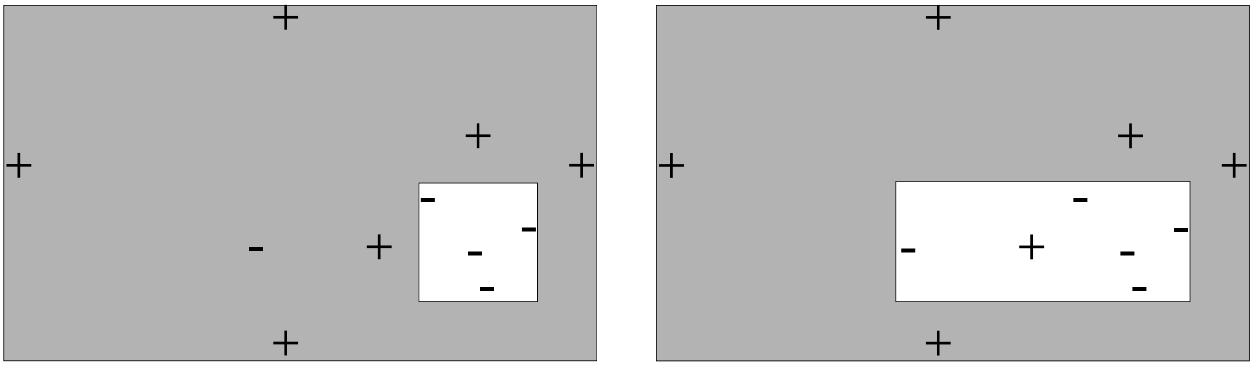

Figure 8: The new negative counterexample, added to S_2, enlarges the second shell of the hypothesis. Since all positive counterexamples remain in S_1, one of them (not necessarily a noisy one if the target is from $\mathcal{C}^{(3)}$) is misclassified by the hypothesis.