On the Learnability of Vector Spaces

Valentina S. Harizanov[1]* and Frank Stephan[2]**

[1] The Department of Mathematics, The George Washington University, Funger Hall, 2201 G Street, Washington, DC, 20052, USA, Email: harizanv@gwu.edu.
[2] Mathematisches Institut, Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany, EU, Email: fstephan@math.uni-heidelberg.de.

Abstract. The central topic of the paper is the learnability of the recursively enumerable subspaces of V_∞/V, where V_∞ is the standard recursive vector space over the rationals with countably infinite dimension, and V is a given recursively enumerable subspace of V_∞. It is shown that certain types of vector spaces can be characterized in terms of learnability properties: V_∞/V is behaviourally correct learnable from text iff V is finitely dimensional, V_∞/V is behaviourally correct learnable from switching type of information iff V is finite-dimensional, 0-thin, or 1-thin. On the other hand, learnability from an informant does not correspond to similar algebraic properties of a given space. There are 0-thin spaces W_1 and W_2 such that W_1 is not explanatorily learnable from informant and the infinite product $(W_1)^\infty$ is not behaviourally correct learnable, while W_2 and the infinite product $(W_2)^\infty$ are both explanatorily learnable from informant.

1 Introduction

A central theme in inductive inference is the relation between learning from all data, that is, learning from informant, and learning from positive data only, that is, learning from text. Learning from text is much more restrictive than learning from informant, as shown by Gold [7]. Gold proved that the collection consisting of an infinite set together with all of its finite subsets can be learned from informant, but not from text. Sharma [23] showed that combining learning from informant with a restrictive convergence requirement, namely, that the first hypothesis is already the correct one, implies learnability from text, provided that the usual convergence requirement is applied and the hypothesis may be changed finitely often before converging to the correct one.

Hence it is natural to investigate what reasonable notions might exist between these two extremes. Using nonrecursive oracles as additional tools cannot completely close the gap. Even the most powerful oracles do not permit learning

* Valentina Harizanov was partially supported by the UFF grant of the George Washington University.

** Frank Stephan was supported by the Deutsche Forschungsgemeinschaft (DFG) Heisenberg grant Ste 967/1-1.

all sets [9], while the oracle K, the halting set, does for learning from informant. Restrictions on texts reduce their nonregularity and allow them to provide further information implicitly [21, 26]. Texts can be strengthened by permitting additional queries [13] to retrieve information not contained in standard texts. Ascending texts allow the learner to reconstruct complete negative information in the case of infinite sets, but might fail to do so in the case of finite sets. Thus, the class consisting of an infinite set together with all of its finite subsets remains unlearnable even for learning from ascending text.

Motoki [16] and later Baliga, Case and Jain [3] added to the positive information on the language L to be learned some, but not all, negative information They considered two ways of supplying negative data: (a) there is a finite set of negative information $S \subseteq \overline{L}$ such that the learner always succeeds in learning the set L from input S plus a text for L; (b) there is a finite set $S \subseteq \overline{L}$ such that the learner always succeeds in learning the set L from a text for L plus a text for a set H, disjoint from L and containing S, that is, satisfying $S \subseteq H \subseteq \overline{L}$. Since in case (a) one can learn all recursively enumerable sets by a single learner, the notion (b) is more interesting.

Jain and Stephan [10] treated positive and negative data symmetrically and defined notions less powerful than the ones in [3] that we discussed. The most convenient way to introduce these notions is to use the idea of a minimum adequate teacher as, for example, described by Angluin [2]. Among the learning concepts considered by Jain and Stephan [10], the following one turned out to be most important. A learner requests positive or negative data items from a teacher who has, whenever almost all requests are of the same type, to eventually reveal all information of that type.

In the present work, this type of information is applied to a natural model-theoretic setting: learning recursively enumerable subspaces of a given recursive vector space. Such a subspace is given as the quotient space of the standard recursive infinitely dimensional space over the rationals with the dependence algorithm, V_∞, and its recursively enumerable subspace V. Alternatively, this can be viewed as learning the following class of vector spaces: $\mathcal{L}(V) = \{W : V \subseteq W \subseteq V_\infty \wedge (W \text{ is recursively enumerable})\}$. This class forms a filter in the lattice $\mathcal{L}(V_\infty)$ of all recursively enumerable subspaces of V_∞. Stephan and Ventsov [25] have previously shown that, in the case of learning all ideals of a recursive ring, learnability from text has strong connections to the algebraic properties of the ring. Here, it also turns out that the two notions of learnability of the class $\mathcal{L}(V)$, from positive data or from switching type of information, have corresponding algebraic characterizations. On the other hand, we show that supplying complete information, that is, learning from informant, no longer gives such nice algebraic characterizations. The reason is that while switching type of information provides more learning power than giving positive information only, it is still much weaker than providing information from informant.

Note that some of the proofs in this version are omitted due to size constraints. You can obtain the complete paper as Forschungsberichte Mathematische Logik 55 / 2002, Mathematisches Institut, Universitaet Heidelberg, Heidelberg, 2002.

