LEARNING HALF-SPACES AND OTHER CONCEPT CLASSES IN THE LIMIT WITH ITERATIVE LEARNERS

ARDALAN KHAZRAEI, TIMO KÖTZING, KAREN SEIDEL

ABSTRACT. In order to model an efficient learning paradigm, iterative learning algorithms access data one by one, updating the current hypothesis without regress to past data. Past research on iterative learning analyzed for example many important additional requirements and their impact on iterative learners.

In this paper, our results are twofold. First, we analyze the relative learning power of various settings of iterative learning, including learning from text and from informant, as well as various further restrictions, for example we show that strongly non-U-shaped learning is restrictive for iterative learning from informant.

Second, we investigate the learnability of the concept class of half-spaces and provide a constructive iterative algorithm to learn the set of half-spaces from informant.

1. INTRODUCTION

We are interested in the problem of algorithmically learning a description for a formal language (a computably enumerable subset of the set of natural numbers) when presented successively all and only the elements of that language; this is sometimes called *inductive inference*, a branch of (algorithmic) learning theory. For example, a learner M might be presented more and more even numbers. After each new number, M outputs a description for a language as its conjecture. The learner M might decide to output a program for the set of all multiples of 4, as long as all numbers presented are divisible by 4. Later, when h sees an even number not divisible by 4, it might change this guess to a program for the set of all multiples of 2.

Many criteria for deciding whether a learner M is *successful* on a language L have been proposed in the literature. Gold, in his seminal paper [Gol67], gave a first, simple learning criterion, $\mathbf{TxtEx}$-*learning*[1], where a learner is *successful* iff, on every *text* for L (a listing of all and only the elements of L) it eventually stops changing its conjectures, and its final conjecture is a correct description for the input sequence. Trivially, each single, describable language L has a suitable constant function as a $\mathbf{TxtEx}$-learner (this learner constantly outputs a description for L). As we want algorithms for more than a single learning task, we are interested in analyzing for which *classes of languages* $\mathcal{L}$ is there a *single learner* M learning *each* member of $\mathcal{L}$. This framework is also sometimes known as *language learning in the limit* and has been studied extensively, using a wide range of learning criteria similar to $\mathbf{TxtEx}$-learning (see, for example, the textbook [JORS99]).

One major criticism of the model suggested by Gold is its excessive use of memory: for each new hypothesis the entire history of past data is available. Iterative learning is the most common variant of learning in the limit which addresses memory constraints: the memory of the learner on past data is just its current hypothesis. Due to the padding lemma, this memory is still not void, but finitely many data can be memorized in the hypothesis.

There is already a quite comprehensive body of work on iterative learning [CK10, CM08, JKMS16, JMZ13, JORS99]. However, this work focuses on learning from from text, that is, from positive data only. In this paper we are also interested in the other important paradigm of learning from both positive and negative information. For example, when learning half-spaces, one could see data declaring that $\langle 1, 1\rangle$ is in the target half-space, further is $\langle 3, 2\rangle$, but $\langle 1, 10\rangle$ is not, and so on. This setting is called *learning from informant* (in contrast to learning from *text*).

Iterative learning from informant was analyzed by [JLZ07], where various natural restrictions were considered; the authors focused on the case of learning indexable families (classes of languages which are uniformly decidable). Here they showed for example that learners can be assumed to be consistent with the data just seen, but not necessarily with

[1] $\mathbf{Txt}$ stands for learning from a *text* of positive examples; $\mathbf{Ex}$ stands for *explanatory*.

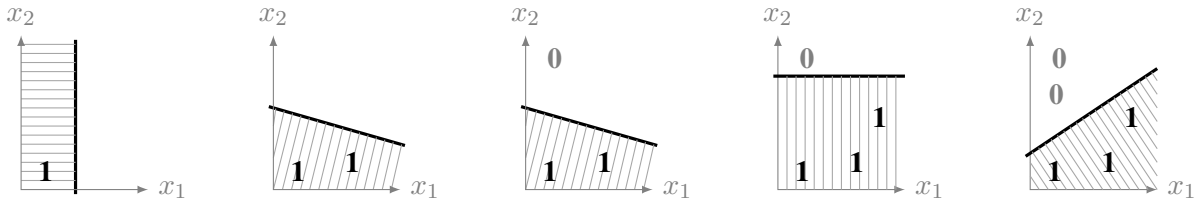

FIGURE 1. Learning Process when the hypotheses correspond to half-spaces and data is binary labeled.

all previously presented data, both for learning from text and from informant. In this paper we additionally consider learning of arbitrary classes of computably enumerable languages and of classes with only recursive languages.

In Section 2.1 we consider two restrictions on learning from informant: learning from text and learning iteratively. We show that both these restrictions render fewer classes of languages learnable; in fact, the two restrictions yield two incomparable sets of language classes being learnable, which also shows that learning iteratively from text is weaker than supposing just one of the two restrictions.

For understanding iterative learners we analyze what normal forms can be assumed about such learners in Section 2.2. First we show that, analogously to the case of learning from text (as analyzed in [CM09]), we cannot assume learners to be total (i.e. always giving an output). However, from [CM07] we know that we can assume iterative text learners to be *canny*; we adapt this normal form for the case of iterative learning from informant and show that it can be assumed to hold for iterative learners generally.

Many works focus on understanding these properties via relating different learning restrictions for the learning setting at hand; for example, [JKMS16] mapped out all pairwise relations for a group of learning restrictions for iterative learning from text. A similar map for the case of iterative learning from informant is not known, but we believe that the normal form of canniness is an important stepping stone to understand iterative learners better and determine such pairwise relations. In Section 2.3 we collect all previously known results for such a map, give more such relations and discuss which questions remain open.

We complement these structural insights with an analysis of the learnability of the language class of half-spaces in Section 3. Fundamental machine learning algorithms for supervised binary classification like support vector machines and the perceptron use half-spaces as hypothesis space. With a fixed computable kernel function even more learning tasks can be reduced to classifying with half-spaces. The learnability of linear predictors has been investigated with respect to other learning models and respective research questions, e.g. PAC-learning [Sha15], Preference-based Teaching [GRSZ17]. See [SSBD14] for an introduction to this concept class and different implemented learning algorithms. As we are concerned with computable learners, we first formalize the problem by encoding it appropriately. We then observe that the set of half-spaces forms an indexable family and is therefore learnable by enumeration from informant by a full-information learner, due to [Gol67]. Our contribution is a geometric and therefore constructive iterative learning algorithm for the family of half-spaces. The iterative learner patiently waits for data indicating that he already encountered a locking sequence. Every so-called LOCK-state directly corresponds to a half-space. In a LOCK state the learner ignores all further consistent data. Hence, our iterative learning algorithm employs the option to store data as part of the hypothesis in order to wait for helpful data and on the other hand is smart enough to know, when to stop collecting. We illustrate the algorithm in dimension 2 and give the algorithm and a complete correctness proof for arbitrary dimension in Appendix D.

We continue this paper with some mathematical preliminaries in Section 2 before discussing our results in more detail.

2. ITERATIVE LEARNING FROM INFORMANT

We let $\mathbb{N}$ denote the *natural numbers* including 0 and write ∞ for an *infinite cardinality*. Moreover, for a function f we write $\mathrm{dom}(f)$ for its *domain* and $\mathrm{ran}(f)$ for its *range*. If we deal with (a subset of) a cartesian product, we are going to refer to the *projection functions* to the first or second coordinate by pr_1 and pr_2, respectively. Further, $X^{<\omega}$ denotes the *finite sequence*s over X and X^ω stands for the *countably infinite sequence*s over X. Additionally, $X^{\leqslant\omega} := X^{<\omega} \cup X^\omega$ denotes the set of all *countably finite or infinite sequence*s over X. For every $f \in X^{\leqslant\omega}$ and $t \in \mathbb{N}$, we let $f[t] := \{(s, f(s)) \mid s < t\}$ denote the *restriction of f to t*. Finally, for sequences $\sigma, \tau \in X^{<\omega}$ their

concatenation is denoted by $\sigma^\frown\tau$ and we write $\sigma \sqsubseteq \tau$, if σ is an initial segment of τ, i.e., there is some $t \in \mathbb{N}$ such that $\sigma = \tau[t]$. Moreover, we concatenate sequences by writing them consecutively. In our setting, we typically have $X = \mathbb{N}$ or $X = \mathbb{N} \times \{0,1\}$.

As far as possible, notation and terminology on the learning theoretic side follow [OSW86] and [JORS99], whereas on the computability theoretic side we refer to [Odi99], [Rog67] and [Köt09].

A *language* L is a recursively enumerable subset of $\mathbb{N}$. A *prediction model* f is a function $f : \mathbb{N} \to \{0,1\}$. We identify subsets of $\mathbb{N}$ with their characteristic functions $\mathbb{N} \to \{0,1\}$. Hence, there is a one-one correspondence between recursive languages and recursive binary functions. We denote the characteristic function for $L \subseteq \mathbb{N}$ by f_L.

When considering binary supervised learning, the *set of all training data sequences* $\mathbb{S}$ is the set of all finite sequences

$$\sigma = ((n_0, y_0), \ldots, (n_{|\sigma|-1}, y_{|\sigma|-1}))$$

of *consistently* binary labeled natural numbers. In case of learning from positive data only, we encounter the set $\mathbb{T}$ of finite sequences $\tau = (n_0, \ldots, n_{|\tau|-1})$ of natural numbers.

In the context of language learning, [Gol67], in his seminal paper, distinguished two major different kinds of information presentation. A function

$$I : \mathbb{N} \to \mathbb{N} \times \{0,1\}$$

is an *informant for language* L, if there is a surjection $n : \mathbb{N} \to \mathbb{N}$ such that for every $t \in \mathbb{N}$ holds $I(t) = (n(t), f_L(n(t)))$. As f_L is used to label the range of n, only consistently labeled sequences result. Hence, the range of I is a complete information about L but I is free to repeat data. Moreover, for an informant I we let

$$\begin{aligned} \mathrm{pos}(I) &:= \{y \in \mathbb{N} \mid \exists x \in \mathbb{N}\colon \mathrm{pr}_1(I(x)) = y \wedge \mathrm{pr}_2(I(x)) = 1\} \text{ and} \\ \mathrm{neg}(I) &:= \{y \in \mathbb{N} \mid \exists x \in \mathbb{N}\colon \mathrm{pr}_1(I(x)) = y \wedge \mathrm{pr}_2(I(x)) = 0\} \end{aligned}$$

denote the sets of all natural numbers, about which I gives some positive or negative information, respectively.

A *text for language* L is a function $T : \mathbb{N} \to \mathbb{N} \cup \{\#\}$ with range L after removing $\#$. The symbol $\#$ is interpreted as pause symbol and added to deal with finite languages. The main difference between an informant and a text for L is that the informant tells you also that a natural number is *not* in L.

A set $\mathcal{L} = \{L_i \mid i \in \mathbb{N}\}$ of languages is called *indexable family* if there is a computer program that on input $(i, n) \in \mathbb{N}^2$ returns 1 if $n \in L_i$ and 0 otherwise. Important examples are $\mathbf{Fin}$ and $\mathbf{CoFin}$, the set of all finite subsets of $\mathbb{N}$ and the set of all complements of finite subsets of $\mathbb{N}$, respectively.

A *learner* M *from informants (texts)* is a (partial) computable function

$$M : \mathbb{S} \to \mathbb{N} \qquad (M : \mathbb{T} \to \mathbb{N})$$

with the output interpreted with respect to a prefixed hypothesis space $\mathcal{H}$.

Often the hypothesis space is an indexable class or the established W-hypothesis space defined in Subsection 2.2.

Let $\mathcal{L}$ be a collection of languages that we want to learn. We will refer to $\mathcal{L}$ as the concept class which will often be an indexable family. Further, let $\mathcal{H} = \{L_i \mid i \in \mathbb{N}\}$ with $\mathcal{L} \subseteq \mathcal{H}$ be a second collection of languages called the hypothesis space. In general we do *not* assume that for every $L \in \mathcal{L}$ there is a unique index $i \in \mathbb{N}$ with $L_i = L$. Indeed, ambiguity in the hypothesis space helps memory-resticted learners to remember data.

Let I be an informant (T be a text) for L and $\mathcal{H} = \{L_i \mid i \in \mathbb{N}\}$ a hypothesis space. A learner $M : \mathbb{S} \to \mathbb{N}$ ($M : \mathbb{T} \to \mathbb{N}$) is *successful on* I *(on* T*)* if it eventually settles on $i \in \mathbb{N}$ with $L_i = L$. This means that when receiving increasingly long finite initial segments of I (of T) as inputs, it will from some time on be correct and not change the output on longer initial segments of I (of T).

M *learns* L if it is successful on every informant I (on every text T) for L. M *learns* $\mathcal{L}$ if there is a hypothesis space $\mathcal{H}$ such that M learns every $L \in \mathcal{L}$. We denote the collection of all $\mathcal{L}$ learnable from informant (text) by $[\mathbf{InfEx}]$ ($[\mathbf{TxtEx}]$). If we fix the hypothesis space, we denote this by a subscript for $\mathbf{Ex}$.

According to [Wie76], [LZ96], [CJLZ99] a learner M is *iterative* if its output on $\sigma \in \mathbb{S}$ ($\tau \in \mathbb{T}$) only depends on the last input $\mathrm{last}(\sigma)$ and the hypothesis $M(\sigma^-)$ after observing σ without its last element $\mathrm{last}(\sigma)$. In this sense the learner forgets all prior data and can only refer to the hypothesis which resulted from this data. The collection of all $\mathcal{L}$ learnable by an iterative learner from informant (text) is denoted by $[\mathbf{ItInfEx}]$ ($[\mathbf{ItTxtEx}]$).

