Learning for Families of Algebraic Structures

Nikolay Bazhenov

Sobolev Institute of Mathematics, Novosibirsk, Russia

International Conference on Algebra,
Analysis and Geometry 2021

Kazan, Russia
August 24, 2021

Introduction: Algorithmic learning for languages

Gold in his paper “Language Identification in the Limit” [Information and Control, 10 (1967), 447–474] writes:

«I wish to construct a precise model for the intuitive notion “able to speak a language” in order to be able to investigate theoretically how it can be achieved artificially.

...an artificial intelligence which is designed to speak English will have to learn its rules from implicit information. That is, its information will consist of examples of the use of English and/or of an informant who can state whether a given usage satisfies certain rules of English, but cannot state these rules explicitly.»

The question of how children learn languages was one of the sources of motivation for Gold’s work.

Learning from examples (Gold 1967)

A *language* is a set of finite strings over some fixed finite alphabet. The *target language* is the language that has to be learned.

- Languages have to be learned from some implicit information – i.e. *examples*.

Learning from examples (Gold 1967)

A *language* is a set of finite strings over some fixed finite alphabet. The *target language* is the language that has to be learned.

- Languages have to be learned from some implicit information – i.e. *examples*.
- A *learner* is allowed to revise their hypothetical conjectures about the target language from time to time.
  So, a learner receives step-by-step more and more examples, and at each step they can revise their conjecture.

Learning from examples (Gold 1967)

A *language* is a set of finite strings over some fixed finite alphabet. The *target language* is the language that has to be learned.

- Languages have to be learned from some implicit information – i.e. *examples*.
- A *learner* is allowed to revise their hypothetical conjectures about the target language from time to time.
  So, a learner receives step-by-step more and more examples, and at each step they can revise their conjecture.
- How one can know that the target language L is successfully learned?
  - At each step, the learner explicates its internal conjecture about L in the form of some finite description, which aims to completely characterize the language.
    These descriptions are called *hypotheses*.
  - Typically, one has a fixed system of descriptions of languages – the *hypothesis space*.

Learning from examples (Gold 1967)

- At each step, the learner explicates its internal conjecture about L in the form of a finite hypothesis.
- We have a fixed system HS of descriptions of languages — the *hypothesis space*.

A **learner** M is an algorithmic device which is given more and more finite data, step-by-step. In each of the infinitely many steps of the learning process, M returns a hypothesis $h \in HS$.

Learning from examples (Gold 1967)

- At each step, the learner explicates its internal conjecture about L in the form of a finite hypothesis.
- We have a fixed system HS of descriptions of languages — the *hypothesis space*.

A **learner** M is an algorithmic device which is given more and more finite data, step-by-step. In each of the infinitely many steps of the learning process, M returns a hypothesis $h \in HS$.

A learner M successfully **learns** the target language L if

(a) M converges to some single hypothesis h_0;

(b) this hypothesis h_0 is a correct description of the target language L.

Learning from examples (Gold 1967)

- At each step, the learner explicates its internal conjecture about L in the form of a finite hypothesis.
- We have a fixed system HS of descriptions of languages — the *hypothesis space*.

A **learner** M is an algorithmic device which is given more and more finite data, step-by-step. In each of the infinitely many steps of the learning process, M returns a hypothesis $h \in HS$.

A learner M successfully **learns** the target language L if

(a) M converges to some single hypothesis h_0;

(b) this hypothesis h_0 is a correct description of the target language L.

After the Gold's paper, learning for languages was studied by Angluin, Barzdins, L. Blum, M. Blum, Case, Freivalds, Jain, Lange, Sharma, Stephan, Trakhtenbrot, Wiehagen, Zeugmann, Zilles, and other authors.

Two models of input: Text and informant (Gold 1967)

Let L be a language.

- «A *text* for L is a sequence of strings $x_1, x_2, \dots$ from L such that every string of L occurs at least once in the text. At time t the learner is presented x_t.» [Gold 1967]

  So, a text for L contains only positive information about L.

- «An *informant* for L can tell the learner whether any string is an element of L, and does so at each time t for some string y_t.» [Gold 1967]

  An informant for L contains both positive and negative data about L.

Example: TxtEx-Learning for c.e. languages

Here our target languages are computably enumerable (c.e.) subsets of the set of natural numbers $\mathbb{N}$.

Hypothesis Space:

We consider the standard computable numbering of the family of all c.e. sets $(W_e)_{e \in \mathbb{N}}$:

recall that $k \in W_e$ if and only if the e-th Turing machine program converges on the input k.

So, the possible hypotheses are natural numbers (viewed as indices of c.e. sets).

Definition

A *text* for a language $L \subseteq \mathbb{N}$ is an arbitrary function $t\colon \mathbb{N} \to (\mathbb{N} \cup \{\bot\})$ such that $L = \mathrm{range}(t) \setminus \{\bot\}$.

A *learner* is a function $M\colon (\mathbb{N} \cup \{\bot\})^{<\omega} \to \mathbb{N}$.

["$\bot$" means that a text t currently "refuses" to provide us with an example.]