2 Preliminaries

Notions from Recursion Theory. Let $\mathbb{N}$ be the set of natural numbers. Sets are often identified with their characteristic functions, so we may write $X(n) = 1$ for $n \in X$ and $X(n) = 0$ for $n \in \overline{X}$. A subset of $\mathbb{N}$ is *recursive* if its characteristic function is recursive. A set of natural numbers is *recursively enumerable* if it is the domain of a partial recursive function or, equivalently, the range of a partial (even total) recursive function. Let $\varphi_0, \varphi_1, \varphi_2, \ldots$ be a fixed effective enumeration of all unary partial recursive functions on $\mathbb{N}$, where φ_e is computed by the Turing program with Gödel index (code) e. We write $\varphi_{e,s}(x) = y$ if $x, y, e < s$ and y is the output of $\varphi_e(x)$ in up to s steps of the Turing program with code e. For $e, s \in \mathbb{N}$, let W_e be the domain of φ_e and $W_{e,s}$ be the domain of the finite function $\varphi_{e,s}$. Then $W_0, W_1, W_2, \ldots$ is a fixed effective enumeration of all recursively enumerable subsets of $\mathbb{N}$. A Turing degree is recursively enumerable if it contains a recursively enumerable set. Let $\langle \cdot, \cdot \rangle$ be a fixed recursive 1-1 pairing function. We define the set $K = \{\langle e, x\rangle : e \in \mathbb{N} \wedge x \in W_e\}$, where $\langle e, x\rangle$ is the natural number that codes the pair (e, x). The set K is a version of the universal halting problem. It is a recursively enumerable and nonrecursive set. Its Turing degree is $\mathbf{0}'$. A Turing degree $\mathbf{a} \leq \mathbf{0}'$ is *high* if its jump has the highest possible value, that is $\mathbf{a}' = \mathbf{0}''$. A set $M \subseteq \mathbb{N}$ is called *maximal* if M is recursively enumerable and its complement $\overline{M}$ is cohesive. A set $\overline{M}$ is *cohesive* if it is infinite and there is no recursively enumerable set W such that $W \cap \overline{M}$ and $(\mathbb{N} - W) \cap \overline{M}$ are both infinite. Every maximal set has a high Turing degree. Conversely, every recursively enumerable high Turing degree contains a maximal set. This characterization was established by Martin. For more information, see [12, 22, 24].

We consider only countable algebraic structures and computable first-order languages. A countable structure for a computable language is *recursive* if its domain is recursive and its operations and relations are uniformly recursive. An example of a recursive structure is the field $(Q, +, \cdot)$ of rational numbers.

Notions from Algebra. Let $(F, +, \cdot)$ be a fixed recursive field. Then $(V_\infty, +, \cdot)$ is a computable $\aleph_0$-dimensional vector space over $(F, +, \cdot)$, consisting of all finitely nonzero infinite sequences of elements of F, under pointwise operations. Metakides and Nerode [14] showed that the study of recursive and other algorithmic vector spaces can be reduced to the study of V_∞ and its subspaces. A standard (default) basis for V_∞ is $\{\epsilon_0, \epsilon_1, \ldots\}$, where ϵ_i is the infinite sequence with the i-th term 1 and all other terms 0. Having a recursive basis is equivalent to having a recursively enumerable basis, or to the existence of a dependence algorithm. A *dependence algorithm* decides whether any finite subset of vectors is linearly dependent. If B is a basis and v a vector, then the *support* of v with respect to B is defined to be the least subset of B whose linear closure (span) contains v.

Every vector in V_∞ can be identified with its Gödel code, so the set V_∞ can be identified with $\mathbb{N}$. A subspace V of V_∞ is recursive (recursively enumerable, respectively) if its domain is a recursive (recursively enumerable, respectively) subset of the set V_∞. In that case, we also say that the quotient space V_∞/V is recursive (recursively enumerable, respectively). Let $W_0, W_1, \ldots$ be an effective enumeration of all recursively enumerable subsets of V_∞. For every e, let V_e be the vector space generated by W_e, that is, the linear closure of W_e. Then $V_0, V_1, \ldots$ is an effective enumeration of all recursively enumerable subspaces of V_∞. The set of all recursively enumerable vector subspaces of V_∞ is denoted by $\mathcal{L}(V_\infty)$. The class $\mathcal{L}(V_\infty)$, together with the operations of intersection and the direct sums of vector spaces, forms a modular nondistributive lattice. Let V be a fixed recursively enumerable subspace of V_∞. By $\mathcal{L}(V)$ we denote the lattice of all recursively enumerable spaces W such that $V \subseteq W \subseteq V_\infty$. These spaces can be viewed as representatives of the corresponding classes of recursively enumerable subspaces of V_∞/V. For more information see [19]. In the next two sections we assume that the field $(F, +, \cdot)$ is infinite. Without loss of generality, we can assume that it is $(Q, +, \cdot)$.

Vector spaces are special cases of the so-called closure systems or matroids. A matroid consists of a set X equipped with a closure operator Φ, which satisfies certain axioms. In the case of vector spaces, the closure operator is the linear closure of vector spaces. The full axiomatization of matroids will be given in Section 5.

Notions from Learning Theory. The main setting in *inductive inference* is that a learner receives more and more data on an object to be learned, and outputs a sequence of hypotheses that converges to the description of the object. In general, learning can be viewed as a dialogue between a teacher and a learner, where the learner must succeed in learning, provided the teacher satisfies a certain protocol. The formalization has two aspects: convergence behaviour and teacher constraints.

Definition 2.1 [7, 10]. A class $\mathcal{L}$ of subsets of V_∞ is learnable according to the criteria specified below iff there is a (total) recursive M, which alternately requests new data items and outputs hypotheses, and which learns every $W \in \mathcal{L}$, whenever the corresponding teacher meets the following requirements.

- All models of learning have in common that the learner makes infinitely many requests that are always either of type 0 (requesting negative data) or of type 1 (requesting positive data). The teacher answers to each request of type y, either by giving a pause symbol or a data item x such that $W(x) = y$.
- Learning from *Text*: The learner requests only information of type 1, and the teacher provides eventually all x with $W(x) = 1$.
- Learning from *Negative Text*: The learner requests only information of type 0, and the teacher provides eventually all x with $W(x) = 0$.
- Learning from *Switching Type of Information*: Whenever the learner almost always requests information of the same type y, the teacher eventually gives all x with $W(x) = y$.

