TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

Algorithmic Learning of Structures

Ekaterina Fokina

TU Wien

Midwest Computability Seminar/
Computability Theory and Applications
February 21, 2022

Learning structures

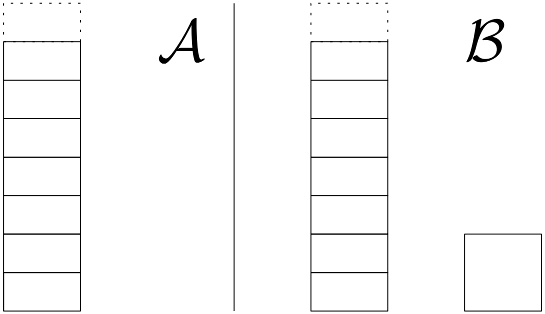
- Suppose we have a class of (countable) structures.
- Suppose we are stage by stage seeing one of the structures from the class: at each step a larger and larger finite piece of the structure.

Question

Can we, after finitely many steps, identify the structure (up to an isomorphism or other equivalence relations)?

Example 1

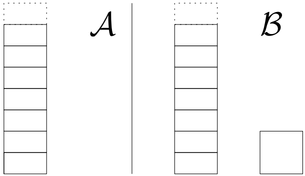
Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.



Graphics by Luca San Mauro

Example 1

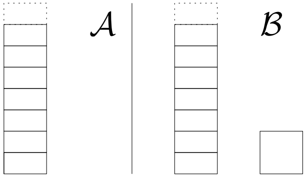
Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.



$\mathcal{S}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$

Graphics by Luca San Mauro

Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.

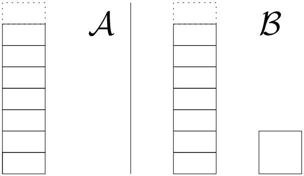


$\mathcal{S}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$

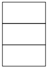
Graphics by Luca San Mauro

Example 1

Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.



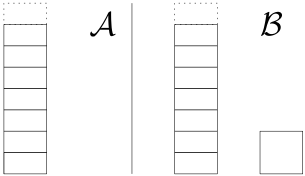
$\mathcal{S}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$



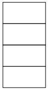
Graphics by Luca San Mauro

Example 1

Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.



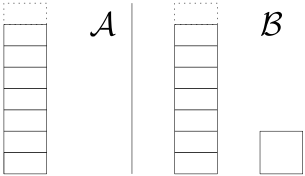
$\mathcal{S}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$



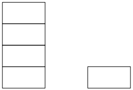
Graphics by Luca San Mauro

Example 1

Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.



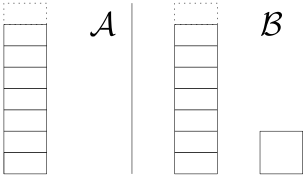
$\mathcal{S}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$

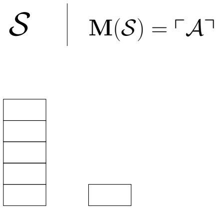


Graphics by Luca San Mauro

Example 1

Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.

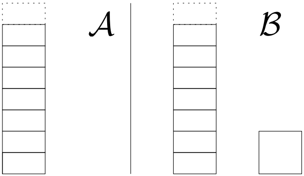


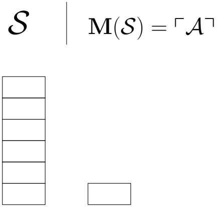


Graphics by Luca San Mauro

Example 1

Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.

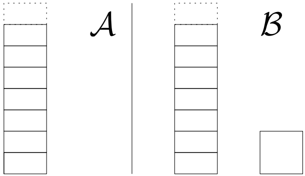


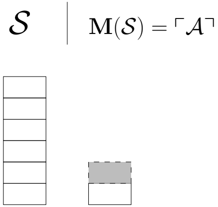


Graphics by Luca San Mauro

Example 1

Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.