TxtEx-Learning for c.e. languages

A *text* for a language $L \subseteq \mathbb{N}$ is an arbitrary function $t\colon \mathbb{N} \to (\mathbb{N} \cup \{\bot\})$ such that $L = \mathrm{range}(t) \setminus \{\bot\}$.

A *learner* is a function $M\colon (\mathbb{N} \cup \{\bot\})^{<\omega} \to \mathbb{N}$.

Definition

A learner M **$\mathbf{TxtEx}$-learns** a c.e. language L if for any text t for L, we have:

- There exists a limit $\lim_s M(L \upharpoonright s)$ — let denote this limit by $e^*(t)$.
- $W_{e^*(t)} = L$.

A learner M $\mathbf{TxtEx}$-*learns a family of languages* $\mathfrak{K}$ if M $\mathbf{TxtEx}$-learns every language from $\mathfrak{K}$.

TxtEx-Learning for c.e. languages

A *text* for a language $L \subseteq \mathbb{N}$ is an arbitrary function $t \colon \mathbb{N} \to (\mathbb{N} \cup \{\bot\})$ such that $L = \mathrm{range}(t) \setminus \{\bot\}$.

A *learner* is a function $M \colon (\mathbb{N} \cup \{\bot\})^{<\omega} \to \mathbb{N}$.

Definition

A learner M $\mathbf{TxtEx}$**-learns** a c.e. language L if for any text t for L, we have:

- There exists a limit $\lim_s M(L \upharpoonright s)$ — let denote this limit by $e^*(t)$.
- $W_{e^*(t)} = L$.

A learner M $\mathbf{TxtEx}$*-learns a family of languages* $\mathfrak{K}$ if M $\mathbf{TxtEx}$-learns every language from $\mathfrak{K}$.

- $\mathbf{Txt}$ stands for learning from *text* (i.e. only positive information).
- $\mathbf{Ex}$ stands for *explanatory learning*.

TxtEx-Learning for c.e. languages

Examples.

(0) The family of all finite sets is $\mathbf{TxtEx}$-learnable (by a computable learner M):

- Recall that there is the standard numbering of all finite sets: $D_0 = \emptyset$, and

$$\{k_1 < k_2 < \cdots < k_m\} = D_{2^{k_1}+2^{k_2}+\cdots+2^{k_m}}.$$

- One can choose a computable function $g(x)$ such that $D_x = W_{g(x)}$, for all $x \in \mathbb{N}$.
- Given an arbitrary text t, at a stage s:
  - We find the finite set $F = \mathrm{range}(t \upharpoonright s) \setminus \{\bot\}$.
  - Compute the index $x \in \mathbb{N}$ such that $F = D_x$.
  - The learner M outputs its conjecture $g(x)$.

TxtEx-Learning for c.e. languages

Examples.

(1) Any non-empty finite family of c.e. sets is $\mathbf{TxtEx}$-learnable (by a computable learner).

(2) The family $\{\mathbb{N} \setminus \{k\} : k \in \mathbb{N}\}$ is $\mathbf{TxtEx}$-learnable (by a computable learner).

(3) If a family $\mathfrak{K}$ contains a subfamily of the form

$$\{V : V \text{ is an infinite c.e. subset of } W\}$$

for some infinite c.e. W, then $\mathfrak{K}$ is not $\mathbf{TxtEx}$-learnable [Gold].

Plan of the further discussion

(i) Learning from informant for families of computable structures.

(ii) Learning, and continuous reducibility for equivalence relations on the Cantor space $2^{\mathbb{N}}$.

[We note that in 2000-s, Stephan and his co-authors developed another approach to learning for algebraic structures:

for example, they considered $\mathbf{TxtEx}$-learning for the family of all c.e. ideals in a computable commutative ring, learning for the family of c.e. closed sets in a c.e. matroid, etc.]

(i) Learning from informant for families of computable structures.

Computable algebraic structures

In the 1960s, Mal'tsev and Rabin initiated the systematic development of *computable structure theory* (or the theory of constructive models).

In the talk, we work only with at most countable algebraic structures $\mathcal{S}$. For the sake of simplicity, we assume that all our structures have finite signatures.

Recall that a structure $\mathcal{S}$ in the signature

$$\{P_0^{n_0}, P_1^{n_1}, \dots, P_k^{n_k}; f_0^{m_0}, f_1^{m_1}, \dots, f_\ell^{m_\ell}; c_0, c_1, \dots, c_r\}$$

is **computable** if:

- the domain of $\mathcal{S}$ is a computable subset of $\mathbb{N}$;
- the predicates $P_i^{\mathcal{S}}$ and the operations $f_j^{\mathcal{S}}$ are computable.

As usual, a computable structure $\mathcal{S}$ can be identified with its atomic diagram $D(\mathcal{S})$ (which, in turn, can be viewed as a computable subset of $\mathbb{N}$).

InfEx-Learning of isomorphism types

The paradigm essentially goes back to the monograph of Martin and Osherson "Elements of Scientific Inquiry" (1998). A modern exposition first appears in [Fokina, Kötzing, and San Mauro 2019].

Let L be a fixed finite signature.