2.1. **Comparison with Learning from Text.** As every informant incorporates a text for the language presented, we gain $[\mathbf{ItTxtEx}] \subseteq [\mathbf{ItInfEx}]$ by ignoring negative information.

It has been observed in [OSW86] that the superfinite language class $\mathbf{Fin} \cup \{\mathbb{N}\}$ is in $[\mathbf{InfEx}]\backslash[\mathbf{ItInfEx}]$. Moreover, with $L_k = 2\mathbb{N} \cup \{2k+1\}$ and $L'_k = L_k\backslash\{2k\}$ the indexable family $\mathcal{L} = \{2\mathbb{N}\} \cup \{L_k, L'_k \mid k \in \mathbb{N}\}$ lies in $[\mathbf{TxtEx}] \cap [\mathbf{ItInfEx}]$ but not in $[\mathbf{ItTxtEx}]$. In [JORS99] the separations are witnessed by the indexable family $\{\mathbb{N}\backslash\{0\}\} \cup \{D \cup \{0\} : D \in \mathbf{Fin}\}$.

We already observed that not every indexable family is learnable by an iterative learner from informant. On the other hand, learning by enumeration makes every indexable family learnable by an iterative learner from the informants labeling all natural numbers in the canonical order, see [Gol67].

It can easily be verified that $\mathbf{CoFin} \in [\mathbf{ItInfEx}]\backslash[\mathbf{TxtEx}]$ and with the next result $[\mathbf{ItInfEx}] \perp [\mathbf{TxtEx}]$.

Lemma 2.1. *There is an indexable family in* $[\mathbf{TxtEx}]\backslash[\mathbf{ItInfEx}]$.

Proof. As there is a computable bijection between $\mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$, we can also consider subsets of $\mathbb{N} \times \mathbb{N}$ as languages. Denote by $L_{S,D} = S \times (D \cup \{0\}) \cup (\mathbb{N}\backslash S) \times (\mathbb{N}\backslash\{0\}) \subseteq \mathbb{N} \times \mathbb{N}$ the language with $D \cup \{0\}$ in all rows numbered by an $s \in S$ and $\mathbb{N}\backslash\{0\}$ in all other rows. Consider the indexable family

$$\mathcal{L} = \{L_{S,D} \mid S, D \in \mathbf{Fin}\}.$$

$\mathcal{L}$ is clearly an indexable family, as there is a computable enumaration of all pairs (S, D) where S is a finite subset of $\mathbb{N}$ and D is a finite subset of $\mathbb{N}\backslash\{0\}$. Moreover, there is a uniform procedure to check whether (n_1, n_2) is in $L_{S,D}$.

$\mathcal{F} \in [\mathbf{TxtEx}]$**:** Maintain full information at step n of the entire sequence $T[n]$ read from text. Conjecture $S' := \{x|(x,0) \in T[n]\}$ and $D' := \{y|\exists x \in S' : (x,y) \in T[n]\}$. S' will eventually converge to S as all $(x,0)$ will be received by the learner at some point for all $x \in S$. After $S' = S$, we can say that D' will also converge to D (if it has not already) because at some point all (x,y) will have been received for all $x \in S$.

$\mathcal{F} \notin [\mathbf{ItInfEx}]$**:** Suppose an iterative learner M learns $\mathcal{F}$ from informants. Let σ be a locking sequence of M for $\mathbb{N} \times (\mathbb{N}\backslash\{0\})$. Let x_0 be such that $(x_0, 0)$ does not appear in σ. Such an x_0 must exist because there are infinitely many $(x,0)$ but σ is a finite sequence. Define $D := \{y|(x_0, y) \in \mathrm{pos}(\sigma)\}$. $L := \{x_0\} \times (D \cup \{0\}) \cup (\mathbb{N}\backslash\{x_0\}) \times (\mathbb{N}\backslash\{0\})$ is then consistent with σ, so let $\sigma' \supseteq \sigma$ be a locking sequence for L. Define y_0 such that $y_0 > \max(\{0\} \cup \{y|\exists x : (x,y) \in \mathrm{pos}(\sigma') \cup \mathrm{neg}(\sigma')\})$. The element (x_0, y_0) is consistent with $\mathbb{N} \times (\mathbb{N}\backslash\{0\})$ if and only if it is labeled positively and with L if and only if it is labeled negatively. Because σ is a locking sequence for $\mathbb{N} \times (\mathbb{N}\backslash\{0\})$ and $((x_0, y_0), 1))$ is consistent with it, $M(\sigma((x_0, y_0), 1)) = M(\sigma) = e_1$ such that $W_{e_1} = \mathbb{N} \times (\mathbb{N}\backslash\{0\})$ so by iterativeness of M we have that if $\tau := \sigma((x_0, y_0), 1)(\sigma' - \sigma)$ where $\sigma' - \sigma$ is the subsequence of σ' starting after σ ends, then $M(\tau) = M(\sigma')$ meaning τ is also a locking sequence for L. This is a contradiction because if I is an informant for L, then $J := I\backslash\{((x_0, y_0), 0)\}$ is also consistent with L so for all $\ell \geqslant 0$ we have $M(\tau J[\ell]) = M(\sigma') = e_2$ such that $W_{e_2} = L$ but τJ is an informant for $L' := \{x_0\} \times (D \cup \{(x_0, y_0)\} \cup \{0\}) \cup (\mathbb{N}\backslash\{x_0\}) \times (\mathbb{N}\backslash\{0\}) \in \mathcal{F}$ and $L' \neq L$, a contradiction. □

Summing up, we know $[\mathbf{ItTxtEx}] \subsetneq [\mathbf{TxtEx}] \perp [\mathbf{ItInfEx}] \subsetneq [\mathbf{InfEx}]$.

In the following we give a procedure to generate more separating classes in $[\mathbf{TxtEx}]\backslash[\mathbf{ItInfEx}]$. With the help of the Boolean function $\mathbf{f}$ being defined in Definition 2.2 we obtain from an indexable family $\mathcal{L} \in [\mathbf{InfEx}]\backslash[\mathbf{ItInfEx}]$ an indexable family $\mathbf{f}(\mathcal{L}) \in [\mathbf{TxtEx}]\backslash[\mathbf{ItInfEx}]$.

A more general result also applicable to a selection of informants is stated in Appendix A. Also the proof of Theorem 2.3 can be found there.

The idea is to apply the Boolean function $\mathbf{f}$, defined in the following, to an indexable family, a set of informants and to a hypothesis space being a candidate to witness the learnability. With this notation we can draw conclusions from the learnability in the setting before applying $\mathbf{f}$ to the setting after applying $\mathbf{f}$ and vice versa.

Definition 2.2. We refer to the function $\mathbf{f} \colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ defined by

$$(2n \in \mathbf{f}(L) \Leftrightarrow n \in L) \wedge (2n+1 \in \mathbf{f}(L) \Leftrightarrow n \notin L)$$

as the Boolean mapping. For a set of languages $\mathcal{L}$ we define $\mathbf{f}(\mathcal{L}) = \{\mathbf{f}(L)|L \in \mathcal{L}\}$.

Note that for an indexable class $\mathcal{L}$ the image $f(\mathcal{L})$ is again an indexable class. In the following we state a Corollary of the general Theorem A.2 stated and proven in Appendix A.

Theorem 2.3. *Consider the Boolean mapping* $\mathbf{f}$ *from Definition 2.2. Let* $\mathcal{L}$ *be an indexable concept class and require that learnability is witnessed by indexable hypothesis spaces. Then* $\mathcal{L} \in [\mathbf{InfEx}]$ *implies* $\mathbf{f}(\mathcal{L}) \in [\mathbf{TxtEx}]$. *Moreover, from* $\mathbf{f}(\mathcal{L}) \in [\mathbf{ItInfEx}]$ *we can conclude* $\mathcal{L} \in [\mathbf{ItInfEx}]$.

Therefore every set of languages separating $[\mathbf{ItInfEx}]$ and $[\mathbf{InfEx}]$ yields a separating class for $[\mathbf{ItInfEx}]$ and $[\mathbf{TxtEx}]$.

2.2. **Total and Canny Learners.** For the rest of this section, without further notation, all results are understood with respect to the W-hypothesis space defined in the following. We fix a programming system φ as introduced in [RC94]. Briefly, in the φ-system, for a natural number p, we denote by φ_p the partial computable function with program code p. We also call p an *index* for W_p defined as $\mathrm{dom}(\varphi_p)$. In reference to a Blum complexity measure, for all $p, t \in \mathbb{N}$, we denote by $W_p^t \subseteq W_p$ the recursive set of all natural numbers less or equal to t, on which the machine executing p halts in at most t steps. Moreover, by s-m-n we refer to a well-known recursion theoretic observation, which gives nice finite and infinite recursion theorems, like Case's Operator Recursion Theorem **ORT**.

Let us discuss Theorem A.2 for W-indices. For, let p be such that $W_p \in \mathcal{L}$. There is an obvious mapping from an W-index q for $\mathbf{f}(W_p) \in \mathbf{f}(\mathcal{L})$ to some p' with $W_p = W_{p'}$. Unfortunately, it is not possible to map a W-index for a non-recursive W_p to a W-index for $\mathbf{f}(W_p)$.

The question whether excluding partial functions as learners, denoted by $\mathcal{R}$, makes some sets of languages unlearnable has been investigated. Allowing only total learners does not restrict full-information learning from informant and text, i.e. $[\mathcal{R}\mathbf{InfEx}] = [\mathbf{InfEx}]$ and $[\mathcal{R}\mathbf{TxtEx}] = [\mathbf{TxtEx}]$. On the other hand [CM09] showed $[\mathcal{R}\mathbf{ItTxtEx}] \subsetneq [\mathbf{ItTxtEx}]$.

We show that totality restricts iterative learning from informant.

Theorem 2.4. $[\mathbf{ItInfEx}] \backslash [\mathcal{R}\mathbf{ItInfEx}] \neq \varnothing$.

Proof. Let o be an index for $\varnothing$ and define the iterative learner M for all $\xi \in \mathbb{N} \times \{0,1\}$ by

$$M(\varnothing) = o;$$
$$h_M(h,\xi) = \begin{cases} \varphi_{\mathrm{pr}_1(\xi)}(0), & \text{else if } \mathrm{pr}_2(\xi) = 1 \text{ and } h \notin \mathrm{ran}(\mathrm{ind}); \\ h, & \text{otherwise.} \end{cases}$$

We argue that $\mathcal{L} := \{ L \subseteq \mathbb{N} \mid L \in \mathbf{ItInfEx}(M) \}$ is not learnable by a total learner from informants. Assume towards a contradiction M' is such a learner.

For a finite informant sequence σ we denote by $\overline{\sigma}$ the corresponding canonical finite informant sequence, ending with σ's datum with highest first coordinate. Then by padded ORT there are $e \in \mathbb{N}$ and a strictly increasing computable function $a : \mathbb{N}^{<\omega} \to \mathbb{N}$, such that for all $\sigma \in \mathbb{N}^{<\omega}$ and all $i \in \mathbb{N}$

$$\begin{aligned} \sigma_0 &= \varnothing; \\ \sigma_{i+1} &= \sigma_i{}^\frown \begin{cases} (a(\sigma_i),1), & \text{if } M'(\overline{\sigma_i{}^\frown(a(\sigma_i),1)}) \neq M'(\overline{\sigma_i}); \\ \varnothing, & \text{otherwise;} \end{cases} \\ W_e &= \bigcup_{i \in \mathbb{N}} \mathrm{pos}(\overline{\sigma_i}); \\ \varphi_{a(\sigma)}(x) &= \begin{cases} e, & \text{if } M'(\overline{\sigma{}^\frown(a(\sigma),1)}) \neq M'(\overline{\sigma}); \\ \mathrm{ind}_{\mathrm{pos}(\sigma)\cup\{a(\sigma)\}}, & \text{otherwise;} \end{cases} \end{aligned} \tag{1}$$

Clearly, we have $W_e \in \mathcal{L}$ and thus M' also $\mathbf{InfEx}$-learns W_e. By the $\mathbf{Ex}$-convergence there are $e', t_0 \in \mathbb{N}$, where t_0 is minimal, such that $W_{e'} = W_e$ and for all $t \geqslant t_0$ we have $M'(\bigcup_{i\in\mathbb{N}} \overline{\sigma_i}[t]) = e'$ and hence by (1) for all i with $|\overline{\sigma_i}| \geqslant t_0$

$$M'(\overline{\sigma_i{}^\frown(a(\sigma_i),1)}) = M'(\overline{\sigma_i}) = M'(\overline{\sigma_i{}^\frown(a(\sigma_i),0)}).$$

It is easy to see, that $W_e = \mathrm{pos}(\sigma_i)$ and $W_e \cup \{a(\sigma_i)\} \in \mathcal{L}$. On the other hand M' is iterative and hence does not learn W_e and $W_e \cup \{a(\sigma_i)\}$. □

The following definition is central in investigating the learning power of iterative learning from texts, see [CM07] and [JKMS16]. We transfer it to learning from informants.

Definition 2.5. A learner M from informant is called *canny* in case for every finite informant sequence σ holds

(1) if $M(\sigma)$ is defined then $M(\sigma) \in \mathbb{N}$;
(2) for every $x \in \mathbb{N}\backslash\mathrm{content}(\sigma)$ and $i \in \{0,1\}$ a mind change $M(\sigma^\frown(x,i)) \neq M(\sigma)$ implies for all finite informant sequences τ with $\sigma^\frown(x,i) \sqsubseteq \tau$ that $M(\tau^\frown(x,i)) = M(\tau)$.