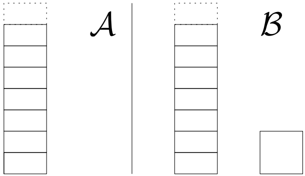


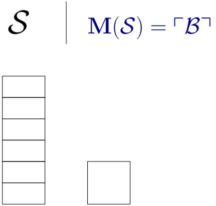


Graphics by Luca San Mauro

Example 1

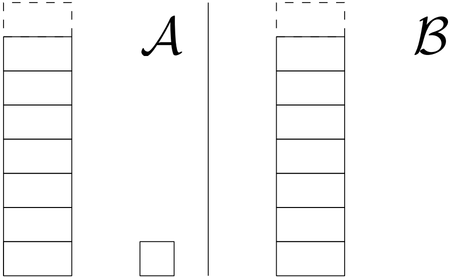
Let $\mathcal{A} = (\omega, E)$ and $B = (\omega, F)$ be equivalence structures.





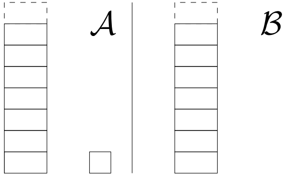
Graphics by Luca San Mauro

Example 2



Graphics by Luca San Mauro

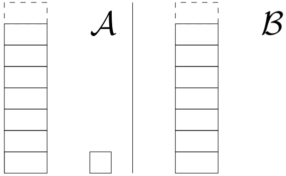
Example 2



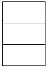
$\mathcal{S} \cong \mathcal{B} \;\Big|\; \mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$

Graphics by Luca San Mauro

Example 2

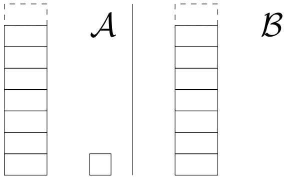


$\mathcal{S} \cong \mathcal{B}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$



Graphics by Luca San Mauro

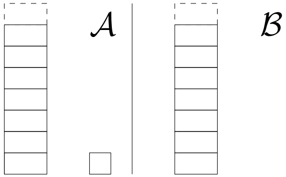
Example 2



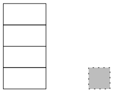
$\mathcal{S} \cong \mathcal{B}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{B} \urcorner$

Graphics by Luca San Mauro

Example 2

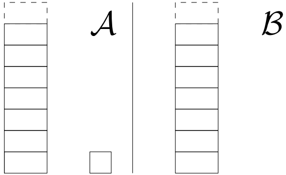


$\mathcal{S} \cong \mathcal{B}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{B} \urcorner$

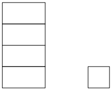


Graphics by Luca San Mauro

Example 2

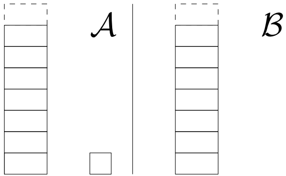


$\mathcal{S} \cong \mathcal{A}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{B} \urcorner$

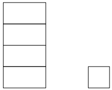


Graphics by Luca San Mauro

Example 2

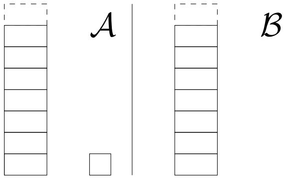


$\mathcal{S} \cong \mathcal{A}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{B} \urcorner$

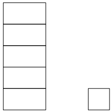


Graphics by Luca San Mauro

Example 2

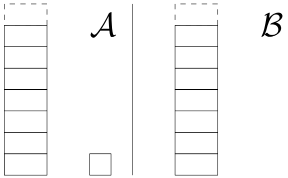


$\mathcal{S} \cong \mathcal{A}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{B} \urcorner$

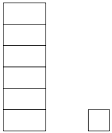


Graphics by Luca San Mauro

Example 2

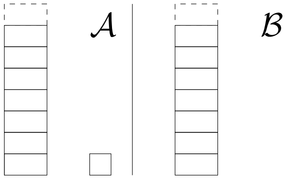


$\mathcal{S} \cong \mathcal{A}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$