An informal description of the learning paradigm:

For a (uniformly computable) family of computable L-structures $\mathfrak{K} = \{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \dots\}$, one wants to learn the isomorphism types of $\mathcal{A}_i$-s:

- A learner M obtains, step-by-step, finite pieces of the atomic diagram of an isomorphic copy $\mathcal{S}$ of some $\mathcal{A}_i$.

  Note that the copy $\mathcal{S}$ is *not necessarily computable*.

InfEx-Learning of isomorphism types

The paradigm essentially goes back to the monograph of Martin and Osherson "Elements of Scientific Inquiry" (1998). A modern exposition first appears in [Fokina, Kötzing, and San Mauro 2019].

Let L be a fixed finite signature.

An informal description of the learning paradigm:

For a (uniformly computable) family of computable L-structures $\mathfrak{K} = \{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \dots\}$, one wants to learn the isomorphism types of $\mathcal{A}_i$-s:

- A learner M obtains, step-by-step, finite pieces of the atomic diagram of an isomorphic copy $\mathcal{S}$ of some $\mathcal{A}_i$.

  Note that the copy $\mathcal{S}$ is *not necessarily computable*.
- At each stage, the learner outputs its current conjecture about the isomorphism type of $\mathcal{S}$:

  "right now, I think that the input $\mathcal{S}$ is isomorphic to, say, $\mathcal{A}_5$".
- The learning process is successful if:
  - the conjectures of M eventually stabilize — from some point of time, the learner will always say "$\mathcal{S}$ is a copy of $\mathcal{A}_{i^*}$", for some fixed i^*;
  - the final conjecture is correct: indeed, the input $\mathcal{S}$ is isomorphic to $\mathcal{A}_{i^*}$.

[Notice the following: If $\mathcal{S}$ is not a copy of a structure from $\mathfrak{K}$, then one does not care about the learner's behavior on input $\mathcal{S}$.]

A formal model: Finite families as a test case

We work with finite relational signatures L. In what follows, for simplicity, we assume that every considered structure $\mathcal{S}$ has domain $\mathbb{N}$.

Definition

The *canonical informant* for the structure $\mathcal{S}$ is the sequence $(\mathcal{S} \upharpoonright k)_{k \in \omega}$, where

$\mathcal{S} \upharpoonright k$ is (the code of the atomic diagram of) the restriction of $\mathcal{S}$ to the domain $\{0, 1, \ldots, k\}$.

Let $\mathfrak{K} = \{\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n\}$ be a finite family of infinite computable structures.

As usual, we assume that $\mathcal{A}_i$ are pairwise not isomorphic.

A formal model: Finite families as a test case

Let $\mathfrak{K} = \{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_n\}$ be a finite family of infinite computable structures.

An $\mathbf{InfEx}_{\cong}$-learner (or just $\mathbf{InfEx}$-learner) for the family $\mathfrak{K}$ is a function M such that:

- inputs of M are (codes of) finite L-structures with domain $\{0, 1, \dots, k\}$, where $k \in \omega$;
- $\mathrm{range}(M) \subseteq \{i : i \leq n\} \cup \{?\}$;

  [The symbol "?" in the output means that the learner abstains from giving a meaningful conjecture.]
- If $\mathcal{S} \cong \mathcal{A}_i$ and $\mathrm{dom}(\mathcal{S}) = \mathbb{N}$, then the learner M "guesses" the isomorphism type of $\mathcal{S}$ in the limit. More formally, we have

$$\lim_{k \to \infty} M(\mathcal{S} \upharpoonright k) = i.$$

Note that the input copy $\mathcal{S}$ can have arbitrary Turing degree.

A formal model: Finite families as a test case

Let $\mathfrak{K} = \{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_n\}$ be a finite family of infinite computable structures.

An $\mathbf{InfEx}_{\cong}$-learner (or just $\mathbf{InfEx}$-learner) for the family $\mathfrak{K}$ is a function M such that:

- inputs of M are (codes of) finite L-structures with domain $\{0, 1, \dots, k\}$, where $k \in \omega$;
- $\mathrm{range}(M) \subseteq \{i : i \leq n\} \cup \{?\}$;

  [The symbol "?" in the output means that the learner abstains from giving a meaningful conjecture.]
- If $\mathcal{S} \cong \mathcal{A}_i$ and $\mathrm{dom}(\mathcal{S}) = \mathbb{N}$, then the learner M "guesses" the isomorphism type of $\mathcal{S}$ in the limit. More formally, we have

$$\lim_{k \to \infty} M(\mathcal{S} \upharpoonright k) = i.$$

Note that the input copy $\mathcal{S}$ can have arbitrary Turing degree.

• $\mathbf{Inf}$ stands for learning from *informant* (both positive and negative data from the atomic diagram).

• $\mathbf{Ex}$ stands for *explanatory* learning.

InfEx-Learnability for finite families of structures

An **InfEx**-learner for the family $\mathfrak{K}$ is a function M such that:

- inputs of M are finite L-structures with domain $\{0, 1, \ldots, k\}$, where $k \in \omega$;
- $\mathrm{range}(M) \subseteq \{i : i \leq n\} \cup \{?\}$;
- If $\mathcal{S} \cong \mathcal{A}_i$, then the learner M "guesses" the isomorphism type of $\mathcal{S}$ in the limit — we have

$$\lim_{k \to \infty} M(\mathcal{S} \upharpoonright k) = i.$$

Definition

The family $\mathfrak{K}$ is **InfEx**-learnable if there exists an **InfEx**-learner for $\mathfrak{K}$.