Hence, the learner is canny in case it always outputs a hypotheses and no datum twice causes a mind change of the learner. We prove in Appendix B that also for learning from informant, the learner can be assumed canny and in the following only state the result.

Lemma 2.6. *For every iterative learner M, there exists a canny iterative learner N such that*

$$\mathbf{InfEx}(M) \subseteq \mathbf{InfEx}(N).$$

2.3. **Additional Requirements.** In the following we review additional properties one might require the learning process to have in order to consider it successful. For this, we employ the following notion of consistency.

As in [LZZ08] according to [BB75] and [Bār77] for $A \subseteq \mathbb{N}$ we define

$$\mathbf{Cons}(f, A) \quad :\Leftrightarrow \quad \mathrm{pos}(f) \subseteq A \ \wedge \ \mathrm{neg}(f) \subseteq \mathbb{N}\backslash A$$

and say *f is consistent with A* or *f is compatible with A*.

Learning restrictions incorporate certain desired properties of the learners' behavior relative to the information being presented. We state the definitions for learning from informant here.

Definition 2.7. Let M be a learner and I an informant. We denote by $h_t = M(I[t])$ the hypothesis of M after observing $I[t]$ and write

(1) $\mathbf{Conv}(M, I)$ ([Ang80]), if M is *conservative on I*, i.e., for all s, t with $s \leqslant t$ holds $\mathbf{Cons}(I[t], W_{h_s}) \ \Rightarrow \ h_s = h_t$.
(2) $\mathbf{Dec}(M, I)$ ([OSW82]), if M is *decisive on I*, i.e., for all r, s, t with $r \leqslant s \leqslant t$ holds $W_{h_r} = W_{h_t} \ \Rightarrow \ W_{h_r} = W_{h_s}$.
(3) $\mathbf{Caut}(M, I)$ ([OSW86]), if M is *cautious on I*, i.e., for all s, t with $s \leqslant t$ holds $\neg W_{h_t} \subsetneq W_{h_s}$.
(4) $\mathbf{WMon}(M, I)$ ([Jan91],[Wie91]), if M is *weakly monotonic on I*, i.e., for all s, t with $s \leqslant t$ holds $\mathbf{Cons}(I[t], W_{h_s}) \ \Rightarrow \ W_{h_s} \subseteq W_{h_t}$.
(5) $\mathbf{Mon}(M, I)$ ([Jan91],[Wie91]), if M is *monotonic on I*, i.e., for all s, t with $s \leqslant t$ holds $W_{h_s} \cap \mathrm{pos}(I) \subseteq W_{h_t} \cap \mathrm{pos}(I)$.
(6) $\mathbf{SMon}(M, I)$ ([Jan91],[Wie91]), if M is *strongly monotonic on I*, i.e., for all s, t with $s \leqslant t$ holds $W_{h_s} \subseteq W_{h_t}$.
(7) $\mathbf{NU}(M, I)$ ([BCM$^+$08]), if M is *non-U-shaped on I*, i.e., for all r, s, t with $r \leqslant s \leqslant t$ holds $W_{h_r} = W_{h_t} = \mathrm{pos}(I) \ \Rightarrow \ W_{h_r} = W_{h_s}$.
(8) $\mathbf{SNU}(M, I)$ ([CM11]), if M is *strongly non-U-shaped on I*, i.e., for all r, s, t with $r \leqslant s \leqslant t$ holds $W_{h_r} = W_{h_t} = \mathrm{pos}(I) \ \Rightarrow \ h_r = h_s$.
(9) $\mathbf{SDec}(M, I)$ ([KP14]), if M is *strongly decisive on I*, i.e., for all r, s, t with $r \leqslant s \leqslant t$ holds $W_{h_r} = W_{h_t} \ \Rightarrow \ h_r = h_s$.

It is easy to observe that $\mathbf{Conv}(M, I)$ implies $\mathbf{SNU}(M, I)$ and $\mathbf{WMon}(M, I)$; $\mathbf{SDec}(M, I)$ implies $\mathbf{Dec}(M, I)$ and $\mathbf{SNU}(M, I)$; $\mathbf{SMon}(M, I)$ implies $\mathbf{Caut}(M, I), \mathbf{Dec}(M, I), \mathbf{Mon}(M, I), \mathbf{WMon}(M, I)$ and finally $\mathbf{Dec}(M, I)$ and $\mathbf{SNU}(M, I)$ imply $\mathbf{NU}(M, I)$. Figure 2 includes the resulting backbone with lines from bottom to top indicating the aforementioned implications.

The text variants can be found in [JKMS16] where all pairwise relations $=$, $\subsetneq$ or $\perp$ between the sets $[\mathbf{ItTxt}\delta\mathbf{Ex}]$ (iterative learners from text) for $\delta \in \Delta$, where $\Delta = \{\mathbf{Conv}, \mathbf{Dec}, \mathbf{Caut}, \mathbf{WMon}, \mathbf{Mon}, \mathbf{SMon}, \mathbf{NU}, \mathbf{SNU}, \mathbf{SDec}\}$, are depicted. The complete map of all pairwise relations between the sets $[\mathbf{Inf}\delta\mathbf{Ex}]$ (full-information learners from

informant) for $\delta \in \Delta$ can be found in [AKS18]. For iterative learning from informants this complete map is not known. We sum up the current status in the following.

Recall the indexable family $\mathcal{L} = \{2\mathbb{N}\} \cup \{L_k, L'_k \mid k \in \mathbb{N}\}$ with $L_k = 2\mathbb{N} \cup \{2k+1\}$ and $L'_k = L_k \backslash \{2k\}$, separating $[\mathbf{ItTxtEx}]$ from $[\mathbf{TxtEx}]$. Clearly, $\mathcal{L} \in [\mathcal{R}\mathbf{ItInfConvSDecMonEx}]$. With a locking sequence argument we can observe $[\mathbf{ItInfSMonEx}] \subsetneq [\mathbf{ItInf}\delta\mathbf{Ex}]$ for all $\delta \in \Delta \backslash \{\mathbf{SMon}\}$.

If we denote by $\mathbf{Inf}_{\mathrm{can}}$ the set of all informants labelling the natural numbers according to their canonical order, we obtain $\mathbf{Fin} \cup \{\mathbb{N}\} \in [\mathcal{R}\mathbf{ItInf}_{\mathrm{can}}\mathbf{ConsConvSDecMonEx}]$ and thus in contrast to full-information learning from informant $[\mathbf{ItInf}_{\mathrm{can}}\mathbf{Ex}] \neq [\mathbf{ItInfEx}]$, see [AKS18].

Theorem 2.4 can be restated as.

Theorem 2.8. $[\mathbf{ItInfConvSDecSMonEx}] \backslash [\mathcal{R}\mathbf{ItInfEx}] \neq \varnothing$.

It has been observed that requiring a monotonic behavior of the learner is restrictive.

Theorem 2.9. [LZ92] *There exists an indexable family in* $[\mathbf{ItInfMonEx}] \subsetneq [\mathbf{ItInfEx}]$.

It is easy to see that requiring a cautious behavior of the learner is also restrictive.

Theorem 2.10. *There exists an indexable family in* $[\mathbf{ItInfCautEx}] \subsetneq [\mathbf{ItInfEx}]$.
Proof. *The indexable family* $\{\mathbb{N}\} \cup \{\mathbb{N} \backslash \{x\} \mid x \in \mathbb{N}\}$ *is clearly not cautiously learnable but conservatively, strongly decisively and monotonically learnable by a total iterative learner from informant.* □

Corollary 2.11. $[\mathbf{ItInfCautEx}] \perp [\mathbf{ItInfMonEx}]$

Moreover, requiring a conservative learning behavior is also restrictive.

Theorem 2.12. [JLZ07] *There exists an indexable family in* $[\mathbf{ItInfConvEx}] \subsetneq [\mathbf{ItInfEx}]$.

Indeed, they provide an indexable family in $[\mathbf{ItInfCautWMonNUDecEx}] \backslash [\mathbf{ItInfConvEx}]$ and an indexable family in $[\mathcal{R}\mathbf{ItTxtCautConvSDecEx}] \backslash [\mathbf{ItInfMonEx}]$.

Hence the map differs from the map on iterative learning from text in [JKMS16] as $\mathbf{Caut}$ is restrictive and also from the map of full-information learning in [AKS18] from informant as $\mathbf{Conv}$ is restrictive too. It has been open how $\mathbf{WMon}$, $\mathbf{Dec}$, $\mathbf{NU}$, $\mathbf{SDec}$ and $\mathbf{SNU}$ relate to each other and the other requirements. We show that also $\mathbf{SNU}$ restricts $\mathbf{ItInfEx}$ in Appendix C with an intricate $\mathbf{ORT}$-argument. The result reads as follows.

Theorem 2.13. $[\mathbf{ItInfSNUEx}] \subsetneq [\mathbf{ItInfEx}]$

We are now attempting to clarify in which sense precisely $\mathbf{Conv}$ is a restriction and more specifically, where exactly and how often there are separations in the implication chains $\mathbf{Conv} \Rightarrow \mathbf{WMon} \Rightarrow \mathbf{T}$, $\mathbf{Conv} \Rightarrow \mathbf{SNU} \Rightarrow \mathbf{NU} \Rightarrow \mathbf{T}$ and $\mathbf{SDec} \Rightarrow \mathbf{Dec} \Rightarrow \mathbf{NU} \Rightarrow \mathbf{T}$. In the following we provide a lemma that might help to investigate $\mathbf{WMon}$, $\mathbf{Dec}$ and $\mathbf{NU}$.

Definition 2.14. Denote the set of all unbounded and non-decreasing functions by $\mathfrak{S}$, i.e.,

$$\mathfrak{S} := \{\, \mathfrak{s} : \mathbb{N} \to \mathbb{N} \mid \forall x \in \mathbb{N}\, \exists t \in \mathbb{N} \colon \mathfrak{s}(t) \geqslant x \text{ and } \forall t \in \mathbb{N} \colon \mathfrak{s}(t+1) \geqslant \mathfrak{s}(t) \,\}.$$

Then every $\mathfrak{s} \in \mathfrak{S}$ is a so called *admissible simulating function*.

A predicate $\beta \subseteq \mathfrak{P} \times \mathcal{I}$ is *semantically delayable*, if for all $\mathfrak{s} \in \mathfrak{S}$, all $I, I' \in \mathcal{I}$ and all learners $M, M' \in \mathfrak{P}$ holds: Whenever we have $\mathrm{pos}(I'[t]) \supseteq \mathrm{pos}(I[\mathfrak{s}(t)])$, $\mathrm{neg}(I'[t]) \supseteq \mathrm{neg}(I[\mathfrak{s}(t)])$ and $W_{M'(I'[t])} = W_{M(I[\mathfrak{s}(t)])}$ for all $t \in \mathbb{N}$, from $\beta(M, I)$ we can conclude $\beta(M', I')$.

Lemma 2.15. *Let* δ *be a semantic learning restriction, i.e.* $\delta \in \{\mathbf{Caut}, \mathbf{Dec}, \mathbf{WMon}, \mathbf{Mon}, \mathbf{SMon}, \mathbf{NU}\}$. *Then* δ *is semantically delayable.*

Lemma 2.6 can be restated as follows. For the proof we refer the reader to Appendix B.

Lemma 2.16. *For every iterative learner M and every semantically delayable learning restriction δ, there exists a canny iterative learner N such that* $\mathbf{Inf}\delta\mathbf{Ex}(M) \subseteq \mathbf{Inf}\delta\mathbf{Ex}(N)$.

Similar to the situation when learning from solely positive information, see [JKMS16], we gain that all non-semantic learning restrictions have equal learning power. To obtain this, we only allow hypothesis spaces, in which all indices refer to recursive languages as we need the Σ_1-predicate stating that a datum is consistent with a hypothesis. We emphasize this by writing $\mathcal{C}$ as a subscript.

Theorem 2.17. $[\mathbf{ItInfSNUEx}_{\mathcal{C}}] = [\mathbf{ItInfConvEx}_{\mathcal{C}}] = [\mathbf{ItInfSDecEx}_{\mathcal{C}}]$

Proof. Clearly, if a learner acts conservatively or strongly decisively, it also acts strongly non-U-shapedly. In particular, $[\mathbf{ItInfConvEx}] \subseteq [\mathbf{ItInfSNUEx}]$ and $[\mathbf{ItInfSDecEx}] \subseteq [\mathbf{ItInfSNUEx}]$.

We will prove the other inclusions via two additional requirements concerning the learning behavior. First, one may require the learner to act witness-based, i.e., each mind-change is witnessed by some false negative or false positive datum. Formally, for all $r \leqslant s \leqslant t$ holds $h_r \neq h_s \Rightarrow \operatorname{pos}(I[\![s]\!]) \cap W_{h_t} \backslash W_{h_r} \neq \varnothing \vee \operatorname{neg}(I[\![s]\!]) \cap W_{h_r} \backslash W_{h_t} \neq \varnothing$. Obviously, $[\mathbf{ItInfWbEx}] \subseteq [\mathbf{ItInfConvEx}]$ and $[\mathbf{ItInfWbEx}] \subseteq [\mathbf{ItInfSDecEx}]$.

On the other hand, the learner may be strongly locking, meaning that on every informant for the language of interest L, after finite time, it settles for a hypothesis which it will not change on data consistent with L. It is easy to see that every strongly non-U-shaped successful learning process is also strongly locking. Hence, it suffices to show that every collection of languages learnable from informant by an iterative learner acting strongly locking is also learnable by an iterative witness-based learner.