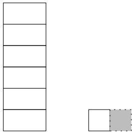


Graphics by Luca San Mauro

Example 2

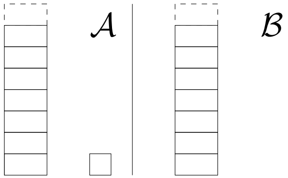


$\mathcal{S} \cong \mathcal{A}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$

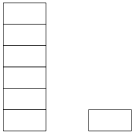


Graphics by Luca San Mauro

Example 2



$\mathcal{S} \cong \mathcal{B}$ | $\mathbf{M}(\mathcal{S}) = \ulcorner \mathcal{A} \urcorner$



Graphics by Luca San Mauro

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

A B

Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

C A B

Hypothesis: ω^*

Examples 3 and 4

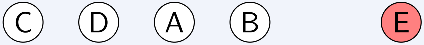
Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

Ⓒ Ⓓ Ⓐ Ⓑ

Hypothesis: ω^*

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$



Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

F C D A B E

Hypothesis: ω^*

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

F C D A B E G

Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

F C D A B E G H

Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

Ⓕ Ⓘ Ⓒ Ⓓ Ⓐ Ⓑ Ⓙ Ⓔ Ⓖ Ⓗ

Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

F I C D A B J E G H $\cdots$

Hypothesis: ω

Examples 3 and 4

Structures $(\omega, \leq)$ and $(\omega^*, \leq)$

F I C D A B J E G H $\cdots$

Hypothesis: ω

Structures $(\omega, \leq)$ and $(\zeta, \leq)$

Hypothesis: alternating between ω and ζ.

Conclusion and overview

- How is the structure "revealed"?
- First part: both the positive and the negative information about the structure (motivated by computable structures).
- Second part: only the positive information about the structure is revealed (motivated by c.e. structures).
- What does it mean "to classify", or "to identify" the structure?
- To formalize these and other issues, we use the ideas from computable structure theory and computational learning theory.

Computational Learning Theory

Computational Learning Theory (CLT): deals with the question of how a learner, provided with more and more data about some environment, is eventually able to achieve systematic knowledge about it.

- (Gold, 1967): language identification.

Most work in CLT concerns

- either learning of total functions (where the order in which the data is received matters)
- or learning of formal languages (where the order does not matter)

These paradigms model the data to be learned as an unstructured flow — but what if one deals with data having some structural content?

CLT and Structures

More recently researchers applied the machinery of CLT to algebraic structures:

- Glymour, 1985
- Martin, Osherson, 1998
- Stephan, Ventsov, 2001: learning ring ideals of commutative rings.
- Merkle, Stephan, 2004 learning isolated branches on uniformly computable sequences of trees.
- Harizanov, Stephan, 2007: learning subspaces of V_∞.
- Gao, Stephan, Wu, Yamamoto, 2012: learning closed sets in matroids.
- F., Kötzing, San Mauro, 2018: learning equivalence structures.

Our goal is: to combine the technology of CLT with notions coming from computable structure theory to develop a general framework for learning the isomorphism type of algebraic structures.

Computable Structures

To learn the isomorphism type of a given structure, one should be able to name such an isomorphism type. This is why we focus on the learning of (copies of) computable or c.e. structures.

Learning should be independent from the way in which data is presented. So, a successful learning procedure should work for all isomorphic copies of a given structure.

Formal Example

Consider a family $\mathfrak{D}$: for each $i \geq 1$, the graph G_i contains infinitely many $(i+1)$-cycles.

- **The learning domain**: is the family $\mathfrak{D}^*$ of *all* presentations of the graphs in $\mathfrak{D}$;
- **The information source**: an informant I for a graph H in $\mathfrak{C}^*$ is an infinite list of pairs containing: all pairs (x, y) of natural numbers, as the first component; and either 0 or 1, as the second component.
- **The hypothesis space**: every conjecture is an element of the set $\omega \cup \{?\}$;
- **The learner**: a function (or an algorithm) that learns, up to isomorphism, any graph in $\mathfrak{D}^*$;
- **The prior knowledge**: the target graph is isomorphic to some graph from the family $\mathfrak{D}$.
- **The criterion of success**: a learner that, receiving larger and larger pieces of any graph G in $\mathfrak{C}^*$, eventually stabilizes to a correct guess about whether G is isomorphic to G_1 or G_2.