- Learning from *Informant*: The learner alternately requests information of type 0 and type 1, and the teacher eventually provides every $x \in V_\infty$ after some request of the type $W(x)$.

The hypotheses output by the learner are indices in the enumeration of recursively enumerable subspaces of V_∞. Following the above protocol of the dialogue with the teacher, the learner M has to converge in one of the models below, where $e_0, e_1, \ldots$ is the infinite sequences of hypotheses output by M during the learning dialogue.

- *Explanatory* Learning: For almost every n, e_n is the same hypothesis e, which is the index of an enumeration of W.
- *Behaviourally Correct* Learning: For almost every n, the hypothesis e_n is the index of an enumeration of W, although these indices are permitted to be different (syntactically).

Txt, Sw, and Inf stand for the protocols of learning from text, switching type of information, and informant, respectively. Ex and BC stand for explanatorily and behaviourally correct learning. For example, $\mathcal{L}$ is $SwBC$-learnable iff there is a recursive M which for every $W \in \mathcal{L}$ and every teacher for W, respecting the constraints for the dialogue with the learner, outputs almost always a hypothesis for W.

Jain and Stephan [10] introduced three main notions for switching protocols. Among these three notions, the one denoted by $NewSw$ in [10] turned out to be most appropriate to model switching types of information. Since the other notions are not considered here, $NewSwEx$ and $NewSwBC$ are just denoted by $SwEx$ and $SwBC$, respectively.

Theorem 2.2. Assume that there is $W \in \mathcal{L}$ such that for every finite set D, there are $U, U' \in \mathcal{L}$ such that $U \subset W \subset U'$ and $D \cap U = D \cap W = D \cap U'$. Then $\mathcal{L}$ cannot be $SwBC$-learned.

Proof. Let M be a given $SwBC$-learner and let $\leq$ be the ordering induced by some fixed recursive 1-1 enumeration of V_∞, that is, $x \leq y$ iff x is enumerated before y or $x = y$. Then there is a teacher who knows M and might be nonrecursive doing the following.

- If the current hypothesis of M is correct and there is a finite sequence of answers of some length k, a sequence $x_1, x_2, \ldots, x_k$, corresponding to requests $y_1, y_2, \ldots, y_k$ of M, such that after the hypothesis of M is incorrect, the teacher gives x_1 from one of the shortest such sequences.
- If the current hypothesis is incorrect and y is the request, then output the least x, with respect to the ordering $\leq$, such that x has not yet appeared in any output and $W(x) = y$.
- In the remaining case, all future answers to requests consistent with W result in hypotheses for W. One considers the following two subcases.

- If U and U' have already been chosen, then take the least x, with respect to the ordering $\leq$, which has not yet appeared in the data given by the teacher, and which satisfies $x \in U$ in the case of a request of type 1, and $x \notin U'$ in the case of a request of type 0. If such x does not exist or U, U' have not been chosen, then give the pause symbol #.
- Otherwise, one chooses U, U' and gives to the learner the symbol #. Let D be the set of examples given to the learner so far. Now, one just chooses U, U' according to the condition in the statement of the Theorem: $U \subset W \subset U'$ and $D \cap U = D \cap W = D \cap U'$.

For the verification, assume that M infinitely often conjectures a false hypothesis. Then the second case applies infinitely often and the teacher gives either all elements or all nonelements of W to the corresponding requests. Otherwise, the learner ends up in the third case and eventually chooses U and U' such that $U \subset W \subset U'$. If infinitely often a data item of type 1 is required, then M sees all elements of U and some nonelements of U'. If infinitely often a data item of type 0 is required, then M sees all nonelements of U' and some elements of U. In the first case, M is expected to learn U, in the second case, M is expected to learn U'. However, by choice, M in both cases almost always conjectures the set W, and hence M does not learn $\mathcal{L}$ from switching type of information. Note that this proof holds for both criteria, $SwEx$ and $SwBC$. ▮

This condition also implies $SwBC$-nonlearnability with respect to general learners, which are not required to be recursive. The reason is that the proof does not use the fact that M is recursive, and, thus, also works for nonrecursive learners.

3 Learnability and Types of Quotient Spaces

For the criteria of learning from text and learning from switching type of information, it is possible to characterize learnability in terms of algebraic properties of the recursively enumerable subspaces of the quotient space V_∞/V.

Theorem 3.1. The following statements are equivalent for any recursively enumerable subspace $V \subseteq V_\infty$.

(a) The dimension of V_∞/V is finite.
(b) The class $\mathcal{L}(V)$ is $TxtEx$-learnable.
(c) The class $\mathcal{L}(V)$ is $TxtBC$-learnable.
(d) The class $\mathcal{L}(V)$ is $SwEx$-learnable.

Proof. (a) $\Rightarrow$ (b): Since the dimension of V_∞/V is finite, there is an algorithm that can check for every finite set $D \subset V_\infty$ and every vector x, whether x is in the linear closure of $V \cup D$. As a consequence, the following learner is recursive.

- Initially set $D = \emptyset$.
- The current hypothesis of M is always the linear closure of $V \cup D$, the hypothesis changes iff a new element is put into D.

– Whenever the teacher provides a data item x, M checks whether x is in the linear closure of $V \cup D$.
– If yes, M does not change D and, therefore, keeps the current hypothesis.
– Otherwise, M updates D to $D \cup \{x\}$ and then also updates its hypothesis.

Since the dimension of V_∞/V is finite, every space in $\mathcal{L}(V)$ is generated by $V \cup D$ for some finite set D. It is easy to verify that the algorithm finds this D in the limit. Furthermore, M makes a mind change iff the hypothesis is properly increased. Thus, M does not make more mind changes than the dimension of V_∞/V, and so the algorithm converges.