Let $\mathcal{L}$ be a collection of languages learned by the iterative learner M in a strongly locking manner. Further, let f be a computable one-one function such that for all $p \in \mathbb{N}$

$$W_{f(p)} = \{x \in W_p \mid h_M(p,(x,1)) = p\} \cup \{x \notin W_p \mid h_M(p,(x,0)) \neq p\}.$$

The learner M' taking $f(M(.))$ as its conjectures is locally conservative. Moreover, M' learns $\mathcal{L}$ because M acts strongly locking.

As we are interested in a witness-based learner N, we always enlarge the guess of M' by all data witnessing a mind-change in the past. As we want N to be iterative, this is done via padding the set of witnesses to the hypothesis and a total computable function g adding this information to the hypothesis of M' as follows:

$$\begin{aligned} W_{g(\operatorname{pad}(h,\langle MC\rangle))} &= (W_h \cup \operatorname{pos}[MC]) \backslash \operatorname{neg}[MC]; \\ N(\varnothing) &= g(\operatorname{pad}(f(M(\varnothing)),\langle\varnothing\rangle)); \\ h_N(g(\operatorname{pad}(f(h),\langle MC\rangle)),\xi) &= \begin{cases} g(\operatorname{pad}(f(h),\langle MC\rangle)), & \text{if } h_M(h,\xi) = h \vee \xi \in MC; \\ g(\operatorname{pad}(f(h_M(h,\xi)),\langle MC \cup \{\xi\}\rangle)), & \text{otherwise.} \end{cases} \end{aligned}$$

Clearly, N is iterative. Further, whenever M is strongly locked on h and $W_h = L$, since MC is consistent with L, we also have $W_{g(\operatorname{pad}(f(h),\langle MC\rangle))} = L$. As N simulates M' on an informant omitting all data that already caused a mind-change beforehand, N does explanatory learn $\mathcal{L}$. As M' learns locally conservatively and by employing g, the learner N acts witness-based. □

3. Learning Half-spaces

An important concept class for many machine learning algorithms are binary classifiers given by half-spaces. We will define the language class of halfspaces, show that they from an indexable family and provide a hypothesis space and constructive algorithm making them learnable by an iterative learner from informant.

Definition 3.1 (Coding, Halfspace, $\mathcal{C}$)**.** For an integer $x \in \mathbb{Z}$ and natural number $i \in \mathbb{N}$ we write $i = \langle x \rangle$ if i is the code of x in the sense of a computable bijection with computable inverse, for example:

$\mathbb{Z}$	0	−1	1	−2	2	−3	3	−4	4	...
$\mathbb{N}$	0	1	2	3	4	5	6	7	8	...

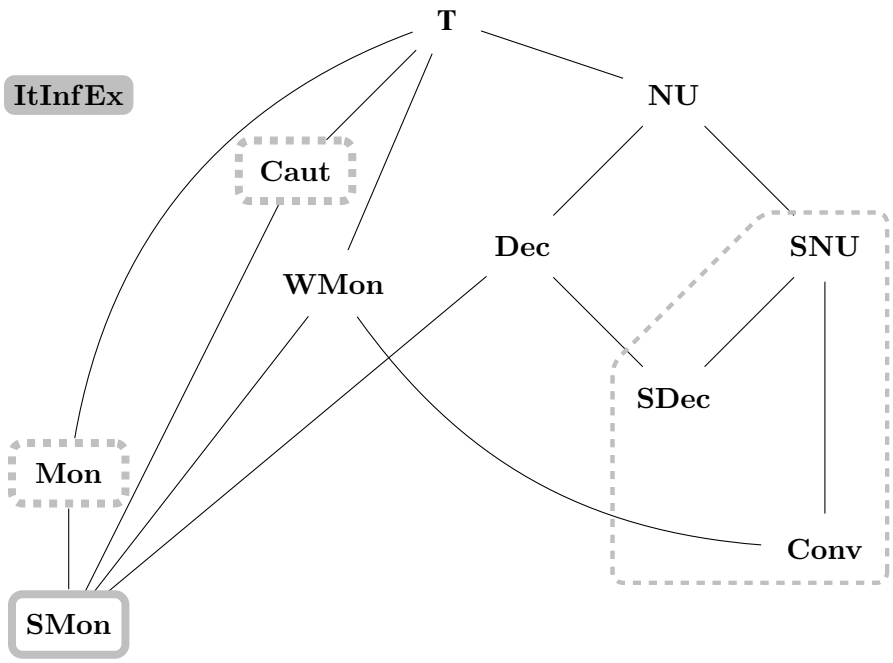

FIGURE 2. Hierarchy of delayable learning restrictions in iterative explanatory language learning from informants with C-indices. The implications independent of the learning model are depicted by black lines from bottom to top. Further, the gray outlined collections form possible equivalence classes with respect to learning power suggested due to prior research.

Moreover, for a computable bijection $\mathbb{N}\times\mathbb{N} \to \mathbb{N}$ with computable inverse, $d > 0$ and natural numbers $i, i_0, i_1, \ldots, i_d \in \mathbb{N}$ we write

$$i = \langle i_0, i_1 \rangle, \text{ if } i \text{ is the image of the vector } (i_0, i_1);$$
$$i = \langle i_0, i_1, \ldots, i_d \rangle, \text{ if } i \text{ is the image of the vector } (\langle i_0, \ldots, i_{d-1} \rangle, i_d).$$

We say that i encodes the vector (i_0, i_1) or $(i_0, i_1, \ldots, i_d)$, respectively.

Let $d > 0$. For $a_0, a_1, \ldots, a_d \in \mathbb{Z}$ the corresponding halfspace is given by

$$H_{\langle\langle a_0\rangle,\langle a_1\rangle,\ldots,\langle a_d\rangle\rangle} = \{\langle\langle x_1\rangle, \ldots, \langle x_d\rangle\rangle \mid a_0 \geqslant \sum_{i=1}^{d} a_i x_i\}.$$

Let $A = \{\langle\langle a_0\rangle, 0\rangle \mid a_0 \in \mathbb{Z}\}$ be the set of all i encoding a vector of integers $(a_0, a_1, \ldots, a_d)$ with $a_1 = \ldots = a_d = 0$. The *concept class of all halfspaces* is defined as $\mathcal{C} = \{H_i \mid i \in \mathbb{N}\backslash A\}$.

Lemma 3.2 ($\mathcal{C}$ is indexable)**.** *The concept class of halfspaces $\mathcal{C}$ is an indexable family.*

Proof. We describe the uniform decision procedure for $\mathcal{C}$. Given i and n first decode $a_0, a_1, \ldots, a_d, x_1, \ldots, x_d \in \mathbb{Z}$ such that $i = \langle\langle a_0\rangle, \langle a_1\rangle, \ldots, \langle a_d\rangle\rangle$ and $n = \langle\langle x_1\rangle, \ldots, \langle x_d\rangle\rangle$. Then check whether $a_0 \geqslant a_1 x_1 + \ldots + a_d x_d$ and return 1 if the inequality is true and 0 otherwise. □

Due to [Gol67] every indexable family is conservatively and consistently learnable by an iterative learner. Therefore, we immediately obtain.

Corollary 3.3 ($\mathcal{C} \in [\mathbf{InfEx}]$)**.** *The concept class of halfspaces $\mathcal{C}$ is learnable from informant by enumeration.*

We now state the main result of this section.

Theorem 3.4 ($\mathcal{C} \in [\mathbf{ItInfEx}]$)**.** *The concept class of halfspaces $\mathcal{C}$ is learnable by an iterative learner.*

Proof. We sketch the argument for $d = 2$ and refer the interested reader to Appendix D for a general proof. □

With the help of the following definition, we can give another uniform decision procedure for $\mathcal{H}$, to which the iterative learner will refer. This procedure allows the iterative learner to store a finite amount of information as part of its current hypothesis.

Definition 3.5 (LOCK property for $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y} \in \mathbb{Z} \times \mathbb{Z}$)**.** Let $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ lie on the two-dimensional integer grid, $\mathbb{Z} \times \mathbb{Z}$. The four points $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ *have the* LOCK*-property* if

(1) $\mathbf{u} \neq \mathbf{v}$ and $\mathbf{x} \neq \mathbf{y}$,
(2) the lines through $\mathbf{u}, \mathbf{v}$ and $\mathbf{x}, \mathbf{y}$ are parallel, in particular distinct,
(3) the lines through $\mathbf{u}, \mathbf{v}$ and $\mathbf{x}, \mathbf{y}$ are of minimal distance with respect to the integer grid, i.e. there is no parallel line passing through an integral point and strictly between them,
(4) there is a point on the line segment between $\mathbf{u}, \mathbf{v}$, such that the corresponding points with the same first/second coordinate on the line through $\mathbf{x}, \mathbf{y}$ lie on the line segment between $\mathbf{x}, \mathbf{y}$.

Note that 4. implies that

5. the minimal distance is realized between the line segments $\overline{\mathbf{uv}}$ and $\overline{\mathbf{xy}}$.

Lemma 3.6. *Let $a_1, a_2 \in \mathbb{Z}$ such that $\gcd(a_1, a_2) = 1$. Then the minimal distance between distinct lines with normal vector (a_1, a_2) passing through integral points is $\frac{1}{\sqrt{a_1^2+a_2^2}}$.*

Proof. We denote by $|a|$ the distance between a and 0, e.g. $|2| = |-2| = 2$. The minimal horizontal/vertical distances between two lines with normal vector (a_1, a_2) passing through integral points are $\frac{1}{|a_1|}$ and $\frac{1}{|a_2|}$, respectively. From this follows that the minimal distance between the lines is as claimed. □

As we encode integers and vectors (of vectors) of integers into natural numbers, we transfer the definition of the LOCK property to natural numbers.

Definition 3.7 (LOCK Property for $j \in \mathbb{N}$)**.** Let $j \in \mathbb{N}$. Extract four points $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ on the two-dimensional integer grid from j. (As $\mathbf{u} = (u_1, u_2), \ldots, \mathbf{y} = (y_1, y_2) \in \mathbb{Z} \times \mathbb{Z}$, this can be done with a repeated application of the computable inverse by assuming $j = \langle\langle\langle u_1\rangle, \langle u_2\rangle\rangle, \langle\langle v_1\rangle, \langle v_2\rangle\rangle, \langle\langle x_1\rangle, \langle x_2\rangle\rangle, \langle\langle y_1\rangle, \langle y_2\rangle\rangle\rangle$.) We say that j *has the* LOCK *property*, if $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ have the LOCK property.

We now describe the uniform decision procedure to which the iterative learner will refer.

Basically, the first coordinate of the input tells whether the learner thinks it is finished or is in data collection mode. If it thinks it is finished, it interprets the coordinate as 4 points on the integer grid. If these four points are candidates for defining the prediction model to be learned, then the decision procedure computes a halfspace from them. It then checks whether the point given by the second coordinate of the input fits the halfspace. If the four points are no valid candidates or the learner is in data collection mode, the decision procedure will treat it as a hypothesis for the upper halfplane (second coordinate $\geqslant 0$), which simply serves as a dummy hypothesis.

More formally, assume the input of the decision procedure are natural numbers $i, n \in \mathbb{N}$. If $i = 2j + 1$ for $j \in \mathbb{N}$, this is interpreted as maybe being finished. Then the procedure checks whether j has the LOCK property. If it does, the decision procedure computes a_0, a_1, a_2 for the halfspace given by $\ell_{\mathbf{u},\mathbf{v}}$, while assuming that $\mathbf{x}, \mathbf{y}$ are not in the halfspace. (For the definition of $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$, see Definition 3.7.) Next, it extracts $\mathbf{z} = (z_1, z_2) \in \mathbb{Z} \times \mathbb{Z}$ such that for the second input n holds $n = \langle\langle z_1\rangle, \langle z_2\rangle\rangle$. Finally, the procedure checks whether $a_0 \geqslant a_1 z_1 + a_2 z_2$ and returns 1 if the inequality is true. In all other cases the decision procedure returns 1 if $z_2 \geqslant 0$.

Note that for every odd number $2j + 1$, with $j \in \mathbb{N}$ having the property LOCK, the prediction model f_{2j+1} represents the unique halfspace L_{a_0,a_1,a_2} with normal vector (a_1, a_2), $\gcd(a_1, a_2) = 1$, and displacement a_0 corresponding to $\ell_{\mathbf{u},\mathbf{v}}$ and (a_1, a_2) pointing towards $\mathbf{x}, \mathbf{y}$.

Moreover, all prediction models f_i for i even or $i = 2j + 1$ with j *not* having property LOCK refer to $L_{0,0,-1} = \{\langle\langle z_1\rangle, \langle z_2\rangle\rangle \mid z_2 \geqslant 0\}$.

Now, we define the iterative learner M for $\mathcal{C}$. Initialize with 0.

If the learner is in data collection mode, check whether the stored data together with the new datum contains points $\mathbf{u}, \mathbf{v}$ positively labeled and $\mathbf{x}, \mathbf{y}$ negatively labeled with $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ having property LOCK. If not, simply add the new datum to the stored data and stay in data collection mode. If yes, switch to the maybe finished mode and store witnessing $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$.

If the learner is in maybe finished mode, i.e. its last hypothesis is $2j+1$, check whether the new datum is consistent with the halfspace corresponding to L_{2j+1}. If not, the learner switches to the data collection mode and stores

$$\langle\langle\mathbf{u}\rangle, 1\rangle, \langle\langle\mathbf{v}\rangle, 1\rangle, \langle\langle\mathbf{x}\rangle, 0\rangle, \langle\langle\mathbf{y}\rangle, 0\rangle$$

and the new datum $\sigma(|\sigma|-1)$. If yes, the learner repeats its last hypothesis $2j+1$ and therefore forgets the current datum.