InfEx-Learnability: Examples

Example 1. Consider two undirected graphs G_1 and G_2:

- G_1 has infinitely many cycles of size three, and nothing else;
- G_2 has infinitely many cycles of size four, and nothing else.

The family $\{G_1, G_2\}$ is **InfEx**-learnable via the following effective procedure:

- Wait until the input data shows a cycle of size 3 or 4. While waiting, continue outputting "$?$".
- When (the first) such cycle appears in the input, start forever outputting the natural guess:
  "G_1" for size 3, and "G_2" for size 4.

Example 2. Consider two equivalence structures $\mathcal{E}_1$ and $\mathcal{E}_2$:

- all equivalence classes of $\mathcal{E}_1$ are singletons;
- $\mathcal{E}_2$ has infinitely many classes of size one, precisely one class of size two, and nothing else.

The family $\{\mathcal{E}_1, \mathcal{E}_2\}$ is **InfEx**-learnable:

- Wait until the input data shows two different elements, which are equivalent. While waiting, continue outputting "$\mathcal{E}_1$".
- When we see that the input structure contains some class of size ≥ 2, we start forever outputting "$\mathcal{E}_2$".

Example 3. The pair of linear orders $\{\omega, \omega^*\}$ is **InfEx**-learnable.

Recall that an input structure $\mathcal{L}$ (which is isomorphic either to ω or to ω^*) is given in stages.

At a stage s, let ℓ_s be the $\leq_{\mathcal{L}}$-least element in the current finite linear order $\mathcal{L} \upharpoonright s$. Let r_s be the current $\leq_{\mathcal{L}}$-greatest element.

We define auxiliary counters:

- $c[\ell_s] = \max\{t \leq s : \ell_{s-t} = \ell_s\}$,
- $c[r_s] = \max\{t \leq s : r_{s-t} = r_t\}$.

The learning algorithm is arranged as follows: At a stage s,

- if $c[\ell_s] > c[r_s]$, then output "ω";
- otherwise, output "ω^*".

A syntactic characterization: Infinitary formulas

It turns out that $\mathbf{InfEx}$-learnability admits a syntactic characterization.

In order to formulate the results, we need to talk about infinitary Σ_2-formulas (Σ_2^{inf}-formulas):

- A Σ_0^{inf}-formula (a Π_0^{inf}-formula) is a quantifier-free formula.
- An *infinitary* Σ_{n+1}*-formula* (Σ_{n+1}^{inf}-formula) is a countable disjunction

$$\bigvee_{i \in I} \exists \bar{y}_i \psi_i(\bar{x}, \bar{y}_i),$$

where each ψ_i is a Π_n^{inf} formula.

- A Π_{n+1}^{inf}*-formula* is a countable conjunction

$$\bigwedge_{i \in I} \forall \bar{y}_i \psi_i(\bar{x}, \bar{y}_i),$$

where each ψ_i is a Σ_n^{inf}-formula.

A syntactic characterization: Infinitary formulas

It turns out that $\mathbf{InfEx}$-learnability admits a syntactic characterization.

In order to formulate the results, we need to talk about infinitary Σ_2-formulas (Σ_2^{inf}-formulas):

- A Σ_0^{inf}-formula (a Π_0^{inf}-formula) is a quantifier-free formula.
- An *infinitary* Σ_{n+1}*-formula* (Σ_{n+1}^{inf}-formula) is a countable disjunction

$$\bigvee_{i \in I} \exists \bar{y}_i \psi_i(\bar{x}, \bar{y}_i),$$

where each ψ_i is a Π_n^{inf} formula.

- A Π_{n+1}^{inf}*-formula* is a countable conjunction

$$\bigwedge_{i \in I} \forall \bar{y}_i \psi_i(\bar{x}, \bar{y}_i),$$

where each ψ_i is a Σ_n^{inf}-formula.

For an oracle $X \subseteq \mathbb{N}$, one can also formally define the class of X-computable infinitary Σ_n-formulas ($\Sigma_n^c(X)$-formulas) — their conjunctions and disjunctions range over X-c.e. sets of formulas.

InfEx-Learnability: Going syntactic

In the setting of finite families, everything “boils down” to the syntactic properties of structures:

Theorem 1 (implicit in Martin and Osherson 1998; a new proof by B., Fokina, and San Mauro 2020)

Let $\mathfrak{K} = \{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_n\}$ be a family of computable L-structures. Then the following conditions are equivalent:

(i) The family $\mathfrak{K}$ is $\mathbf{InfEx}$-learnable.