(b) $\Rightarrow$ (c) and (d): This follows directly from the definition.

(c) $\Rightarrow$ (a): Assume $V \neq V_\infty$. Let $v_0, v_1, \ldots$ be a recursive enumeration of V_∞ and U_n be the vector space generated by $V \cup \{v_0, v_1, \ldots, v_n\}$. Clearly, V_∞ is the ascending union of all spaces U_n. If follows from the basic results on learning from text [7], that the class can only be learned if this ascending chain is finite, that is, there is m with $U_n = U_m$ for all $m \geq n$. It follows that $V_\infty = U_m$ and the dimension of V_∞/V is at most $m+1$, and hence finite.

(d) $\Rightarrow$ (a): Below, the construction from Theorem 2.2 is adapted to show that V is recursive. Thus, if V_∞/V is recursive, one can find a recursive basis $\{w_0, w_1, \ldots\}$ of a vector space U with $U \cap V = \{0\}$. Let W be the linear closure of $V \cup \{w_x : x \in K\}$. The space W is not recursive, and one could now use the argument below to show that $\mathcal{L}(W)$ and, hence, also $\mathcal{L}(V)$ are not $SwEx$-learnable.

Thus, let V be a nonrecursive, but recursively enumerable subspace of V_∞. Let M be a given recursive learner and let $\leq$ be the ordering defined on V_∞ as in Theorem 2.2. There is a teacher who knows M and who does the following.

– If the current hypothesis of M is old and there is a finite sequence $x_1, x_2, \ldots, x_k$ of answers, corresponding to requests $y_1, y_2, \ldots, y_k$ of M, such that M changes its hypothesis, then the teacher gives x_1 from the shortest such sequence.
– If the current hypothesis is new, then the teacher returns on a request of a datum of type y, the least x, with respect to $\leq$, such that x has not yet been given to the learner and $V(x) = y$.

If the protocol continues, then the learning process goes infinitely often through both cases. It follows that the learner has made infinitely many hypotheses, and that the teacher has either given all elements of V on requests of type 1, or all elements of $\overline{V}$ on requests of type 0. Thus, the learner is given the required information on V, without converging syntactically. Hence, M does not $SwEx$-learn $\mathcal{L}(V)$, and so this case need not be considered.

Therefore, there is a situation without further mind changes when data consistent with V are fed continuously. One can assume that the current hypothesis is a hypothesis for V, since, otherwise, M would not learn V. Now, there are two cases.

- In every situation the learner can find itself, one can give some elements of V such that later a data item of type 0 is requested. This permits the teacher to take some recursively enumerable superspace W of V, and at every request for a datum of type 0, to feed into the learner the least $x \notin W$ that the learner had not seen so far. Then the learner does not identify the space W, although it has seen an infinite data sequence for this space.
- Otherwise, one can feed into M finitely many data consistent with V, such that M never later requests data of type 0. Let D be the set of data of type 0 seen so far. Now, one can enumerate $\overline{V}$ as follows: $x \notin V$ iff one can either continue to feed M with data from the linear closure of $V \cup \{x\}$ until a mind change occurs, or data of type 0 is requested, or some element of D is enumerated into the linear closure of $V \cup \{x\}$. This contradicts the fact that V is nonrecursive.

This completes the proof that if the dimension of V_∞/V is infinite, then $\mathcal{L}(V)$ is not $SwEx$-learnable. ∎

Metakides and Nerode [14] defined a (recursively enumerable) space $V \in \mathcal{L}(V_\infty)$ to be maximal if the dimension of V_∞/V is infinite and for every recursively enumerable space W such that $V \subseteq W \subseteq V_\infty$, we have that either V_∞/W is finitely dimensional or W/V is finitely dimensional. Metakides and Nerode used Friedberg's e-state method to construct a maximal space. Shore (see [14]) established that every maximal subset of a recursive basis of V_∞ generates a maximal subspace of V_∞.

Kalantari and Retzlaff [11] defined a space $V \in \mathcal{L}(V_\infty)$ to be supermaximal if the dimension of V_∞/V is infinite and for every recursively enumerable space $W \supseteq V$, either $W = V_\infty$ or W/V is finitely dimensional. Let $k \geq 0$ be a natural number. Kalantari and Retzlaff [11] further introduced the concept of a k-thin space, and showed its existence. A space $V \in \mathcal{L}(V_\infty)$ is k-thin if the dimension of V_∞/V is infinite and for every recursively enumerable subspace $W \supseteq V$, either the dimension of V_∞/V is $\leq k$ or the dimension of W/V is finite, and there exists $U \in \mathcal{L}(V_\infty)$ such that $U \supseteq V$ and the dimension of V_∞/U is k. Hence supermaximal spaces are also called 0-thin.

Hird [8] introduced a concept of a strongly supermaximal space. A space $V \in \mathcal{L}(V_\infty)$ is strongly supermaximal if the dimension of V_∞/V is infinite and for every recursively enumerable subset $X \subseteq V_\infty - V$, there exists a finite subset $D \subseteq V_\infty$ such that the set X is contained in the linear closure of $V \cup D$. Hird showed that strongly supermaximal spaces exist. He also established that every strongly supermaximal space is supermaximal, and that not every supermaximal space is strongly supermaximal. Downey and Hird [4] showed that strongly supermaximal spaces exist in every nonzero recursively enumerable Turing degree.

Theorem 3.2. The class $\mathcal{L}(V)$ is $SwBC$-learnable iff either V_∞/V is finite dimensional or V is 0-thin or 1-thin.