Formally, M is initialized with the hypothesis 0 standing for $L_{0,0,-1}$. Let $\sigma \in \mathbb{S}$, $|\sigma| > 0$. Then σ^- denotes σ without its last element $\sigma(|\sigma|-1) = (\langle\mathbf{w}\rangle, \lambda)$.

If $M(\sigma^-) = 2j$ is even, the learner extracts from j two numbers s, w. With the interpretation of s to be w's length, it extracts from w the stored data

$$(\langle\mathbf{w}_1\rangle, \lambda_1), \ldots, (\langle\mathbf{w}_s\rangle, \lambda_s) \in \mathbb{N} \times \{0,1\}.$$

The learner now considers the set $W = \{\mathbf{w}, \mathbf{w}_1, \ldots, \mathbf{w}_s\}$. Now, if there are $\mathbf{u}, \mathbf{v} \in W$ positively labeled and $\mathbf{x}, \mathbf{y} \in W$ negatively labeled with the property LOCK, the learner outputs the hypothesis

$$2\langle\langle\mathbf{u}\rangle, \langle\mathbf{v}\rangle, \langle\mathbf{x}\rangle, \langle\mathbf{y}\rangle\rangle + 1.$$

If there are no such witnesses for the property LOCK, especially if $s < 3$, it outputs

$$2\langle s+1, \langle\langle\langle\mathbf{w}_1\rangle, \lambda_1\rangle, \ldots, \langle\langle\mathbf{w}_s\rangle, \lambda_s\rangle, \langle\langle\mathbf{w}\rangle, \lambda\rangle\rangle,$$

i.e., appends the new datum to the array of stored labeled data.

If $M(\sigma^-) = 2j+1$ is odd, the learner extracts $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y}$ from j and checks whether the new datum $(\langle\mathbf{w}\rangle, \lambda)$ is consistent with the halfspace corresponding to the four points. If not, the learner switches to data collection mode by outputting

$$2\langle 5, \langle\langle\langle\mathbf{u}\rangle, 1\rangle, \langle\langle\mathbf{v}\rangle, 1\rangle, \langle\langle\mathbf{x}\rangle, 0\rangle, \langle\langle\mathbf{y}\rangle, 0\rangle, \langle\langle\mathbf{w}\rangle, \lambda\rangle\rangle.$$

Otherwise, it repeats its last hypothesis

$$2j+1.$$

The learner converges for the following reasons:

If the learner is first locked on a halfspace with positive/negative slope, then all other slopes corresponding to locking hypotheses will be positive/negative, due to (4). This holds due to the size of the overlap of the defining positive/negative line segments of a locking hypothesis. In more detail, because a_1 and a_2 are greater or equal 1, $\frac{1}{a_1}$ is less or equal to a_2.

If the halfspace L to be learned is vertical or horizontal, the learner will never reach a locking hypothesis $2j+1$ with f_{2j+1} not corresponding to L.

Due to (3) the sequence of locking distances is strictly decreasing and bounded from below by the minimal distance corresponding to the halfspace L to be learned. Hence the learner will never lock on a hypothesis with the same corresponding normal vector (a_1, a_2) with $\gcd(a_1, a_2) = 1$ as a previously discarded locking hypothesis again and there are only finitely many choices for (a_1, a_2) due to the lower bound on the value of the distance function given by Lemma 3.6.

The learner will finally learn L because for every locking hypothesis $2j+1$ not corresponding to L, there are infinitely many positively and infinitely many negatively labeled points in $\mathbb{Z} \times \mathbb{Z}$, labeled with respect to L, and not consistent with L_{2j+1}. Hence, having discarded finitely many is not be problematic.

For every halfspace L and every informant for L, the observations immediately yield the success of the iterative learning algorithm.

Acknowledgements. We are grateful to the people supporting us. Especially, the third author thanks André Nies for pointing out the idea to study linear functions and Eugen Hellmann, Sanjay Jain, Peter Scholze, Frank Stephan and Simon Wietheger for helpful feedback regarding early forms or isolated parts of the proof for the learnability of halfspaces by this constructive iterative learner. Moreover, the first and the last author thank Vanja Doskoč and Armin Wells for helpful discussions of proof ideas for the learnability. We thank Thomas Zeugmann and Sandra Zilles for pointers to prior research.
This work was supported by DFG Grant Number KO 4635/1-1.

REFERENCES

[AKS18] M. Aschenbach, T. Kötzing, and K. Seidel. Learning from informants: Relations between learning success criteria. *arXiv preprint arXiv:1801.10502*, 2018.

[Ang80] D. Angluin. Inductive inference of formal languages from positive data. *Information and control*, 45(2):117–135, 1980.

[Bār77] J. Bārzdiņš. Inductive inference of automata, functions and programs. In *Amer. Math. Soc. Transl.*, pages 107–122, 1977.

[BB75] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[BCM+08] G. Baliga, J. Case, W. Merkle, F. Stephan, and R. Wiehagen. When unlearning helps. *Information and Computation*, 206:694–709, 2008.

[CJLZ99] J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Information and Computation*, 152:74–110, 1999.

[CK10] J. Case and T. Kötzing. Strongly non-U-shaped learning results by general techniques. In Adam Tauman Kalai and Mehryar Mohri, editors, *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 181–193. Omnipress, 2010.

[CM07] J. Case and S. Moelius. U-shaped, iterative, and iterative-with-counter learning. In N. Bshouty and C. Gentile, editors, *Proceedings of the 20th Annual Conference on Learning Theory (COLT'07)*, volume 4539 of *Lecture Notes in Artificial Intelligence*, pages 172–186, 2007.

[CM08] J. Case and S. E. Moelius. U-shaped, iterative, and iterative-with-counter learning. *Machine Learning*, 72:63–88, 2008.

[CM09] J. Case and S. Moelius. Parallelism increases iterative learning power. *Theoretical Computer Science*, 410(19):1863 – 1875, 2009.

[CM11] J. Case and S. Moelius. Optimal language learning from positive data. *Information and Computation*, 209:1293–1311, 2011.

[Gol67] E. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[GRSZ17] Z. Gao, C. Ries, H. U. Simon, and S. Zilles. Preference-based teaching. *The Journal of Machine Learning Research*, 18(1):1012–1043, 2017.

[Jan91] K. P. Jantke. Monotonic and nonmonotonic inductive inference of functions and patterns. In *Nonmonotonic and Inductive Logic, 1st International Workshop, Proc.*, pages 161–177, 1991.

[JKMS16] S. Jain, T. Kötzing, J. Ma, and F. Stephan. On the role of update constraints and text-types in iterative learning. *Information and Computation*, 247:152–168, 2016.

[JLZ07] S. Jain, S. Lange, and S. Zilles. Some natural conditions on incremental learning. *Information and Computation*, 205:1671–1684, 2007.

[JMZ13] S. Jain, S. Moelius, and S. Zilles. Learning without coding. *Theoretical Computer Science*, 473:124–148, 2013.

[JORS99] S. Jain, D. Osherson, J. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory*. MIT Press, Cambridge, Massachusetts, second edition, 1999.

[Köt09] T. Kötzing. *Abstraction and Complexity in Computational Learning in the Limit*. PhD thesis, University of Delaware, 2009.

[KP14] T. Kötzing and R. Palenta. A map of update constraints in inductive inference. In *Algorithmic Learning Theory*, pages 40–54, 2014.

[LZ92] S. Lange and T. Zeugmann. Types of monotonic language learning and their characterization. In *Proc. 5th Annual ACM Workshop on Comput. Learning Theory*, pages 377–390, New York, NY, 1992. ACM Press.

[LZ96] S. Lange and T. Zeugmann. Incremental learning from positive data. *Journal of Computer and System Sciences*, 53:88–103, 1996.

[LZZ08] S. Lange, T. Zeugmann, and S. Zilles. Learning indexed families of recursive languages from positive data: A survey. *Theoretical Computer Science*, 397(1):194–232, 2008.

[Odi99] P. Odifreddi. *Classical Recursion Theory*, volume II. Elsivier, Amsterdam, 1999.

[OSW82] D. Osherson, M. Stob, and S. Weinstein. Learning strategies. *Information and Control*, 53:32–51, 1982.

[OSW86] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, Cambridge, Mass., 1986.

[RC94] J. Royer and J. Case. *Subrecursive Programming Systems: Complexity and Succinctness*. Research monograph in *Progress in Theoretical Computer Science*. Birkhäuser Boston, 1994.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, New York, 1967. Reprinted, MIT Press, 1987.

[Sha15] O. Shamir. The sample complexity of learning linear predictors with the squared loss. *The Journal of Machine Learning Research*, 16(1):3475–3486, 2015.

[SSBD14] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Wie76] R. Wiehagen. Limes-erkennung rekursiver funktionen durch spezielle strategien. *J. Inf. Process. Cybern.*, 12 (1-2):93–99, 1976.

[Wie91] R. Wiehagen. A thesis in inductive inference. In *Nonmonotonic and Inductive Logic, 1st International Workshop, Proc.*, pages 184–207, 1991.

APPENDIX A. GENERAL RESULT ABOUT THE SEPARATION OF $\mathbf{ItInfEx}$ FROM $\mathbf{TxtEx}$

To obtain a result also applicable in other contexts, we generalize the notation. Let $\mathcal{I}$ be a set of informants (texts), for example the ones containing each information only once or infinitely often. *M learns L from $\mathcal{I}$* if it is successful on every $I \in \mathcal{I}$ for L. *M learns $\mathcal{L}$ from $\mathcal{I}$* if it learns every $L \in \mathcal{L}$ from $\mathcal{I}$. We denote the collection of all $\mathcal{L}$ learnable from $\mathcal{I}$ by $[\mathcal{I}\mathbf{Ex}]$.

The idea is to apply the Boolean function $\mathbf{f}$ to an indexable family, a set of informants and a hypothesis space possibly witnessing the learnability. With this notation we can draw conclusions from the learnability in the setting before applying $\mathbf{f}$ to the setting after applying $\mathbf{f}$ and vice versa.

Definition A.1. We refer to the function $\mathbf{f}\colon \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ defined by

$$(2n \in \mathbf{f}(L) \Leftrightarrow n \in L) \wedge (2n+1 \in \mathbf{f}(L) \Leftrightarrow n \notin L)$$

as the Boolean mapping. For a set of languages $\mathcal{L}$ we define $\mathbf{f}(\mathcal{L}) = \{\mathbf{f}(L) | L \in \mathcal{L}\}$. For an informant I for L we obtain an informant $\mathbf{f}(I)$ for $\mathbf{f}(L)$ by interweaving I_+ and I_- where

$$I_+(t) = \begin{cases} (2n_t, 1) & \text{if } I(t) = (n_t, 1); \\ (2n_t+1, 1) & \text{if } I(t) = (n_t, 0). \end{cases} \quad \text{and} \quad I_-(t) = \begin{cases} (2n_t+1, 0) & \text{if } I(t) = (n_t, 1); \\ (2n_t, 0) & \text{if } I(t) = (n_t, 0). \end{cases}$$

Moreover, the projection of I_+ to the first coordinate yields a text for $\mathbf{f}(L)$. For a set of informants $\mathcal{I}$ we define the corresponding sets of informants $\mathbf{f}(\mathcal{I})$ and texts $T_{\mathbf{f}}(\mathcal{I})$ by

$$\mathbf{f}(\mathcal{I}) := \{\mathbf{f}(I) \mid I \in \mathcal{I}\} \quad \text{and} \quad T_{\mathbf{f}}(\mathcal{I}) := \{\mathrm{pr}_1 \circ I_+ \mid I \in \mathcal{I}\}.$$

Note that for an indexable class $\mathcal{L}$ the image $f(\mathcal{L})$ is again an indexable class.

We will apply the following result to the full set of informants but state it more generally for arbitrary sets of informants $\mathcal{I}$.

Theorem A.2. *Let $\mathcal{I}$ be a set of informants, $\mathcal{L} \subseteq \{pos(I) | I \in \mathcal{I}\}$ a concept class and $\mathcal{H}$ an indexable family as suitable fixed hypothesis space. Consider the Boolean mapping $\mathbf{f}$ from Definition 2.2.*

If $\mathcal{L} \in [\mathcal{I}\mathbf{Ex}_{\mathcal{H}}]$, then $\mathbf{f}(\mathcal{L}) \in [T_{\mathbf{f}}(\mathcal{I})\mathbf{Ex}_{\mathbf{f}(\mathcal{H})}]$.

Moreover, if $\mathcal{I}$ is upwards closed with respect to the subsequence relation, then $\mathcal{L} \in [(\mathbf{It})\mathcal{I}\mathbf{Ex}_{\mathcal{H}}]$ is equivalent to $\mathbf{f}(\mathcal{L}) \in [(\mathbf{It})\mathbf{f}(\mathcal{I})\mathbf{Ex}_{\mathbf{f}(\mathcal{H})}]$.