Enumerations I

How does one formally define the set of possible conjectures?

First Solution:
For $m \in \omega$, the conjecture "m" means "$H \cong G_{m+1}$."

- This solution is similar to the so-called *exact learning*, considered in the setting of c.e. languages, where one assumes that the hypothesis space of the problem is precisely the class being learned with the corresponding indexing.

  General framework: Fix a Friedberg enumeration of the class $\mathfrak{D}$ and interpret the output hypotheses with respect to this enumeration.
- Drawback: it can be computationally very hard to enumerate certain familiar families of computable structures, up to isomorphism.
- Goncharov and Knight: for the classes of computable Boolean algebras, linear orders, and abelian p-groups one cannot even hyperarithmetically enumerate their isomorphism types.

Enumerations II

Second Solution:
Fix a uniformly computable sequence $(\mathcal{M}_e)_{e\in\omega}$ of *all* computable undirected graphs. The conjecture "m" means "$H \cong \mathcal{M}_m$."
This solution is similar to the so-called *class-comprising learning*, where one assumes that the hypothesis space of the problem should only contain the class being learned.

General framework: consider an arbitrary superclass $\mathfrak{K}_0 \supseteq \mathfrak{D}$ which is *uniformly enumerable*, i.e. there is a uniformly computable sequence of structures $(\mathcal{N}_e)_{e\in\omega}$ such that:

1. Any structure from $\mathfrak{K}_0$ is isomorphic to some $\mathcal{N}_e$.
2. For every e, $\mathcal{N}_e$ belongs to $\mathfrak{K}_0$.

Then for a number $e \in \omega$, the conjecture "e" is interpreted as "the input structure is isomorphic to $\mathcal{N}_e$."

Our framework

Let L be a relational signature.
Let $\mathfrak{K}_0$ be a class of L-structures, and let ν be an effective enumeration of $\mathfrak{K}_0$. Suppose that $\mathfrak{K}$ is a subclass of $\mathfrak{K}_0$.

Definition

We say that $\mathfrak{K}$ is $\mathbf{InfEx}_{\cong}[\nu]$-*learnable* if there is a learner M with the following property:
If I is an informant for a structure $\mathcal{A} \in \mathfrak{K}$, then there are e and s_0 such that $\nu(e) \cong \mathcal{A}$ and $M(I[s]) = e$ for all $s \geq s_0$. In other words, in the limit, the learner M learns all isomorphism types from $\mathfrak{K}$.

Infinitary formulas

Together with Bazhenov and San Mauro we are able to characterize learnable families of structures. To do so, we use the logic $\mathcal{L}_{\omega_1\omega}$.

The class of infinitary Σ_α L-formulas:

(a) Σ_0^{inf} and Π_0^{inf} formulas are quantifier-free first-order L-formulas.

(b) A $\Sigma_\alpha^{\mathrm{inf}}$ formula $\psi(x_0, \dots, x_m)$ is an countable disjunction

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

where each ξ_i is a $\Pi_{\beta_i}^{\mathrm{inf}}$ formula, for some $\beta_i < \alpha$.

(c) A $\Pi_\alpha^{\mathrm{inf}}$ formula $\psi(\bar{x})$ is a countable conjunction

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

where each ξ_i is a $\Sigma_{\beta_i}^{\mathrm{inf}}$ formula, for some $\beta_i < \alpha$.

Today we will only need $\Sigma_\alpha^{\mathrm{inf}}$ formulas for $\alpha \leq 2$.

Infinitary formulas

Together with Bazhenov and San Mauro we are able to characterize learnable families of structures. To do so, we use the logic $\mathcal{L}_{\omega_1\omega}$.