(ii) There are Σ_2^{inf}-sentences $\psi_0, \psi_1, \dots, \psi_n$ such that

$$\mathcal{A}_i \models \psi_j \text{ if and only if } i = j.$$

InfEx-Learnability: Going syntactic

In the setting of finite families, everything "boils down" to the syntactic properties of structures:

Theorem 1 (implicit in Martin and Osherson 1998; a new proof by B., Fokina, and San Mauro 2020)

Let $\mathfrak{K} = \{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_n\}$ be a family of computable L-structures. Then the following conditions are equivalent:

(i) The family $\mathfrak{K}$ is $\mathbf{InfEx}$-learnable.

(ii) There are Σ_2^{inf}-sentences $\psi_0, \psi_1, \dots, \psi_n$ such that

$$\mathcal{A}_i \models \psi_j \text{ if and only if } i = j.$$

The new proof is *effective*:

Effective Theorem 1 (B., Fokina, and San Mauro 2020)

For an arbitrary oracle $X \subseteq \mathbb{N}$, the following are equivalent:

(1) The family $\mathfrak{K}$ is $\mathbf{InfEx}$-learnable by an X-computable learner.

(2) There are $\Sigma_2^c(X)$-sentences $\psi_0, \psi_1, \dots, \psi_n$ such that

$$\mathcal{A}_i \models \psi_j \text{ if and only if } i = j.$$

A comment on the proof of Effective Theorem 1

Turing computable embeddings play the key role in the proof.

Definition (Calvert, Cummins, Knight, and S. Miller 2004)

For $i \in \{1, 2\}$, let $\mathfrak{K}_i$ be a class of countable L_i-structures (closed under isomorphisms). A Turing operator Φ is a *Turing computable embedding* from $\mathfrak{K}_1$ to $\mathfrak{K}_2$ if:

- For every $\mathcal{A} \in \mathfrak{K}_1$, the function $\Phi^{D(\mathcal{A})}$ computes the atomic diagram of a structure from the class $\mathfrak{K}_2$ — this structure is denoted by $\Phi(\mathcal{A})$.
- For any $\mathcal{A}, \mathcal{B} \in \mathfrak{K}_1$, we have:

$$\mathcal{A} \cong \mathcal{B} \;\Leftrightarrow\; \Phi(\mathcal{A}) \cong \Phi(\mathcal{B}).$$

Roughly speaking, a Turing computable embedding is an *effectively continuous* function from the space $Mod(L_1) \upharpoonright \mathfrak{K}_1$ into $Mod(L_2) \upharpoonright \mathfrak{K}_2$ (to be elaborated in the second part).

Turing degrees of InfEx-learners

Theorem 2 (B. and San Mauro 2021)

(a) Every $\mathbf{InfEx}$-learnable finite family is learnable by a $\mathbf{0}'$-computable learner.

(b) There exists an $\mathbf{InfEx}$-learnable pair of structures $\{\mathcal{A}, \mathcal{B}\}$, which is not learnable by a computable learner.

The general case: InfEx-Learnability for infinite families

Let $\mathfrak{K}$ be a family of computably presentable structures (closed under isomorphisms). We say that a function $\nu\colon \mathbb{N} \to \mathfrak{K}$ is an effective enumeration (up to isomorphism) of the family $\mathfrak{K}$ if ν satisfies the following:

1. The sequence of structures $(\nu(e))_{e\in\mathbb{N}}$ is uniformly computable.
2. For any structure $\mathcal{S}$ from $\mathfrak{K}$, there is an index e such that $\nu(e) \cong \mathcal{S}$.

The general case: InfEx-Learnability for infinite families

Let $\mathfrak{K}$ be a family of computably presentable structures (closed under isomorphisms). We say that a function $\nu\colon \mathbb{N} \to \mathfrak{K}$ is an effective enumeration (up to isomorphism) of the family $\mathfrak{K}$ if ν satisfies the following:

1. The sequence of structures $(\nu(e))_{e\in\mathbb{N}}$ is uniformly computable.
2. For any structure $\mathcal{S}$ from $\mathfrak{K}$, there is an index e such that $\nu(e) \cong \mathcal{S}$.

Definition

Let $\mathfrak{K}$ be a family of computably presentable structures, and let ν be an effective enumeration of the family $\mathfrak{K}$.
The family $\mathfrak{K}$ is $\mathbf{InfEx}[\nu]$-learnable if there is an $\mathbf{InfEx}$-learner M such that for any $\mathcal{S} \in \mathfrak{K}$ with $\mathrm{dom}(\mathcal{S}) = \mathbb{N}$, there exists a limit

$$\lim_{k\to\infty} M(\mathcal{S} \upharpoonright k) = e \in \mathbb{N},$$

satisfying $\nu(e) \cong \mathcal{S}$.

The general case: InfEx-Learnability for infinite families

Theorem 1 Revisited

Let $\mathfrak{K} = \{\mathcal{A}_i : i \in \mathbb{N}\}$ be a family of computable L-structures, and let ν be an effective enumeration of $\mathfrak{K}$. We assume that $\mathcal{A}_i \not\cong \mathcal{A}_j$ for $i \neq j$.

Then the following conditions are equivalent:

(i) The family $\mathfrak{K}$ is $\mathbf{InfEx}[\nu]$-learnable.