Proof. *Let $\mathbf{f}$, $\mathcal{I}$, $\mathcal{L}$ and $\mathcal{H}$ be as stated above.*

$\mathcal{L} \in [\mathcal{I}\mathbf{Ex}_{\mathcal{H}}] \Rightarrow \mathbf{f}(\mathcal{L}) \in [T_{\mathbf{f}}(\mathcal{I})\mathbf{Ex}_{\mathbf{f}(\mathcal{H})}]$: Let M be a learner for $\mathcal{L}$ from $\mathcal{I}$. Let $\mathbf{f}(L) \in \mathbf{f}(\mathcal{L})$ and $T \in T_{\mathbf{t}}(\mathcal{I})$ a text for $\mathbf{f}(L)$. Then there is an informant $I \in \mathcal{I}$ for L such that $T = \mathrm{pr}_1 \circ I_+$. If for every $t \in \mathbb{N}$ we denote the first and second coordinate of $I(t)$ by n_t and λ_t, respectively, we obtain $T = (2n_t + 1 - \lambda_t)_{t\in\mathbb{N}}$. Therefore, we can in a computable way reconstruct $I[t]$ from $T[t]$. We define a learner M' which simulates M by $M'(T[t]) = M(I[t])$. It is easy to see that M' learns $\mathbf{f}(\mathcal{L})$ from $T_{\mathbf{f}}(\mathcal{I})$.

If $\mathcal{I}$ is upwards closed with respect to the subsequence relation, $\mathcal{L} \in [\mathbf{It}\mathcal{I}\mathbf{Ex}_{\mathcal{H}}] \Rightarrow f(\mathcal{L}) \in [\mathbf{It}\mathbf{f}(\mathcal{I})\mathbf{Ex}_{\mathbf{f}(\mathcal{H})}]$: The proof is very similar to the last paragraph. Let M be a learner for $\mathcal{L}$ from $\mathcal{I}$. Let $\mathbf{f}(L) \in \mathbf{f}(\mathcal{L})$ and $I' \in \mathbf{f}(\mathcal{I})$ an informant for $\mathbf{f}(L)$. Then there is an informant $I \in \mathcal{I}$ for L such that I' results from interweaving I_+ and I_-. We compute $\tilde{I}(t) = (\lfloor \frac{x_t}{2} \rfloor, (x_t - w_t) \mod 2)$ from $I'(t) = (x_t, w_t)$ and define M' by $M'(I'[t]) = M(\tilde{I}[t])$. Because $\tilde{I}$ contains I as a subsequence, we obtain $\tilde{I} \in \mathcal{I}$. Again, it is easily verified that M' learns $\mathbf{f}(\mathcal{L})$ from $\mathbf{f}(\mathcal{I})$. Moreover, it easy to see that M' is iterative, in case M is.

$f(\mathcal{L}) \in [\mathbf{It}\mathbf{f}(\mathcal{I})\mathbf{Ex}_{\mathbf{f}(\mathcal{H})}] \Rightarrow \mathcal{L} \in [\mathbf{It}\mathcal{I}\mathbf{Ex}_{\mathcal{H}}]$: We proceed in a similar fashion. Let M' be a learner for $\mathbf{f}(\mathcal{L})$ from $\mathbf{f}(\mathcal{I})$. Let $L \in \mathcal{L}$ and I an informant for L. We recursively construct initial segments σ_t with $|\sigma_t| = 2t$ for the informant $\mathbf{f}(I)$ for $\mathbf{f}(L)$ from I as follows: $\sigma_0 = \varnothing$; if σ_t is defined and $I(t) = (n_t, \lambda_t)$ then let $\sigma_{t+1} = \sigma_t(2n_t + 1 - \lambda_t, 1)(2n_t + \lambda_t, 0)$. Clearly, $\mathbf{f}(I) = \bigcup_{t\in\mathbb{N}} \sigma_t$. The learner $M(I[t]) = M'(\sigma_t)$ learns $\mathcal{L}$ from $\mathcal{I}$. Finally, if M' is iterative, so is M. □

If $\mathcal{I}$ is the set of all informants for $\mathcal{L}$, then $T_{\mathbf{f}}(\mathcal{I})$ is the set of all texts for $\mathbf{f}(\mathcal{L})$. $\mathbf{f}(\mathcal{I})$ is the set of all informants for $\mathbf{f}(\mathcal{L})$ that have the positive and negative informations in the order given by interweaving.

Corollary A.3. *Consider the Boolean mapping* $\mathbf{f}$ *from Definition 2.2. Then for indexable concept classes and hypothesis spaces holds:* $\mathcal{L} \in [\mathbf{InfEx}] \Rightarrow \mathbf{f}(\mathcal{L}) \in [\mathbf{TxtEx}]$*, and* $\mathcal{L} \in [\mathbf{ItInfEx}] \Leftarrow \mathbf{f}(\mathcal{L}) \in [\mathbf{ItInfEx}]$.
Proof. *For the second implication note that* $\mathbf{f}(\mathcal{L}) \in [\mathbf{ItInfEx}] \Rightarrow \mathbf{f}(\mathcal{L}) \in [\mathbf{It}\,\mathbf{f}(\mathbf{Inf})\mathbf{Ex}] \Rightarrow \mathcal{L} \in [\mathbf{ItInfEx}]$. □

Therefore every set of languages separating $[\mathbf{ItInfEx}]$ and $[\mathbf{InfEx}]$ yields a separating class for $[\mathbf{ItInfEx}]$ and $[\mathbf{TxtEx}]$.

It seems promising to generalize Theorem A.2 to additional requirements.

Appendix B. Proof of Lemmas 2.6 and 2.16

Let f be a computable 1-1 function mapping every finite informant sequence σ to a natural number encoding a program with $W_{f(\sigma)} = W_{M(\sigma)}$ if $M(\sigma) \in \mathbb{N}$ and $W_{f(\sigma)} = \varnothing$ otherwise. Clearly, σ can be reconstructed from $f(\sigma)$. We define the canny learner M' by letting

$$M'(\varnothing) = f(\varnothing)$$

$$h_{M'}(f(\sigma),(x,i)) = \begin{cases} f(\sigma^\frown(x,i)), & \text{if } x \notin \operatorname{pos}(\sigma) \cup \operatorname{neg}(\sigma) \wedge M(\sigma^\frown(x,i))\!\downarrow \neq M(\sigma)\!\downarrow; \\ f(\sigma), & \text{if } M(\sigma^\frown(x,i))\!\downarrow = M(\sigma)\!\downarrow \vee\, x \in \operatorname{content}(\sigma); \\ \uparrow, & \text{otherwise.} \end{cases}$$

M' mimics M via f on a possibly finite informant subsequence of the originally presented informant with ignoring data not causing mind changes of M or that has already caused a mind change.

Let $L \in \mathbf{InfEx}(M)$ and $I' \in \mathbf{Inf}(L)$. As M has to learn L from every informant for it, M' will always be defined. Further, let $\sigma_0 = \varnothing$ and

$$\sigma_{t+1} = \begin{cases} \sigma_t{}^\frown I'(t), & \text{if } I'(t) \notin \operatorname{ran}(\sigma_t) \wedge M(\sigma_t{}^\frown I'(t))\!\downarrow \neq M(\sigma_t)\!\downarrow; \\ \sigma_t, & \text{otherwise.} \end{cases}$$

Then by induction for all $t \in \mathbb{N}$ holds $M'(I'[t]) = f(\sigma_t)$.

The following function translates between the two settings

$$\begin{aligned} \mathfrak{r}(0) &= 0; \\ \mathfrak{r}(t+1) &= \min\{r > \mathfrak{r}(t) \mid I'(r-1) \notin \operatorname{ran}(\sigma_{\mathfrak{r}(t)})\}. \end{aligned}$$

Intuitively, the infinite range of $\mathfrak{r}$ captures all points in time r at which a datum that has not caused a mind change so far, is seen and a mind-change of M' is possible. Thus the mind change condition is of interest in order to decide whether $\sigma_{\mathfrak{r}(t+1)} \neq \sigma_{\mathfrak{r}(t)}$. Note that $\sigma_r = \sigma_{\mathfrak{r}(t)}$ for all r with $\mathfrak{r}(t) \leqslant r < \mathfrak{r}(t+1)$.

Let $I(t) = I'(\mathfrak{r}(t+1) - 1)$ for all $t \in \mathbb{N}$. Since only already observed data is ommited, I is an informant for L.

We next argue that $M(I[t]) = M(\sigma_{\mathfrak{r}(t)})$ for all $t \in \mathbb{N}$. As $I[0] = \varnothing = \sigma_0$, the claim holds for $t = 0$. Now we assume $M(I[t]) = M(\sigma_{\mathfrak{r}(t)})$ and show $M(I[t+1]) = M(\sigma_{\mathfrak{r}(t+1)})$ as follows

$$M(I[t+1]) = M(I[t]^\frown I(t)) = M(\sigma_{\mathfrak{r}(t)}{}^\frown I(t)).$$

As by the definitions of I and $\mathfrak{r}$ we have $I(t) = I'(\mathfrak{r}(t+1) - 1) \notin \operatorname{ran}(\sigma_{\mathfrak{r}(t)})$ there are two cases:

(1) If $M(\sigma_{\mathfrak{r}(t)}{}^\frown I(t)) = M(\sigma_{\mathfrak{r}(t)})$, then from $\sigma_{\mathfrak{r}(t+1)-1} = \sigma_{\mathfrak{r}(t)}$ and the definition of M' we obtain $\sigma_{\mathfrak{r}(t+1)} = \sigma_{\mathfrak{r}(t)}$. Putting both together the claimed equality $M(\sigma_{\mathfrak{r}(t)}{}^\frown I(t)) = M(\sigma_{\mathfrak{r}(t+1)})$ follows.

(2) If $M(\sigma_{\mathfrak{r}(t)}{}^\frown I(t)) \neq M(\sigma_{\mathfrak{r}(t)})$, the definition of M' yields $\sigma_{\mathfrak{r}(t+1)} = \sigma_{\mathfrak{r}(t)}{}^\frown I(t)$. Hence the claimed equality also holds in this case.

We now argue that M' explanatory learns L from I'. In order to see this, first observe $\sigma_{\mathfrak{r}(t+1)} = \sigma_{\mathfrak{r}(t)}$ if and only if $M(I'[t+1]) = M(I'[t])$ for every $t \in \mathbb{N}$. This is because

$$\begin{aligned}\sigma_{\mathfrak{r}(t+1)} = \sigma_{\mathfrak{r}(t)} &\Leftrightarrow M(\,\sigma_{\mathfrak{r}(t)}{}^\frown I(t)\,) = M(\sigma_{\mathfrak{r}(t)})\\ &\Leftrightarrow M(I[t]^\frown I(t)) = M(I[t])\\ &\Leftrightarrow M(I[t+1]) = M(I[t]).\end{aligned}$$

As I is an informant for L, the learner M explanatory learns L from I. Hence there exists some t_0 such that $W_{M(I[t_0])} = L$ and for all $t \geqslant t_0$ holds $M(I[t]) = M(I[t_0])$. With this follows $\sigma_{\mathfrak{r}(t)} = \sigma_{\mathfrak{r}(t_0)}$ for all $t \geqslant t_0$. As for every r there exists some t with $\mathfrak{r}(t) \leqslant r$ and $\sigma_r = \sigma_{\mathfrak{r}(t)}$, we obtain $\sigma_r = \sigma_{\mathfrak{r}(t_0)}$ for all $r \geqslant \mathfrak{r}(t_0)$. We conclude $M'(I'[t]) = f(\sigma_t) = f(\sigma_{\mathfrak{r}(t_0)})$ for all $t \geqslant \mathfrak{r}(t_0)$ and by the definition of f finally $W_{f(\sigma_{\mathfrak{r}(t_0)})} = W_{M(\sigma_{\mathfrak{r}(t_0)})} = W_{M(I[t_0])} = L$.

We could have added δ in front of $\mathbf{Ex}$ in the above arguments and obtain Lemma 2.16 with the following additional argument.

We define a simulating function (Definition 2.14) by

$$\mathfrak{s}(t) = \max\{s \in \mathbb{N} \mid \mathfrak{r}(s) \leqslant t\}.$$

It is easy to check that $\mathfrak{s}$ is unbounded and clearly it is non-decreasing. Then by the definitions of I and $\mathfrak{s}$ we have $\mathrm{pos}(I[\mathfrak{s}(t)]) \subseteq \mathrm{pos}(I'[\mathfrak{r}(\mathfrak{s}(t))]) \subseteq \mathrm{pos}(I'[t])$ and similarly $\mathrm{neg}(I[\mathfrak{s}(t)]) \subseteq \mathrm{neg}(I'[t])$ for all $t \in \mathbb{N}$. As $M'(I'[t]) = f(\sigma_t)$ and $M(\sigma_{\mathfrak{r}(\mathfrak{s}(t))}) = M(I[\mathfrak{s}(t)])$ for all $t \in \mathbb{N}$, in order to obtain $W_{M'(I'[t])} = W_{M(I[\mathfrak{s}(t)])}$ it suffices to show $W_{f(\sigma_t)} = W_{M(\sigma_{\mathfrak{r}(\mathfrak{s}(t))})}$. Since $W_{f(\sigma_t)} = W_{M(\sigma_t)}$ for all $t \in \mathbb{N}$, this can be concluded from $\sigma_t = \sigma_{\mathfrak{r}(\mathfrak{s}(t))}$. But this obviously holds because $\mathfrak{r}(\mathfrak{s}(t)) \leqslant t < \mathfrak{r}(\mathfrak{s}(t)+1)$ follows from the definition of $\mathfrak{s}$.

Finally, from $\delta(M, I)$ we conclude $\delta(M', I')$.