The class of X-computable infinitary Σ_α L-formulas:

(a) $\Sigma^c_0(X)$ and $\Pi^c_0(X)$ formulas are quantifier-free first-order L-formulas.

(b) A $\Sigma^c_\alpha(X)$ formula $\psi(x_0, \ldots, x_m)$ is an X-computably enumerable (X-c.e.) disjunction

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

where each ξ_i is a $\Pi^c_{\beta_i}(X)$ formula, for some $\beta_i < \alpha$.

(c) A $\Pi^c_\alpha(X)$ formula $\psi(\bar{x})$ is an X-c.e. conjunction

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

where each ξ_i is a $\Sigma^c_{\beta_i}(X)$ formula, for some $\beta_i < \alpha$.

Today we will only need $\Sigma^c_\alpha(X)$ formulas for $\alpha \leq 2$.

Main Theorem

Suppose that $\mathfrak{K}_0$ is a class of L-structures, and ν is an effective enumeration of the class $\mathfrak{K}_0$.

Theorem (Bazhenov, F., San Mauro)

Let $\mathfrak{K} = \{\mathcal{B}_i : i \in \omega\}$ be a family of structures such that $\mathfrak{K} \subseteq \mathfrak{K}_0$, and the structures $\mathcal{B}_i$ are infinite and pairwise non-isomorphic. Then the following conditions are equivalent:

1. *The class $\mathfrak{K}$ is $\mathbf{InfEx}_{\cong}[\nu]$-learnable;*
2. *There is a sequence of $\Sigma_2^{\inf}$ sentences $\{\psi_i : i \in \omega\}$ such that for all i and j, we have $\mathcal{B}_j \models \psi_i$ if and only if $i = j$.*

The statement is similar to a result due to Martin and Osherson. Yet, our proof is novel and based on Pullback Theorem in the context of Turing computable embeddings introduced by Knight, S. Miller, and Vanden Boom. This provides us with an upper bound for the Turing complexity of the learners which we apply later.

Bounding the complexity

Corollary (Bazhenov, F., San Mauro)

Let $X \subseteq \omega$ be an oracle. Let $\mathfrak{K}_0$ be a class of countably infinite L-structures, and ν be an effective enumeration of $\mathfrak{K}_0$. Assume that either $I = \omega$, or I is a finite initial segment of ω. Consider a subclass $\mathfrak{K} = \{\mathcal{B}_i : i \in I\}$ inside $\mathfrak{K}_0$. Suppose that:

(i) *There is uniformly X-computable sequence of $\Sigma^c_2(X)$ sentences $(\psi_i)_{i \in I}$ such that:*

$$\mathcal{B}_j \models \psi_i \;\Leftrightarrow\; i = j.$$

(ii) *There is an X-computable sequence $(e_i)_{i \in I}$ such that $\nu(e_i) \cong \mathcal{B}_i$ for all i. Note that if the set I is finite, then one can always choose this sequence in a computable way.*

Then the class $\mathfrak{K}$ is $\mathbf{InfEx}_{\cong}[\nu]$*-learnable via an X-computable learner.*

Corollary (BFS)

- *There exist learnable classes of:*
  - *lattices,*
  - *abelian groups,*
  - *linear orders (only finite classes)*
- *There are no learnable classes for*
  - *Boolean algebras,*
  - *Infinite classes of linear orders.*

Bazhenov, Cipriani, San Mauro: Learning and Borel ER's

Theorem (Bazhenov, Cipriani, San Mauro)

A family of structures $\mathfrak{K}$ is learnable if and only if there is a continuous function $\Gamma : 2^\omega \to 2^\omega$ such that

$$\mathcal{A} \cong B \Longleftrightarrow \Gamma(\mathcal{A}) E_0 \Gamma(B),$$

for all $\mathcal{A}, B \in LD(\mathfrak{K})$.