Proof. First assume that V_∞/V has infinite dimension, and that V is neither 0-thin nor 1-thin. Then there is a recursively enumerable space W such that $V \subset W \subset V_\infty$, the quotient space W/V has infinite dimension, and V_∞/W has dimension at least 2. In particular, there are vectors $x_1, x_2 \notin W$ that are linearly independent over W. Now, for every finite set D of vectors, one can choose a positive integer n such that none of the vectors in $D - W$ is in the linear closure of $W \cup \{x_1 + nx_2\}$. Furthermore, the linear closure of $V \cup (W \cap D)$ has finite dimension over V, and thus is different from W. So the condition in Theorem 2.2 is satisfied, and hence $\mathcal{L}(V)$ is not $SwBC$-learnable.

To prove the converse, we have to consider only the cases of 0-thin and 1-thin spaces, since Theorem 3.1 deals with the case when the dimension of V_∞/V is finite. In these two cases, there is a minimal space W such that $V \subseteq W$ and W/V is infinitely dimensional. Furthermore, if V is 0-thin, we have that $W = V_\infty$. If V is 1-thin, we have that $W \subset V_\infty$ and there is no other such recursively enumerable vector space U with the quotient space U/V having infinite dimension. This property allows us to give the following learning algorithm.

- The learner M requests examples of type 0 until one of them is enumerated into W. The hypothesis is V_∞ while no example of type 0 (except pause signs) has shown up, and W otherwise.
- If some example of type 0 has shown up so far, then M requests examples of type 1 and the current hypothesis is the linear closure of $V \cup E$, where E is the set of all examples of type 1 seen so far.

In the cases of V_∞ and W, the learner requests only information of type 0. If none is supplied, the hypothesis V_∞ is correct, if some examples are given, but they are all outside W, then the hypothesis W is correct. Otherwise, the vector space to be learned is the linear closure of $V \cup D$, for some finite set D. As that space cannot cover W, an example of type 0 and inside W shows up and causes that, from that time on, M only requests data of type 1. So the teacher has eventually to reveal all elements of the linear closure of $V \cup D$ and, from some time on, D is contained in the set E of current examples used in the hypothesis. ∎

4 Learning Vector Spaces from Informant

The two notions of learning from informant do not seem to have similar algebraic characterizations. In the case of 0-thin vector spaces, the class $\mathcal{L}(V)$ consists just of V_∞ and the vector spaces that are the linear closures of V together with finitely many other vectors. Nevertheless, it depends on the actual choice of V whether the class $\mathcal{L}(V)$ is $InfEx$-learnable. Furthermore, the infinite product of 0-thin spaces (as formalized in Definition 4.3) can be $InfEx$-learnable, as well as nonlearnable. Unfortunately, due to size limitations, this section had to be shortened and we just state its major results.

Theorem 4.1. There is a recursively enumerable vector space V such that $\overline{K}$, the complement of the halting problem, is uniformly enumerable relative to every recursively enumerable vector space W with $V \subseteq W \subset V_\infty$. In particular, $\mathcal{L}(V)$ is $InfEx$-learnable.

Theorem 4.2. There is a strongly supermaximal (and hence 0-thin) vector space V such that the class $\mathcal{L}(V)$ is not $InfEx$-learnable.

Definition 4.3. For $i \in \mathbb{N}$, let l_i be the linear mapping defined by $l_i(\epsilon_j) = \epsilon_{\langle i,j \rangle}$ for every $j \in \mathbb{N}$. Let V_{lp} be the linear closure of the union $l_0(V) \cup l_1(V) \cup \ldots$ of linear projections and let, as usual, $\mathcal{L}(V_{lp})$ be the class of all recursively enumerable superspaces of V_{lp}.

Theorem 4.4. There is a 0-thin vector space V such that the class $\mathcal{L}(V_{lp})$ is $InfEx$-learnable. On the other hand, there is also a 0-thin vector space W such that the class $\mathcal{L}(W_{lp})$ is even not $InfBC$-learnable.

5 Generalizing the Results

The previous results hold when the vector space V_∞ is over any infinite recursive field F. If V_∞ is over a finite recursive field, then one of the results changes, namely, the characterization of $SwBC$-learnable classes of superspaces.

Proposition 5.1. Assume that V_∞ is over a finite field F. Let V be a recursively enumerable subspace of V_∞.

(a) $\mathcal{L}(V)$ is $SwEx$-learnable iff V_∞/V is finite dimensional.
(b) $\mathcal{L}(V)$ is $SwBC$-learnable if either V_∞/V is finite dimensional or V is k-thin.
(c) $\mathcal{L}(V)$ is not $SwBC$-learnable if there is a recursively enumerable subspace W such that $V \subset W \subset V_\infty$ and the quotient spaces V_∞/W and W/V both have infinite dimension.

Proof. Part (a) follows from the proof of Theorem 3.1 since the proof does not use the fact that V is vector space over an infinite field.

Part (b), in the case when V_∞/V is finitely dimensional, is proven using the same learning algorithm as in the case of an infinite field. So let V be k-thin and let $W \in \mathcal{L}(V_\infty)$ be such that V_∞ is the closure of $W \cup \{w_1, w_2, \ldots, w_k\}$, for some $w_1, w_2, \ldots, w_k \in V_\infty$. The set U of linear combinations of $w_1, w_2, \ldots, w_k$ is finite. Now one applies the following learning algorithm.

- As long as no element of W shows up, the learner asks for negative examples and considers the set $\tilde{U}$ of those elements of U that have not appeared so far as negative examples in the data. Then the current hypothesis is the linear closure of $W \cup \tilde{U}$.
- Otherwise, that is, when some element of W had been returned as a negative example, the learner starts requesting positive examples and always conjectures the linear closure of $W \cup E$, where E is the set of positive examples seen so far.