(ii) There are Σ_2^{inf}-sentences ψ_i, $i \in \mathbb{N}$, such that

$$\mathcal{A}_i \models \psi_j \text{ if and only if } i = j.$$

The general case: InfEx-Learnability for infinite families

Theorem 1 Revisited

Let $\mathfrak{K} = \{\mathcal{A}_i : i \in \mathbb{N}\}$ be a family of computable L-structures, and let ν be an effective enumeration of $\mathfrak{K}$. We assume that $\mathcal{A}_i \not\cong \mathcal{A}_j$ for $i \neq j$.

Then the following conditions are equivalent:

(i) The family $\mathfrak{K}$ is $\mathbf{InfEx}[\nu]$-learnable.

(ii) There are Σ_2^{inf}-sentences ψ_i, $i \in \mathbb{N}$, such that

$$\mathcal{A}_i \models \psi_j \text{ if and only if } i = j.$$

Via a careful syntactic analysis, one can obtain the following:

Theorem 3 (B., Fokina, and San Mauro 2020)

(a) There are no $\mathbf{InfEx}$-learnable families $\mathfrak{K}$ of Boolean algebras with $\mathrm{card}(\mathfrak{K}) \geq 2$.

(b) There are no infinite $\mathbf{InfEx}$-learnable families of linear orders.

These bounds on cardinality are sharp.

A simple example: An InfEx-learnable family $\mathfrak{K}$ of abelian p-groups

For $k \in \mathbb{N}$, consider the following abelian p-group:

$$\nu(k) = \mathcal{A}_k := \bigoplus_{j \in \omega} \mathbb{Z}(p^{k+1}).$$

Informally, a separating formula ψ_k (which distinguishes $\mathcal{A}_k$) says that:

- $\mathbb{Z}(p^{k+1})$ is a subgroup of $\mathcal{A}$ (this is a finitary $\exists$-formula), and
- $\mathbb{Z}(p^{k+2})$ is not a subgroup of $\mathcal{A}$ (a finitary $\forall$-formula).

Therefore, by Theorem 1, the family $\mathfrak{K}$ is $\mathbf{InfEx}[\nu]$-learnable.

(ii) Learning, and continuous reducibility for equivalence relations on the Cantor space $2^{\mathbb{N}}$.

Polish spaces

Recall that a *Polish space* is a separable, completely metrizable topological space.

Here we work with the following Polish spaces:

- The Cantor space $2^{\mathbb{N}}$ consists of infinite binary strings.

  For strings $\alpha \neq \beta$, the distance $d(\alpha, \beta)$ equals 2^{-n}, where $n \in \mathbb{N}$ is the least such that the n-th bit of α is not equal to the n-th bit of β.

  The space $2^{\mathbb{N}}$ is homeomorphic to the standard Cantor set from real analysis.

Polish spaces

Recall that a *Polish space* is a separable, completely metrizable topological space.

Here we work with the following Polish spaces:

- The Cantor space $2^{\mathbb{N}}$ consists of infinite binary strings.

  For strings $\alpha \neq \beta$, the distance $d(\alpha, \beta)$ equals 2^{-n}, where $n \in \mathbb{N}$ is the least such that the n-th bit of α is not equal to the n-th bit of β.

  The space $2^{\mathbb{N}}$ is homeomorphic to the standard Cantor set from real analysis.
- Let L be a finite signature. By using an appropriate encoding, one can identify an L-structure $\mathcal{S}$ such that $\mathrm{dom}(\mathcal{S}) = \mathbb{N}$ with an infinite binary string.

  This procedure induces a topological space $Mod(L)$ of countably infinite L-structures. The space is homeomorphic to 2^{ω}.

Continuous reducibility

One of the major topics in descriptive set theory is that of *Borel reducibility*.

Definition

Let X and Y be Polish spaces. Suppose that E is an equivalence relation on X, and F is an equivalence relation on Y.

The relation E is *Borel reducible* to F if there exists a Borel map $f\colon X \to Y$ such that for all $x, y \in X$

$$(x\,E\,y) \;\Leftrightarrow\; (f(x)\,F\,f(y)). \tag{1}$$

Here we work with a stronger reducibility:

E is continuously reducible to F if the map f from Eq. (1) is continuous.

A connection to InfEx-learning

The relation E is *continuously reducible* to F if there exists a continuous map $f\colon X \to Y$ such that for all $x, y \in X$

$$(x\,E\,y) \;\Leftrightarrow\; (f(x)\,F\,f(y)).$$

- Let $\mathfrak{K}$ be a family of L-structures. By $Mod(L) \upharpoonright \mathfrak{K}$ we denote the subspace of $Mod(L)$, which contains the isomorphic copies of structures from $\mathfrak{K}$.
- The equivalence relation E_0 ("almost equality") on the Cantor space $2^{\mathbb{N}}$ is defined as follows:

$$(\alpha\,E_0\,\beta) \;\Leftrightarrow\; \exists n(\forall m \geq n)[\alpha(m) = \beta(m)].$$