Definition

A family of structures $\mathfrak{K}$ is E-learnable if there is function $\Gamma : 2^\omega \to 2^\omega$ which continuously reduce $LD(\mathfrak{K})$ to E.

Theorem (BCS)

A family $\mathfrak{K}$ is E_2-learnable if and only if $\mathfrak{K}$ is E_1-learnable if and only if $\mathfrak{K}$ is E_0-learnable.

Bazhenov, Cipriani, San Mauro: Learning and Borel ER's

Theorem (BCS)

- *A finite family $\mathfrak{K}$ is E_3-learnable if and only if $\mathfrak{K}$ is E_0-learnable.*
- *There exists an infinite family $\mathfrak{K}$ which is E_3- learnable, but not E_0-learnable.*

Theorem (BCS)

Let $\mathfrak{K} = \{\mathcal{A}_i : i \in \omega\}$ be a countable family. The family $\mathfrak{K}$ is E_3-learnable if and only if there exists a countable family of Σ_2^{inf} sentences $\Theta$with the following properties:

1. *if θ is a formula from Θ, then there is a formula $\psi \in \Theta$such that for every $\mathcal{A} \in \mathfrak{K}$,*
$$A \models \theta \Leftrightarrow \mathcal{A} \not\models \psi;$$
2. *if $\mathcal{A} \ncong B$ are structures from K, then there is a sentence theta $\in \Theta$ such that*
$$\mathcal{A} \models \theta \text{ and } B \not\models \theta.$$

The notion of E-learnability opens the possibility to define and study uncountable learnable classes of structures.

Other learnability classes, I

We obtain different learnability classes by replacing the main ingredients of $\mathbf{InfEx}_{\cong}$ with natural alternatives:

1. $\mathbf{Inf} \mapsto \mathbf{Txt}$: in **Txt**-learning (short for *text*) the learner receives only positive information of the structure to be learnt.
2. $\cong \mapsto E$, where E is some nice equivalence relations relation between elements of $\mathbb{K}$, such as bi-embeddability ($\approx$), computable isomorphism ($\cong^{\mathbf{0}}$), computable bi-embeddability ($\approx^{\mathbf{0}}$) – and so forth.
3. $\mathbf{Ex} \mapsto \mathbf{BC}$: in **BC**-learning (short for *behaviourally correct*) the learner is allowed to change its mind infinitely many times as far as almost all its conjectures lie in the same E-class (with E defined as in 2.).
4. Yet another dimension to consider is the complexity of the learner.

Learning from Text

$\mathfrak{K} = \{\mathcal{A}_i : i \in \omega\}$ be a family of infinite computable L-structures.

- The *learning domain*:

$$\mathrm{LD}(\mathfrak{K}) = \bigcup_{\mathcal{A} \in \mathfrak{K}} \{\mathcal{S} : \mathcal{S} \cong \mathcal{A}, \text{ and } \mathrm{dom}(\mathcal{S}) = \omega\}.$$

- The *hypothesis space* (HS):

$$\mathrm{HS}(\mathfrak{K}) = \omega \cup \{?\}.$$

- A *learner* M sees, stage by stage, *only positive* data about a given structure from $LD(\mathfrak{K})$. The learner M is required to output conjectures from $HS(\mathfrak{K})$.
- The learning is *successful* if for any structure $\mathcal{S} \in \mathrm{LD}(\mathfrak{K})$, the learner eventually stabilizes to a correct conjecture about the isomorphism type of $\mathcal{S}$.

We say that the family $\mathfrak{K}$ is **TxtEx**-*learnable* if there exists a learner M which succesfully learns $\mathfrak{K}$.