Appendix C. Proof of Theorem 2.13

Let h be a learner as follows, where the initial hypothesis is p_0, an index for $\varnothing$. We consider input data x with given label $\ell \in \{0, 1\}$.

$$\forall e, x, i : h(e, (x, \ell)) = \begin{cases} e, & \text{if } e = p_0 \wedge \ell = 0;\\ \mathrm{pad}(\varphi_x(0), x), & \text{else if } e = p_0 \wedge \ell = 1;\\ \mathrm{pad}(\varphi_y(e', x, \ell), y), & \text{else, with } e = \mathrm{pad}(e', y).\end{cases}$$

Let $\mathcal{L}$ be what h learns and suppose h' learns $\mathcal{L}$ also SNU.

We define data $a(i)$ and $b(k)$ as well as hypotheses e_0, e_1 and e_2 using ORT as follows:

$$\varphi_{a(i)}(z) = \begin{cases} e_1(k), & \text{if } z = \langle e_0, b(k), 1\rangle;\\ e_0, & \text{else if } z = 0 \vee z = \langle e_0, x, \ell\rangle;\\ e_2(k), & \text{else if } z = \langle e_1(k), a(i), 0\rangle \wedge i \geqslant k;\\ e, & \text{else if } z = \langle e, x, \ell\rangle;\end{cases}$$

$$\varphi_{b(k)}(z) = \begin{cases} e_1(k), & \text{if } z = 0;\\ e_2(k), & \text{else if } z = \langle e_1(k), a(i), 0\rangle \wedge i \geqslant k;\\ e, & \text{else if } z = \langle e, x, \ell\rangle;\end{cases}$$

Before we define W_{e_0}, $W_{e_1(k)}$ and $W_{e_2(k)}$, note that, while we see only a-data, we stick to e_0 as hypothesis. The first positive $b(k)$-datum makes us change our mind to $e_1(k)$. Any *negative* a-datum after the positive $b(k)$-datum leads to $e_2(k)$.

We give the definitions of what to list into W_{e_0}, $W_{e_1(k)}$ and $W_{e_2(k)}$ as algorithms.

$e \leftarrow$ initial hypothesis of h';
for $i = 0$ **to** ∞ **do**
  if $h'(e, (a(i), 1)) \neq e$ **then**
    $e \leftarrow h'(e, (a(i), 1))$;
    list $a(i)$ into W_{e_0};
  end
end

Algorithm 1: The definition of e_0 in the ORT-argument.

Hence in W_{e_0} we enumerate all $a(i)$ on which h' changes its mind when labeled positively. As h learns W_{e_0}, also h' has to learn it. Let I be the canonical informant for W_{e_0} and k be such that $h'(I[i]) = h'(I[k])$ for all $i \geqslant k$ and $W_{h'(I[k])} = W_{e_0}$.

Input: k;
$e \leftarrow h'(I[k](b(k), 1))$;
$i \leftarrow k$;
list $b(k)$ and the positive information in $I[k]$ into W_{e_1} and W_{e_2};
for $s = 0$ **to** ∞ **do**
  while $h'(e, (a(i), 1)) = e$ **and** $h'(e, (a(i), 0)) = e$ **do**
    list $a(i)$ into W_{e_1};
    $i \leftarrow i + 1$;
  end
  list all of what is already listed in W_{e_1} into W_{e_2};
  if $h'(e, (a(i), 1)) \neq e$ **then**
    list $a(i)$ into W_{e_1} and W_{e_2};
    $e \leftarrow h'(e, (a(i), 1))$;
  end
  else
    $j \leftarrow i$;
    $i \leftarrow i + 1$;
    while $h'(e, (a(i), 1)) = e$ **do**
      list $a(i)$ into W_{e_1} and W_{e_2};
      $i \leftarrow i + 1$;
    end
    list $a(i)$ into W_{e_1} and W_{e_2};
    list $a(j)$ into W_{e_1} and W_{e_2};
    $e \leftarrow h'(e, (a(i), 1)(a(j), 1))$;
  end
  $i \leftarrow i + 1$;
end

Algorithm 2: The definition of $e_1(k)$ and $e_2(k)$ in the ORT-argument.

We will now argue that every possible outcome of Algorithm 2 is contradictory.

(1) If all stages s are visited, then $W_{e_1(k)} = W_{e_2(k)}$ contains essentially all $a(i)$ with $i \geqslant k$. Hence h will eventually output the correct hypothesis $e_1(k)$ while h' makes infinitely many mind changes on a suitable informant I'. More precisely, the informant I' starts with $I[k](b(k), 1)$ and afterwards enumerates all $a(i)$ with $i \geqslant k$ in the order they were listed into $W_{e_1(k)}$.

(2) If the first while loop does not terminate for some stage s, then $W_{e_1(k)}$ and $W_{e_2(k)}$ are different. As $W_{e_2(k)}$ is finite, h learns it by changing its mind on some negative a-datum. On the other hand $W_{e_1(k)}$ contains all $a(i)$ with $i \geqslant k$ and h learns it by not changing its mind. Let e_{s-1} denote the current value of variable e when

entering the stage s. By the case assumption, h' does not perform a mind-change on any further positive or negative a-datum. Therefore, we must have $W_{e_1(k)} = W_{e_{s-1}} = W_{e_2(k)}$, a contradiction.

(3) If the second while loop does not terminate for some stage s, then $W_{e_1(k)} = W_{e_2(k)}$ contains all $a(i)$ with $i \geqslant k$ but $a(j_s)$. This is learned by h from any informant (though with different final hypotheses, depending on the informant). Again, we let e_{s-1} denote the current value of e when entering stage s. By the choice of k for all $j \geqslant k$ holds $h'(I[k]^\frown(a(j),1)) = h'(I[k])$ and $h'(I[k]^\frown(a(j),0)) = h'(I[k])$. Hence h' on the informant

$$I'' = I[k](a(j_s),0)(b(k),1)((a(i),1))_{i\geqslant k, i\neq j_s}$$

for $W_{e_1(k)}$ outputs e_{s-1} and therefore e_{s-1} must be correct. On the other hand e_{s-1} cannot be correct, since h' is SNU and changing its mind on the negative information $(a(j_s),0)$ in the informant

$$I''' = I[k](b(k),1)((a(i),1))_{i<j_s}(a(j_s),0)((a(i),1))_{i>j_s}$$

for $W_{e_1(k)}$.

Appendix D. Proof of Theorem 3.4

We now formaly define the concepts involved for arbitrary dimension $d > 0$.

Definition D.1. A hyperplane H in a d-dimensional space is described by an equation

$$\sum_{i=1}^{d} a_i \cdot x_i + a_0 = 0 \tag{2}$$

that is satisfied by all its points $p = (x_1, \ldots, x_d)$. In this equation $a_1, \ldots, a_d$ are called the slope coefficients and a_0 is the displacement.

Lemma D.2. *Let H be a hyperplane in a d dimensional space with rational slope coefficients, that is, any point $p = (x_1, \ldots, x_d)$ on H satisfies $\sum_{i=1}^{d} r_i \cdot x_i + r_0 = 0$ where the r_i are rational numbers. The points on H then also satisfy an equation $\sum_{i=1}^{d} a_i \cdot x_i + a_0 = 0$ where the coefficients $a_1, \ldots, a_d$ are integers such that $\gcd(a_i, \ldots, a_d) = 1$. a_0 is also an integer if and only if H passes through an integral point.*

Proof. *This is achieved by multiplying the equation $\sum_{i=1}^{d} r_i \cdot x_i + r_0 = 0$ by $\operatorname{lcm}(q_1, \ldots, q_d)$ and dividing it by $\gcd(p_1, \ldots, p_d)$ where $r_i = p_i/q_i$ is a reduced fraction meaning $\gcd(p_i, q_i) = 1$. Since $q_i \mid \operatorname{lcm}(q_1, \ldots, q_d)$ the a_i's turn out integers. To see that $\gcd(a_i, \ldots, a_d) = 1$ assume there is an integer c that divides $\frac{p_i}{\gcd(p_1,\ldots,p_d)} \cdot \frac{\operatorname{lcm}(q_1,\ldots,q_d)}{q_i}$ for all i. Because of prime decomposition, we might assume that c is prime. By definition of greatest common divisor, it can not be that $c \mid \frac{p_i}{\gcd(p_1,\ldots,p_d)}$ for all i. This means there exists a j such that $c \nmid \frac{p_j}{\gcd(p_1,\ldots,p_d)}$ so by primality of c we must have $c \mid \frac{\operatorname{lcm}(q_1,\ldots,q_d)}{q_j}$. This in turn means by the definition of least common multiple that there exists a k such that $c \mid q_k$. Now let q_l be divisible by the highest power of c. This means $c \nmid \frac{\operatorname{lcm}(q_1,\ldots,q_d)}{q_l}$ and of course that $c \mid q_l$. Since fractions were reduced we have $\gcd(p_l, q_l) = 1$ meaning $c \nmid p_l$. This implies $c \nmid \frac{p_l}{\gcd(p_1,\ldots,p_d)}$ and therefore $c \nmid \frac{p_l}{\gcd(p_1,\ldots,p_d)} \cdot \frac{\operatorname{lcm}(q_1,\ldots,q_d)}{q_l}$ contrary to assumption.*

For the last statement, note that if there are integer x_i satisfying the equation, by integrality of $a_1, \ldots, a_d$ we get that a_0 must be integer. For the converse, suppose that a_0 is an integer. Since $\gcd(a_1, \ldots, a_d) = 1$ there are by Bezout's identity integral coefficients $y_1, \ldots, y_d$ such that $\sum_{i=1}^{d} y_i \cdot a_i = 1$. Setting $x_i = a_0 \cdot y_i$ we have the desired coordinates of an integral point on the hyperplane H. □

Definition D.3. A hyperplane with defining equation $\sum_{i=1}^{d} a_i \cdot x_i + a_0 = 0$ where the coefficients $a_1, \ldots, a_d$ are integers such that $\gcd(a_i, \ldots, a_d) = 1$ is said to be in integral reduced form.

Definition D.4. The $j-$distance of a point p to a hyperplane H is the distance of p to a point q on the plane H that has all coordinates but the jth equal to those of p. If such a q does not exist the $j-$distance is undefined (or $\frac{1}{0}$).

Lemma D.5. *Let H be a hyperplane with slope coefficients a_i in integral reduced form which passes through an integral point. The smallest $j-$distance to H of an integral point not on H is equal to $1/a_j$. Furthermore, such "$j-$closest" points to H not on the hyperplane can be found on both sides of H.*

Proof. *Rewriting the defining equation for H we get for the $j-$th coordinate*

$$x_j = -\frac{1}{a_j}\left[\sum_{i=1,\neq j}^{d} a_i \cdot x_i + a_0\right]. \tag{3}$$

Define $b = \gcd(\{a_1, \ldots, a_d\}\backslash\{a_j\})$. This means that by Bezout's identity there are integers y_i such that $\sum_{i=1,\neq j}^{d} a_i \cdot y_i = m \cdot b$ for any integer multiple m of b. Since $\gcd(a_1, \ldots, a_d) = 1$ we must have $\gcd(b, a_j) = 1$, meaning there is an integer m such that $m \cdot b \overset{\mod a_j}{=} 1$ or equivalently, there exist integers m and n such that $m \cdot b = n \cdot a_j + 1$. So if the y_i were the values s.t. $\sum_{i=1,\neq j}^{d} a_i \cdot y_i = m \cdot b$, we have by setting the integer valued coordinates $x_i = (\pm 1 - a_0) \cdot y_i$ that $x_j = -\frac{1}{a_j}\left[(\pm 1 - a_0) \cdot n \cdot a_j \pm 1 - a_0 + a_0\right] = (a_0 \mp 1) \cdot n \mp \frac{1}{a_j}$. The integral points having ith coordinates x_i (in each case) for $i \neq j$ and jth coordinate equal to $(\pm a_0 - 1) \cdot n$ have $j-$distance $\frac{1}{a_j}$ to plane H on the two different sides of it. One can easily see that a smaller $j-$distance is not possible for integral points due to equation 3 for the jth coordinate of points on H. □

Lemma D.6. *Assume we have pairwise orthogonal vectors v_i for $i = 1, \ldots, d$ in a $d-$dimensional space, and let H be the hyperplane passing through the heads of these vectors when their tails are placed on the origin. Then the vector h from the origin to H and orthogonal to it is equal to $\frac{\sum_{i=1}^{d} v_i/v_i^2}{\sum_{i=1}^{d} 1/v_i^2}$.*

Proof. *By definition we must have $(v_i - h) \cdot h = 0$ for all i. This implies $|h|^2 = h \cdot v_i$ for all i. If we expand h in the basis of the v_i we have $h = (h_1, \ldots, h_d)$ and so $h_i|v_i| = |h|^2$ for all $i = 1, \ldots, d$. This means $h = |h|^2 \cdot (\frac{1}{|v_i|}, \ldots, \frac{1}{|v_d|})$. Taking the inner product with itself we get $|h|^2 = |h|^4 \cdot \sum_{i=1}^{d} 1/v_i^2 \Rightarrow |h|^2 = 1/\sum_{i=1}^{d} 1/v_i^2$ which proves the statement.* □

Corollary D.7. *The vector h as in lemma D.6 has norm $\frac{1}{\sqrt{\sum_{i=1}^{d} 1/v_i^2}}$.*

Proof. *Follows from lemma D.6.* □

Theorem D.8. *Let H be a hyperplane with integral slope coefficients a_i in integral reduced form which passes through an integral point. The closest parallel hyperplanes to it passing through different integral points have a distance of $1/\sqrt{\sum_{k=1}^{d} a_k^2}$ to it.*

Proof. *By lemma D.5 the distance along the jth axis to these hyperplanes is equal to $1/a_j$. By corollary D.7 the orthogonal distance between two closest such parallel hyperplanes will be*

$$\left(\sum_{k=1}^{d} \frac{1}{1/a_k^2}\right)^{-1/2} = \left(\sum_{k=1}^{d} a_k^2\right)^{-1/2}$$

□

Definition D.9. The integral half grid problem consists of a ground set $G_d = \mathbb{Z}^d$, the integral grid in d dimensions, and a class of half-spaces $\mathcal{L}_{Ihg}$ which consists of a half-space for every hyperplane with rational slope coefficients. For every $(r_1, \ldots, r_d, \Delta_0)$ where $r_1, \ldots, r_d \in \mathbb{Q}$ and $\Delta_0 \in \mathbb{R}$ the language $L_{(r_1,\ldots,r_d,\Delta_0)} \in \mathcal{L}_{Ihg}$ consists of all points $p = (x_1, \ldots, x_d) \in \mathbb{Z}^d$ such that $\sum_{i=1}^{d} r_i \cdot x_i + \Delta_0 \geqslant 0$. The problem is now for a learner to identify a target $L_t \in \mathcal{L}_{Ihg}$ in the limit.