A connection to InfEx-learning

The relation E is *continuously reducible* to F if there exists a continuous map $f\colon X \to Y$ such that for all $x, y \in X$

$$(x\, E\, y) \;\Leftrightarrow\; (f(x)\, F\, f(y)).$$

- Let $\mathfrak{K}$ be a family of L-structures. By $Mod(L) \restriction \mathfrak{K}$ we denote the subspace of $Mod(L)$, which contains the isomorphic copies of structures from $\mathfrak{K}$.
- The equivalence relation E_0 ("almost equality") on the Cantor space $2^{\mathbb{N}}$ is defined as follows:

$$(\alpha\, E_0\, \beta) \;\Leftrightarrow\; \exists n (\forall m \geq n)[\alpha(m) = \beta(m)].$$

Theorem 4 (B., Cipriani, and San Mauro)

Let $\mathfrak{K}$ be a family of L-structures, and let ν be its effective enumeration. Then the following are equivalent:

(a) $\mathfrak{K}$ is $\mathbf{InfEx}[\nu]$-learnable;

(b) There exists a continuous function $\Gamma\colon Mod(L) \to 2^{\mathbb{N}}$ such that Γ induces a continuous reduction from the isomorphism relation $(\cong \restriction \mathfrak{K})$ (on the space $Mod(L) \restriction \mathfrak{K}$) to E_0 — i.e. for any $\mathcal{A}, \mathcal{B} \in \mathfrak{K}$, we have

$$(\mathcal{A} \cong \mathcal{B}) \;\Leftrightarrow\; (\Gamma(\mathcal{A})\, E_0\, \Gamma(\mathcal{B})).$$

Learning as a continuous reduction

Theorem 4 (B., Cipriani, and San Mauro)

Let $\mathfrak{K}$ be a family of L-structures, and let ν be its effective enumeration. Then the following are equivalent:

(a) $\mathfrak{K}$ is $\mathbf{InfEx}[\nu]$-learnable;

(b) There exists a continuous function $\Gamma\colon Mod(L) \to 2^{\mathbb{N}}$ such that Γ induces a continuous reduction from the isomorphism relation $(\cong \restriction \mathfrak{K})$ (on the space $Mod(L) \restriction \mathfrak{K}$) to E_0 — i.e. for any $\mathcal{A}, \mathcal{B} \in \mathfrak{K}$, we have

$$(\mathcal{A} \cong \mathcal{B}) \;\Leftrightarrow\; (\Gamma(\mathcal{A})\, E_0\, \Gamma(\mathcal{B})).$$

Theorem 4 motivates us to introduce the following notion:

Definition

Let $\mathfrak{K}$ be an *arbitrary* family of countable (not necessarily computable) L-structures. Let E be an equivalence relation on the Cantor space.

We say that $\mathfrak{K}$ is <u>E-learnable</u> if there is a continuous function $\Gamma\colon Mod(L) \to 2^{\mathbb{N}}$ such that Γ induces a continuous reduction from the isomorphism relation $(\cong \restriction \mathfrak{K})$ to E — i.e. for any $\mathcal{A}, \mathcal{B} \in \mathfrak{K}$, we have $(\mathcal{A} \cong \mathcal{B}) \;\Leftrightarrow\; (\Gamma(\mathcal{A})\, E\, \Gamma(\mathcal{B}))$.

Combinatorial equivalence relations

Definition

Let $\mathfrak{K}$ be an arbitrary family of countable L-structures. Let E be an equivalence relation on $2^{\mathbb{N}}$.

The family $\mathfrak{K}$ is <u>E-learnable</u> if there is a continuous function $\Gamma\colon Mod(L) \to 2^{\mathbb{N}}$ such that Γ induces a continuous reduction from the isomorphism relation $(\cong \restriction \mathfrak{K})$ to E — i.e. for any $\mathcal{A}, \mathcal{B} \in \mathfrak{K}$, we have $(\mathcal{A} \cong \mathcal{B}) \;\Leftrightarrow\; (\Gamma(\mathcal{A})\, E\, \Gamma(\mathcal{B}))$.

Here we discuss E-learnability only for three benchmark combinatorial equivalence relations on $2^{\mathbb{N}}$:

- <u>$(\alpha\, E_2\, \beta)$</u> if and only if
$$\sum_{k\,:\,\alpha \triangle \beta(k)=1} \frac{1}{k+1} < \infty.$$
- The m-th column of $\alpha \in 2^{\mathbb{N}}$ is defined as follows: $\alpha^{[m]}(i) = \alpha(\langle m, i \rangle)$, for $i \in \mathbb{N}$.
  <u>$(\alpha\, E_3\, \beta)$</u> if and only if $(\forall m)(\alpha^{[m]}\, E_0\, \beta^{[m]})$.
- <u>$(\alpha\, E_{set}\, \beta)$</u> if and only if $\{\alpha^{[m]} : m \in \omega\} = \{\beta^{[m]} : m \in \omega\}$.

E_2-learnability

$\underline{(\alpha \, E_2 \, \beta)}$ if and only if

$$\sum_{k \,:\, \alpha \triangle \beta(k) = 1} \frac{1}{k+1} < \infty.$$

Note that E_0 is continuously reducible to E_2, and E_2 is not even Borel reducible to E_0.