The extension of learning $\mathcal{L}(V)$ from the case of a 1-thin space V to an arbitrary k-thin space V is based on the fact that, due to the finiteness of U, the set $\tilde{U}$ can be completely determined in the limit. The subspace being learned is generated by $W \cup \tilde{U}$, unless a negative example belonging to W is found. Besides this fact, the verification is the same as in the case of a vector space over an infinite field.

Part (c) is proven using Theorem 2.2 and the fact that the dimensions of the spaces V_∞/W and W/V are infinite. Let $\{v_0, v_1, \ldots\}$ be a linearly independent set over W, that is, no v_i is in the closure of $W \cup \{v_j : j < i\}$. Then W is the lower bound of the linear closures of the sets $W \cup \{v_k\}$, $k \in \mathbb{N}$. Moreover, W is the upper bound of the linear closures of $V \cup \{u_0, u_1, \ldots, u_i\}$, where $u_0, u_1, \ldots$ is an enumeration of W. Both sequences of spaces converge pointwise to W, from above and from below, and consist of members of $\mathcal{L}(V)$. Thus, $\mathcal{L}(V)$ is not $SwBC$-learnable by Theorem 2.2. ∎

A more general approach to learning recursively enumerable substructures of a recursive structure with the dependence relations is to consider, instead of V_∞, recursive matroids (see [5, 17, 18]). A matroid consists of a set X and of a closure operator Φ that maps the power set of X into itself and satisfies certain axioms [15]. If we consider recursive infinite matroids, without loss of generality, we can identify X with $\mathbb{N}$. The closure operator in matroids corresponds to the linear closure in vector spaces.

Definition 5.2. We say that an infinite set X and an operator Φ define a recursive matroid (X, Φ) if Φ maps recursively enumerable subsets of X to recursively enumerable subsets of X and satisfies the following axioms:

(a) If $Y \subseteq Z$ then $\Phi(Y) \subseteq \Phi(Z)$;
(b) We have that $Y \subseteq \Phi(Y)$ and $\Phi(\Phi(Y)) = \Phi(Y)$;
(c) For all $x \in \Phi(Y)$, there is a finite subset $D \subseteq Y$ with $x \in \Phi(D)$;
(d) If $Y = \Phi(Y)$ and $x, y \notin Y$ and $x \in \Phi(Y \cup \{y\})$, then $y \in \Phi(Y \cup \{x\})$;
(e) There is a recursive function f with $\Phi(W_e) = W_{f(e)}$ for all e.

The axiom (e) can sometimes be strengthened by requiring that one can compute from x and a finite set D, whether $x \in \Phi(D)$. However, this condition does not always hold for vector spaces considered here, and we are interested in a generalization of the classes $\mathcal{L}(V)$ for $V \in \mathcal{L}(V_\infty)$. Thus, we adopt only the weaker version (e). The set $\Phi(D)$ generalizes the concept of the linear closure after adding V to D, and so $x \in V$ iff $x \in \Phi(\emptyset)$. Thus the axiom (e) corresponds to the case where V is recursively enumerable, while the discussed strengthened version corresponds to V being decidable. Decidability of V is equivalent to the existence of a dependence algorithm over V, and implies recursiveness of V as a set (see [14]).

The axiom (d) given here is omitted by some authors. The following remark is a consequence of axiom (d) and shows that matroids are sufficiently similar to vector spaces. In particular, one can for submatroids Y, Z of X with $Y \subseteq Z$ introduce a dimension of Z over Y. Here, Y is a submatroid of X if $Y \subseteq X$ and $Y = \Phi(Y)$. As it is understood that the operator on the submatroid is the restriction of Φ to Y, the operator is omitted from the submatroid and the submatroid is identified with its domain Y.

Remark 5.3. Assume that Y, Z are submatroids of a matroid (X, Φ). Furthermore, assume that $Y \subseteq Z$. One says that Z/Y has dimension n if there is a set D with n elements, but not one with fewer than n elements, such that $Z = \Phi(Y \cup D)$. The space Z/Y has finite dimension iff there is such a set D of finite cardinality.

If $n > 0$, $z \in Z - Y$ and Z/Y has dimension n, then $Z/(\Phi(Y \cup \{z\}))$ has dimension $n-1$. To see this, first note that there is no set E of cardinality strictly less than $n-1$ such that $Z = \Phi(Y \cup E \cup \{z\})$, so the dimension of $Z/(\Phi(Y \cup \{z\}))$ is at least $n-1$. On the other hand, there is a maximal set $F \subset D$ such that $z \notin \Phi(Y \cup F)$. Then $d \in \Phi(Y \cup F \cup \{z\})$ for all $d \in D - F$. It follows that $Z = \Phi(Y \cup F \cup \{z\})$ and the dimension of $Z/(\Phi(Y \cup \{z\}))$ is at most the cardinality of F, in particular at most $n-1$.

A consequence is that for every submatroid Y, where X/Y has finite dimension, and for every set $U \subseteq X - Y$, there is a finite set D such that $\Phi(Y \cup D) \cap U = \emptyset$ and $\Phi(Y \cup D \cup U) = X$. Note that D is not empty iff $\Phi(Y \cup U) \subset X$.

In the following proposition, we consider the class $\mathcal{L}$ of recursively enumerable submatroids of a given matroid. Furthermore, general learners which do not have to be computable are considered. Adleman and Blum [1] proposed to measure the complexity of such learners in terms of their Turing degrees. Together with Theorem 2.2, one has that, whenever a general $SwEx$-learner exists, this learner can be chosen so that it is computable relative to the halting problem K.

Theorem 5.4. Given a matroid (X, Φ), let $\mathcal{L}$ be the class of its recursively enumerable submatroids. Relative to the halting-problem K, the following conditions are equivalent:

(a) the class $\mathcal{L}$ is $SwEx$-learnable relative to K;
(b) the class $\mathcal{L}$ is $SwBC$-learnable relative to K;
(c) the condition of Theorem 2.2 does not hold.