Lemma D.10. *In the integral half grid problem there is a one to one correspondence between languages in $\mathcal{L}_{Ihg}$ and the elements of $\mathbb{Z}^{d+1}$. Specifically, after putting the defining equations of hyperplanes corresponding to all languages $L \in \mathcal{L}_{Ihg}$ in integral reduced form, the one to one correspondence will be between distinct languages*

(half-spaces) of $\mathcal{L}_{Ihg}$ and equivalence classes of the coefficients defined by taking the integer part of the displacements $(a_1, \ldots, a_d, \lfloor a_0 \rfloor)$. In particular, if two languages $L, L' \in \mathcal{L}_{Ihg}$ have coefficients in integral reduced form a and a' such that $a_i = a'_i$ for $1 \leqslant i \leqslant d$ and $\lfloor a_0 \rfloor = \lfloor a'_0 \rfloor$ then these two languages are identical $L = L'$.

Proof. *For any integral point $p = (x_1, \ldots, x_d)$ satisfying $\sum_{i=1}^d a_i \cdot x_i + a_0 \geqslant 0$ we may take integer parts from both sides to obtain $\sum_{i=1}^d a_i \cdot x_i + \lfloor a_0 \rfloor \geqslant 0$. Conversely, it is clear that since $\lfloor a_0 \rfloor \leqslant a_0$, that $\sum_{i=1}^d a_i \cdot x_i + \lfloor a_0 \rfloor \geqslant 0$ implies $\sum_{i=1}^d a_i \cdot x_i + a_0 \geqslant 0$.* □

Definition D.11. A basic set in d-dimensional space is a set of d affine-independent integral points, i.e. $C = \{c_0, \ldots, c_{d-1}\}$ s.t. the vectors $c_i - c_0$ for $i = 1, \ldots, d-1$ are linearly independent. The unique ($d-1$-dimensional) hyperplane H_c passing through the points of C is simply called C's hyperplane and C is a basic set for H_C. A basic cell $\mathrm{conv}(C)$ is the convex hull of points in a basic set C. Two basic sets C and C' are parallel if their hyperplanes are, they are facing each other if they are parallel and there is a line segment orthogonal to their hyperplanes meeting their cells, that is, there are points $p \in \mathrm{conv}(C)$ and $p' \in \mathrm{conv}(C')$ such that $[p, p']$ is orthgonal to H_C and $H_{C'}$. Two basic sets are adjacent if they are facing each other and their hyperplanes are distinct but as close as possible, having the distance from theorem D.8.

Lemma D.12. *Suppose a language (half-space) $L \in \mathcal{L}_{Ihg}$ is determined by a hyperplane H with coefficients a in integral reduced form such that all grid points $p = (x_1, \ldots, x_d) \in L$ satisfy $\sum_{i=1}^d a_i \cdot x_i + a_0 \geqslant 0$. We then have in addition to all grid points in L satisfying $\sum_{i=1}^d a_i \cdot x_i + \lfloor a_0 \rfloor \geqslant 0$ as stated in lemma D.10, that all grid points not contained in this halfspace $q = (y_1, \ldots, y_d) \in L^c$ satisfy $\sum_{i=1}^d a_i \cdot y_i + \lfloor a_0 \rfloor + 1 \leqslant 0$ or equivalently,*

$$\sum_{i=1}^{d} (-a_i) \cdot y_i + (-\lfloor a_0 \rfloor - 1) \geqslant 0.$$

Furthermore, both these inequalities are tight in the sense that they are satisfied with equality for elements of L and L^c respectively.

Proof. *According to lemma D.10 we must have for every $q = (y_1, \ldots, y_d) \in L^c$ that $\sum_{i=1}^d a_i \cdot y_i + \lfloor a_0 \rfloor < 0$. Since the coordinates of q are integral we have $\sum_{i=1}^d a_i \cdot y_i \in \mathbb{Z}$ and because $\mathbb{Z} \cap \mathbb{R}_{<0} = \mathbb{Z}_{\leqslant -1}$ we must have $\sum_{i=1}^d a_i \cdot y_i + \lfloor a_0 \rfloor \leqslant -1$ proving the statement.*

For the second statement, notice that the a coefficients are in integral reduced form meaning $\gcd(a_1, \ldots, a_d) = 1$ so that by Bezout's identity there are integral coordinates $(z_1, \ldots, z_d)$ such that $\sum_{i=1}^d a_i \cdot z_i = k$ for any integer $k \in \mathbb{Z}$. □

Definition D.13. For a hyperplane H described by an integral reduced form $\sum_{i=1}^d a_i \cdot x_i + a_0 = 0$ we define its positive tangent H_+ as the halfspace described by the inequality $\sum_{i=1}^d a_i \cdot x_i + \lfloor a_0 \rfloor \geqslant 0$ and and its negative tangent H_- as the halfspace described by the inequality $\sum_{i=1}^d (-a_i) \cdot x_i + (-\lfloor a_0 \rfloor - 1) \geqslant 0$.

Corollary D.14. *If a hyperplane H separates points in L from points in L^c of the integral grid which we could see as positive and negative points, the hyperplanes tangent to the positive and negative points are exactly the boundaries of H_+ and H_- as in definition D.13.*

Proof. *Follows from lemma D.12.* □

Definition D.15. We will be considering a hypothesis space consisting of sets of positive and negative data points $\mathcal{H} = \{\{(p, s) | p \in \mathbb{Z}^d, s \in \{+, -\}\}\}$. A locked state is achieved when for a hypothesis $H = \{(p, s) : p \in \mathbb{Z}^d, s \in \{+, -\}\}$ a subset C_+ of the positive points of H and a subset C_- of the negative points of H form adjacent basic sets such that all other data points retained in the hypothesis are separated based on sign by the hyperplanes of these two cells H_{C_+} and H_{C_-} meaning H_{C_+} is the boundary of a half-space $H_+^{C_+}$ and H_{C_-} is the boundary of a half-space $H_-^{C_-}$ such that $H_+ \cap H_- = \varnothing$ and for all $(p, +) \in H$ we have $p \in H_+$ and for all $(q, -) \in H$ we have $q \in H_-$. The distance of a locked state d is the distance between H_{C_+} and H_{C_-}.

Definition D.16. The violation of a locked states happens by receiving a data point (p, s) that does not respect separation by the hyperplanes of the adjacent basic sets, meaning it is on the other side of these hyperplanes than data points of the same sign as it, either $p \in H_-^{C-}$ for data point $(p, +)$ or $q \in H_+^{C+}$ for data point $(q, -)$. Remember that there are no integral points strictly between hyperplanes of adjacent basic sets by definition of their respective hyperplanes being as close as possible.

Initialize $H \leftarrow \varnothing$, $State \leftarrow Open$;
Receive new data point $(p, s) : p \in \mathbb{Z}^d, s \in \{+, -\}$;
if $State = Open$ **then**
  $H \leftarrow H \cup (p, s)$;
  if *H is a locked state* **then**
    $State \leftarrow Locked$;
    Apply convention: either do nothing or discard all previous data not required for this locked state;
  end
else if $State = Locked$ **then**
  if *p violates the locked state* **then**
    $State \leftarrow Open$;
    $H \leftarrow H \cup (p, s)$;
  end
end

Algorithm 3: Iterative learner of integral half-spaces from informants

Lemma D.17. *If d is the distance of a locked state at some point in algorithm 3 which is afterwards violated by a data point and d' is the distance of a later locked state we have $d > d'$. That is, the distance of locked states is strictly decreasing.*

Proof. *Assume H_+ and H_- are the half-spaces of the first locked state of distance d and H'_+ and H'_- are the half-spaces of the second locked state of distance d'. The sign indices indicate in both cases the signs of the data points of the corresponding basic cells. Since all data points respect the separation by the two hyperplanes in the new locked state including the points of the basic cells of the first locked state, we have the distance of any positive point and any negative point in the first locked state is at least d'. This gives us that $d \geqslant d'$ because by definition of adjacency the previous locked state had basic sets facing each other, meaning there were points p and q in the associated basic cells of distance d where p was a convex combination of positive points and q a convex combination of negative points. Since all positive points are now in H'_+ and all negative points are in H'_- the same holds for convex combinations of each label of points and thus $d \geqslant d'$. If we were to have equality $d = d'$ that would mean that the facing points p and q from the basic cells of the first locked state are situated exactly on the boundaries of H'_+ and H'_-, and because $[p, q]$ is orthogonal to the boundaries of H_+ and H_-, we must have $H_+ = H'_+$ and $H_- = H'_-$ which would contradict the first locked state ever being violated in the first place thereby proving $d \ngeqslant d'$.* □

Definition D.18. The target distance d_t is the orthogonal distance between the tangents H_+^t and H_-^t for the hyperplane H^t associated with the target language (half-space) L_t.

Lemma D.19. *The distance of any locked state is bounded from below by the target distance.*

Proof. *Similar to the proof of lemma D.17 since all data points respect separation by H_+^t and H_-^t.* □

Lemma D.20. *If the learner of algorithm 3 is in state $Open$ it will eventually go into $Locked$.*

Proof. *In the $Open$ state all incoming data points are received and aggregated and none is refused. By whatever convention for the $Locked$ state in which we may have discarded previous data points, we have two cases:*

(1) The learner eventually goes into a locked state with tangents different from that of the target's
(2) Not case 1

In the second case, assume all previously received data points (which there are finitely many of) are contained in a bounded ball B. Even if all preveious points were discareded based on convention in line 3 of algorithm 3, there will

still be infinitely many data points on H^t_+ and H^t_- further away from B which will be received and eventually create adjacent basic cells which force the learner into the Locked state with the true target tanget hyperplanes. □

Lemma D.21. *If two languages (half-spaces) $L, L' \in \mathcal{L}_{Ihg}$ are distinct, there will be grid points in their symmetric difference $L \Delta L'$ arbitrarily distant from any compact set B.*
Proof. *For this we make a case distinction:*

(1) L and L' have identical slope coefficients
(2) L and L' don't have identical slope coefficients

In the first case, distinction of the two half-spaces can only mean their displacements in integral reduced form having different integer parts. We know there exists at least one point p_0 labeled differently by the two languages. There are infinitely many integral translation vectors $\delta = (\delta_1, \ldots, \delta_d) \in \mathbb{Z}^d$ that satisfy $\sum_{i=1}^{d} a_i \cdot \delta_i = 0$ and for each one of them $p_0 + \delta$ would also be labeled differently by L and L'.

In the second case, consider the two vectors $a = (a_1, \ldots, a_d)$ and $a' = (a'_1, \ldots, a'_d)$ of the coefficients of the two half-spaces in integral reduced form. They are in integral reduced form but different which implies $a \nparallel a'$. This enables us to find an integral vector b such that $b.a$ and $b.a'$ are both nonzero and of opposite signs. W.l.o.g. assume we have a point p_0 classified by L as positive and by L' as negative and that $b.a > 0$ while $b.a' < 0$ (otherwise take $-b$). Now all points $p_0 + m \cdot b$ for $m \in \mathbb{N}$ will be classified as positive by L and negative by L'. □

Lemma D.22. *If the learner from algorithm 3 goes into a Locked state with tangent hyperplanes other than that of the target's, the Locked state will eventually be violated.*
Proof. *If all previously received data points (which there are finitely many of) are contained in a bounded ball B, there will still be infinitely many data points further away from B corresponding to the true target H^t. But by lemma D.21 any two distinct hyperplanes will label some points differently arbitrarily distant from any compact set B. Therefore, a new data point labeled inconsistently with the separation of the current Locked state will eventually be received by the learner, violating the Locked state and causing the learner to transition to state Open.* □

Lemma D.23. *The learner from algorithm 3 goes into finitely many Locked states in total.*
Proof. *By lemma D.17 the distance of locked states strictly decrease and by lemma D.19 they are bounded from below. By lemma D.8 these distances can only assume certain discrete values and the total set of combinations of the slope coefficients providing distances at least that of the target distance d^t is finite because they need to satisfy $\sum_{i=1}^{d} a_i^2 \leqslant 1/d^{t2}$.* □

Theorem D.24. *The learner from algorithm 3 identifies the target (tanget) hyperplane in a finite number of steps.*
Proof. *By lemma D.20 it will never remain in an Open state indefinitely, and by lemma D.22 it will eventually come out of any Locked state which does not correspond to the target. But by lemma D.23 the learner goes into state Locked only finitely many times, so it must eventually go into a Locked state that does correspond to the target. By algorithm 3 the hypothesis remains constant as long as the learner remains in Locked state, so if the Locked state refers to the half grid it corresponds to, algorithm 3 is able to learn the class of integral half grids in the limit.* □