Theorem 5 (B., Cipriani, and San Mauro)

Let $\mathfrak{K}$ be at most countable family of L-structures. Then $\mathfrak{K}$ is E_2-learnable if and only if $\mathfrak{K}$ is E_0-learnable.

E_3-learnability

The m-th column of α is defined as $\alpha^{[m]}(i) = \alpha(\langle m, i \rangle)$, for $i \in \mathbb{N}$.
$(\alpha \, E_3 \, \beta)$ if and only if $(\forall m)(\alpha^{[m]} \, E_0 \, \beta^{[m]})$.

Again, E_0 is continuously reducible to E_3, and E_3 is not Borel reducible to E_0.

Theorem 6 (B., Cipriani, and San Mauro)

(a) A finite family $\mathfrak{K}$ is E_3-learnable if and only if it is E_0-learnable.

(b) There exists a uniformly computable family which is E_3-learnable, but not E_0-learnable.

E_3-learnability

The m-th column of α is defined as $\alpha^{[m]}(i) = \alpha(\langle m, i\rangle)$, for $i \in \mathbb{N}$.
$(\alpha \, E_3 \, \beta)$ if and only if $(\forall m)(\alpha^{[m]} \, E_0 \, \beta^{[m]})$.

Again, E_0 is continuously reducible to E_3, and E_3 is not Borel reducible to E_0.

Theorem 6 (B., Cipriani, and San Mauro)

(a) A finite family $\mathfrak{K}$ is E_3-learnable if and only if it is E_0-learnable.

(b) There exists a uniformly computable family which is E_3-learnable, but not E_0-learnable.

Theorem 7 (B., Cipriani, and San Mauro)

Let $\mathfrak{K}$ be at most countable family of L-structures. Then $\mathfrak{K}$ is E_3-learnable if and only if there exists a family Θ of Σ_2^{inf}-sentences with the following properties:

- For every $\theta \in \Theta$, there exists $\psi \in \Theta$ such that for any $\mathcal{A} \in \mathfrak{K}$, we have $\mathcal{A} \models (\theta \leftrightarrow \neg\psi)$.
- If $\mathcal{A} \not\cong \mathcal{B}$ are structures from $\mathfrak{K}$, then there is $\theta \in \Theta$ such that $\mathcal{A} \models \theta$ and $\mathcal{B} \models \neg\theta$.

A remark on E_{set}-learnability

The m-th column of α is defined as $\alpha^{[m]}(i) = \alpha(\langle m, i \rangle)$, for $i \in \mathbb{N}$.
$\underline{(\alpha \, E_{set} \, \beta)}$ if and only if $\{\alpha^{[m]} : m \in \omega\} = \{\beta^{[m]} : m \in \omega\}$.

The relation E_3 is continuously reducible to E_{set}, and E_{set} is not Borel reducible to E_3.

Proposition (B., Cipriani, and San Mauro)

The pair of linear orders $\{\omega, \zeta\}$ is E_{set}-learnable, but not E_0-learnable.

Open problems

There are two interesting directions, where many questions are still open.

Direction 1

Study possible Turing degrees for $\mathbf{InfEx}$-learners. For example:

Recall that there is a pair of structures, which is $\mathbf{0}'$-learnable, but not computably learnable. What about other Turing degrees $\mathbf{d}$ in place of $\mathbf{0}'$?

Direction 2

Classify benchmark equivalence relations from descriptive set theory in terms of their "learning capability". For example:

For the relation E_{set}, we still have only a small partial result. Is it possible to obtain a nice syntactic characterization of E_{set}-learnability?

References

- E. Martin and D. Osherson, *Elements of Scientific Inquiry*, MIT Press, 1998.
- S. Jain, D. Osherson, J. S. Royer, and A. Sharma, *Systems that Learn: An Introduction to Learning Theory*, MIT Press, 1999.
- F. Stephan and Yu. Ventsov, *Learning algebraic structures from text*, Theoretical Computer Science, 268(2001), 221–273.
- S. Lange, T. Zeugmann, and S. Zilles, *Learning indexed families of recursive languages from positive data: a survey*, Theoretical Computer Science, 397 (2008), 194–232.
- N. Bazhenov, E. Fokina, and L. San Mauro, *Learning families of algebraic structures from informant*, Information and Computation, 275 (2020), article id 104590.
- N. Bazhenov and L. San Mauro, *On the Turing complexity of learning finite families of algebraic structures*, Journal of Logic and Computation, published online, doi: 10.1093/logcom/exab044.

Appendix: What about TxtEx-learning for structures?

Similarly to $\mathbf{InfEx}_{\cong}$-learning, one can also develop the model of $\mathbf{TxtEx}_{\cong}$-learning for structures:

- Recall that $\mathbf{Txt}$ stands for learning from text (i.e. only positive information from atomic diagrams).
- Topologically speaking, this model corresponds to the Scott topology on $2^{\mathbb{N}}$.
- Syntactically speaking, one can obtain a complete syntactic characterization of $\mathbf{TxtEx}_{\cong}$-learnability. The characterization uses *positive* infinitary Σ_2-formulas [Soskov 2004].

(Work in progress with Fokina, Rossegger, A. Soskova, M. Soskova, and Vatev).