Positive infinite formulas

Let $\{=,\neq\} \subseteq L$. Define a hierarchy of formulas, the $L_{p\omega}$ formulas:

Definition (Soskov)

1. $\varphi \in \Sigma^p_1$ iff
$$\varphi \equiv \mathbb{W}_{i\in\omega} \exists \overline{x}_i \psi_i(\overline{x}_i)$$
where every ψ_i is a finite conjunction of atomic L-formulas,
2. $\varphi \in \Pi^p_1$ iff
$$\varphi \equiv \mathbb{M}_{i\in\omega} \forall \overline{x}_i \psi_i(\overline{x}_i)$$
where every ψ_i is a finite disjunction of negations of atomic L-formulas,
3. $\varphi \in \Sigma^p_{\alpha+1}$ iff
$$\varphi \equiv \mathbb{W}_{i\in\omega} \exists \overline{x}_i (\psi_i(\overline{x}_i) \wedge \theta_i(\overline{x}_i))$$
where ψ_i is Σ^p_α and θ_i is Π^p_α for all i.
4. $\varphi \in \Pi^p_{\alpha+1}$ iff
$$\varphi \equiv \mathbb{M}_{i\in\omega} \forall \overline{x}_i (\psi_i(\overline{x}) \vee \theta_i(\overline{x}))$$
where ψ_i is Σ^p_α and θ_i is Π^p_α for all i.
5. $\varphi \in \Sigma^p_\lambda$ for λ a limit ordinal iff $\varphi \equiv \mathbb{W}_{i\in\omega} \psi_i$, where each ψ_i is $\Sigma^p_{\beta_i}$ with $\beta_i < \lambda$,
6. $\varphi \in \Pi^p_\lambda$ for λ a limit ordinal iff $\varphi \equiv \mathbb{M}_{i\in\omega} \psi_i$, where each ψ_i is $\Sigma^p_{\beta_i}$ with $\beta_i < \lambda$.

Syntactic characterization for TxtEx-learning

Work in progress:

Theorem (Bazhenov, F., Rosseger, A. Soskova, M. Soskova, Vatev)

Let $\mathfrak{K} = \{\mathcal{A}_i : i \in \omega\}$ be a class of computable infinite L-structures (here we assume that $\mathcal{A}_i \not\cong \mathcal{A}_j$ for $i \neq j$). The following are equivalent.

1. *The class $\mathfrak{K}$ is* **TxtEx**-*learnable.*
2. *For every $\mathcal{A}_i \in \mathfrak{K}$ there is a Σ_2^p sentence φ_i such that for all $\mathcal{A}_k \in \mathfrak{K}$, $\mathcal{A}_k \models \varphi_i$ if and only if $k = i$.*

Corollary

Let X be an oracle. Suppose that there exists a uniformly X-computable sequence of $\Sigma_2^{p,c}$ sentence φ_i such that for all $\mathcal{A}_k \in \mathfrak{K}$, $\mathcal{A}_k \models \varphi_i$ if and only if $k = i$. Then the family $\mathfrak{K}$ is **TxtEx**-*learnable by an X-computable learner.*

References

[1] C. Glymour. Inductive inference in the limit. Erkenntnis, 22:23–31, 1985.

[2] E. Martin and D. Osherson. Elements of scientific inquiry. MIT Press, 1998.

[3] F. Stephan and Yu. Ventsov. Learning algebraic structures from text. Theoret. Comput. Sci., 268(2):221–273, 2001.

[4] W. Merkle and F. Stephan. Trees and learning. J. Comput. System Sci., 68(1):134–156, 2004.

[5] V. Harizanov and F. Stephan. On the learnability of vector spaces. J. Comput. System Sci., 73(1):109–122, 2007.

[6] Z. Gao, F. Stephan, G. Wu, and A. Yamamoto. Learning Families of Closed Sets in Matroids. Computation, Physics and Beyond. WTCS 2012. LNCS, 7160: 120–139, 2012.

[7] E. Fokina, T. Kötzing, and L. San Mauro. Limit learning equivalence structures. Proceedings of the 30th International Conference on Algorithmic Learning Theory, volume 98 of PMLR, 383–403, Chicago, Illinois, 22–24 Mar 2019.

[8] N. Bazhenov, E. Fokina, and L. San Mauro. Learning families of algebraic structures from informant, Information and Computation, 275, 2020.