Proof. The part (a) $\Rightarrow$ (b) follows directly from the definition, and (b) $\Rightarrow$ (c) follows from the proof of Theorem 2.2 since it does not use the property that the learner M is recursive. So it also works with general learners which are recursive in K or even in a more powerful oracle. The part (c) $\Rightarrow$ (a) is shown using the following learning algorithm. Here Φ is defined as in the statement of the theorem, $L_i = \Phi(W_i)$ is the i-th recursively enumerable submatroid of X, and D_j is the j-th finite subset of X. Furthermore, there are recursive functions f and g such that $W_{f(i)} = \Phi(W_i)$ and $W_{g(j)} = D_j$. Let X_s consist of the first s elements of X with respect to some recursive default enumeration.

Algorithm. After having seen s input examples, let P consist of the examples of type 1 and N be the examples of type 0 seen so far. Find the first triple (i, j, pos) or (i, j, neg) from an enumeration of all triples such that the triple satisfies the following condition.

(pos) $i = g(j)$ and $\Phi(D_j) = \Phi(P)$;

(neg) $N \subseteq \overline{L_i}$ and $P \subseteq L_i$ and $X_s \subseteq \Phi(L_i \cup D_j)$ and $D_j \subseteq \Phi(L_i \cup N)$.

Output the index $f(i)$. In the case of (pos), request positive data, otherwise, request negative data.

Verification. Note that $\Phi(D_j) = \Phi(P)$ iff $D_j \subseteq \Phi(P)$ and $P \subseteq \Phi(D_j)$. So the algorithm only checks whether explicitly given finite sets are contained in some recursively enumerable sets or their complements. Thus the learner is K-recursive. Since there is a triple (i, j, pos) such that $D_j = P$ and $i = g(j)$, the search always terminates and the learner is total.

Assume that Y is the submatroid to be learned. If the dimensions of X/Y and $Y/\Phi(\emptyset)$ are both infinite, then the condition of Theorem 2.2 is satisfied since for every finite set D, there is a finite set E such that $\Phi(Y \cap D) \subset Y \subset \Phi(Y \cup E)$ and $D \cap \Phi(Y \cup E) = D \cap Y$. So Y could be approximated from below and from above by pointwise convergent series of sets different from Y. Since this cannot happen, at least one of the dimensions X/Y and $Y/\Phi(\emptyset)$ is finite.

Assume that the algorithm takes at some stage s the triple (i, j, pos). Then $L_i = \Phi(D_j) = \Phi(P)$ and the output is consistent with the input. Since $P \subseteq Y$, one has that $L_i \subseteq Y$. Furthermore, the algorithm will then continue to request positive data until some example outside L_i is seen. This happens eventually iff $Y \neq L_i$.

Assume that the algorithm takes at some stage s the triple (i, j, neg). Then the algorithm will keep this index until it either becomes clear that $X \not\subseteq \Phi(Y \cup W_j)$ or a negative example in $Y - L_i$ shows up. In the first case, D_j does not witness that the dimension of X/Y is finite, in the second case, $Y \neq L_i$.

To see that the algorithm converges, let N' and P' be the sets of all negative and positive examples seen by the learner throughout the overall running time of the algorithm. If the dimension of $\Phi(P')/\Phi(\emptyset)$ is finite, then there is a triple (i, j, pos) such that $i = g(j)$ and $\Phi(D_j) = \Phi(P')$. From some time on, enough data has been seen so that this triple will be used unless some of the finitely many triples before it is used almost always. Otherwise, the dimension of $\Phi(P')/\Phi(\emptyset)$ is infinite and the dimension of X/Y is finite. Then the dimension of $Y/\Phi(\emptyset)$ is infinite since $P' \subseteq Y$. It follows that the dimension of X/Y is finite. By Remark 5.3, there is a finite set E such that $N' \cap \Phi(Y \cup E) = \emptyset$ and $X = \Phi(Y \cup E \cup N')$. There is a triple (i, j, neg) such that $L_i = \Phi(Y \cup E)$ and $D_j \cap L_i = \emptyset$ and $X = \Phi(L_i \cup D_j)$. Again, this triple will be used almost always unless some of the finitely many triples before it is used almost always. So it follows that the algorithm converges to one triple that is used almost always. In the paragraphs of the verification preceding this one, it has been shown that whenever a triple is used almost always, then the learner converges to the correct hypothesis. ∎

Each of the $SwBC$-learnable classes of subspaces required only one switch. First, the learner observed negative examples. If these examples ruled out a fixed subspace W, then the learner switched to positive data, and never again abandoned this type of data requests. The following examples give some matroids, where infinitely many switches are required. This situation is more analogous to the general case, for which Jain and Stephan [10] showed that there is a real hierarchy of learnability, depending on the number of switches allowed.

Furthermore, the examples below show that there are classes witnessing both extremes. The class in Example 5.5 has a recursive $SwEx$-learner, while every $SwBC$-learner and thus also every $SwEx$-learner of the class in Example 5.6 needs a K-oracle.

Example 5.5. There is a recursive matroid such that every recursively enumerable submatroid is either finite or cofinite. The class of these submatroids can be $SwEx$-learned, but not with any bound on the number of switches.

Proof. Let $X = \mathbb{N}$. Let A be a maximal subset of X [20, Definition III.4.13]. For any given set Y, let

$$\Phi(Y) = \{x : (\exists y \in Y)\,[y = x \vee (y < x \wedge \{y, y+1, \ldots, x-1\} \subseteq A) \\ \vee (y > x \wedge \{x, x+1, \ldots, y-1\} \subseteq A)\,]\,\}.$$

Clearly, $\Phi(Y)$ is recursively enumerable whenever Y is.

Let $a_0, a_1, \ldots$ be the ascending (and nonrecursive) enumeration of all elements of $\overline{A}$. Let $A_0 = \{0, 1, \ldots, a_0\}$ and $A_n = \{a_{n-1}+1, a_{n-1}+2, \ldots, a_n\}$ for any $n > 0$. Then $\Phi(Y)$ is the union of all sets A_n that meet Y. If Y is finite, so is $\Phi(Y)$. If Y is infinite, then Y meets infinitely many sets A_n; in particular, $\Phi(Y) - A$ is infinite. If Y is in addition recursively enumerable, then the maximality of A implies that $\Phi(Y)$ contains almost all numbers a_n. For these n, the corresponding sets A_n are subsets of $\Phi(Y)$. In particular, $\Phi(Y)$ is cofinite. So whenever Y is recursively enumerable, $\Phi(Y)$ is either finite or cofinite.

Thus, one can $SwEx$-learn all recursively enumerable submatroids, by $SwEx$-learning all finite and cofinite sets, as it is done in [10]. ∎

Example 5.6. There is a recursive matroid whose class of all recursively enumerable submatroids can be $SwBC$-learned with the help of an K-oracle but not with the help of any oracle $A \not\geq_T K$.

References

1. Lenny Adleman and Manuel Blum. Inductive inference and unsolvability. *The Journal of Symbolic Logic*, 56(3):891–900, 1991.
2. Dana Angluin. Learning regular sets from queries and counter-examples. *Information and Computation*, 75:87–106, 1987.
3. Ganesh Baliga, John Case and Sanjay Jain. Language learning with some negative information. *Journal of Computer and System Sciences*, 51(5):273–285, 1995.
4. Rod G. Downey and Geoffrey R. Hird, Automorphisms of supermaximal spaces. *The Journal of Symbolic Logic*, 50(1):1–9, 1985.
5. Rod G. Downey and Jeffrey B. Remmel, Computable algebras and closure systems: coding properties. In: *Handbook of Recursive Mathematics*, volume 2, pages 977–1039. Elsevier, Amsterdam, 1998.
6. Lance Fortnow, William Gasarch, Sanjay Jain, Efim Kinber, Martin Kummer, Stuart Kurtz, Mark Pleszkoch, Theodore Slaman, Robert Solovay and Frank Stephan. Extremes in the degrees of inferability. *Annals of Pure and Applied Logic*, 66:231–276, 1994.

7. E. Mark Gold. Language identification in the limit. *Information and Control*, 10: 447–474, 1967.
8. Geoffrey R. Hird. Recursive properties of relations on models. *Annals of Pure and Applied Logic*, 63:241–269, 1993.
9. Sanjay Jain and Arun Sharma. On the non-existence of maximal inference degrees for language identification. *Information Processing Letters*, 47:81–88, 1993.
10. Sanjay Jain and Frank Stephan. Learning by switching type of information. In: *Algorithmic Learning Theory: Twelfth International Conference (ALT 2001)*, volume 2225 of *Lecture Notes in Artificial Intelligence*, pages 205–218. Springer-Verlag, Heidelberg, 2001.
11. Iraj Kalantari and Allen Retzlaff. Maximal vector spaces under automorphisms of the lattice of recursively enumerable vector spaces. *The Journal of Symbolic Logic*, 42(4):481–491, 1977.
12. Michael Machtey and Paul Young. *An Introduction to the General Theory of Algorithms.* North Holland, New York, 1978.
13. Wolfgang Merkle and Frank Stephan. Refuting learning revisited. Technical Report Forschungsberichte Mathematische Logik 52/2001, Mathematisches Institut, Universität Heidelberg, 2001.
14. George Metakides and Anil Nerode, Recursively enumerable vector spaces. *Annals of Mathematical Logic*, 11:147–171, 1977.
15. George Metakides and Anil Nerode, Recursion theory on fields and abstract dependence. *Journal of Algebra*, 65:36–59, 1980.
16. Tatsuya Motoki. Inductive inference from all positive and some negative data. *Information Processing Letters*, 39(4):177–182, 1991.
17. Anil Nerode and Jeffrey Remmel, Recursion theory on matroids. In: *Patras Logic Symposium, Studies in Logic and the Foundations of Mathematics*, volume 109, pages 41–65. North-Holland, 1982.
18. Anil Nerode and Jeffrey Remmel, Recursion theory on matroids II. In: *Southeast Asian Conference on Logic*, pages 133–184. North-Holland, 1983.
19. Anil Nerode and Jeffrey Remmel, A survey of lattices of r.e. substructures. In: *Proceedings of Symposia in Pure Mathematics*, volume 42, pages 323–375. American Mathematical Society, 1985.
20. Piergiorgio Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.
21. Daniel Osherson, Michael Stob and Scott Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, 1986.
22. Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted, MIT Press, 1987.
23. Arun Sharma. A note on batch and incremental learnability. *Journal of Computer and System Sciences*, 56(3):272–276, 1998.
24. Robert I. Soare. *Recursively Enumerable Sets and Degrees.* Springer-Verlag, Heidelberg, 1987.
25. Frank Stephan and Yuri Ventsov. Learning Algebraic Structures from Text using Semantical Knowledge. *Theoretical Computer Science – Series A*, 268:221-273, 2001.
26. Rolf Wiehagen. Identification of formal languages. In: *Mathematical Foundations of Computer Science*, volume 53 of *Lecture Notes in Computer Science*, pages 571–579. Springer-Verlag, 1977.
27. C. E. M. Yates. Three theorems on the degree of recursively enumerable degrees. *Duke Mathematical Journal*, 32:461–468, 1965.