Learning in the Limit: a mutational and adaptive approach

Reginaldo Inojosa Silva Filho, Ricardo Luis de Azevedo da Rocha, Ricardo Henrique Gracini Guiraldelli

Computing Engineering Department
Engineering School of the University of São Paulo,
Av. Luciano Gualberto s/n Trav 3, n.158 São Paulo - SP, Brazil
reginaldo.uspoli@gmail.com, rlarocha@usp.br, rguira@usp.br

Abstract. The purpose of this work is to show the strong connection between learning in the limit and the second-order adaptive automaton. The connection is established using the mutating programs approach, in which any hypothesis can be used to start a learning process, and produces a correct final model following a step-by-step transformation of that hypothesis by a second-order adaptive automaton. Second-order adaptive automaton learner will be proved to acts as a learning in the limit one[1].

Keywords: Inductive Inference, Learning Model, Code Mutation, Adaptive Devices, Automata Theory

1 Introduction

Ray Solomonoff was the father of the general theory of inductive inference [11], a fruitful area of study that generated many developments in artificial intelligence [5,13]. In short, the goal of inductive inference is to identify the unknown object by picking out one of a (typically infinite) set of hypothesis for this object [1]. The hypothesis is a finite representation of the object and may be consistent with the given incrementally growing segments of object example inputs. It is possible to define many ways to the hypothesis choice and each one, in practice, determines a whole new learning model; the main ones are the probabilistic approach [5,13] and the enumerations strategies [4]. But, as pointed out by Wallace, Dowe and Solomonoff himself, the so-called “Solomonoff Induction” is actually prediction [13,2,12], rather than induction. Therefore, the approach used in this work is closely related to Wallace's Minimum Message Length (MML) approach, but was inspired by Solomonoff's paper [11].

Thus, consider the following constraint: what if the only way to generate a new hypothesis was by “recycling” a former one? What would the behavior of the learner be if the generation of a new hypothesis implies the transformation of an older one? What kind of transformation would be necessary?

[1] The work reported here received support through FAPESP grant 2010/09586-0.

This kind of hypothesis needs to be a "changeable" one to be reused, that is, it would have a "plasticity" feature to adapt to new inputs for which it is not prepared; therefore, the changes must happen in the representation structure used to describe the hypothesis. Using a biological metaphor, the hypothesis must have the mutational property, with the learner having the responsibility to apply transformations in the hypothesis. The use of this kind of metaphor, nonetheless, is not new: Solomonoff was also interested in the study of mutating programs [6]. Now, this work proposes a mutational computational model, called second-order adaptive automaton, aimed at the problem of inductive inference and in which this "changeable" behavior is an essential property of the model itself.

A successful mutation means that the learner has adapted to the new inputs. The fact that learning can be represented by an adaptive process is the fundamental premise of this work. The last two decades presented the development and emergence of new computational models that deviate from a basis on mechanical machines structures and become similar to evolutive, collaborative and biologically inspired models; among these, self-modifying devices [8,10] are prominent ones, which have been developed under a formalism based on automata theory and is one of the basis to represent the adaptive behavior of the model defined here.

The description of the inductive inference under an adaptive aspect will be made using the Emil Mark Gold learning in the limit model [3], also called identification in the limit. Responsible for branch of inductive inference, Gold studied the learning problem for recursive functions and formal languages. In his model, the inductive inference is an infinite process; a learner identifies a language if the generation of hypotheses converges to one and no other changes occur, although new inputs of the language are presented to the learner indefinitely.

The text is organized as follows: section 2 describes the notation and technical preliminaries concerning automata theory and first-order adaptive automata, basis for the second-order adaptive automata; section 3 presents the the second-order adaptive automata. Section 4 shows the relationship between second-order adaptive automata and learning in the limit. Finally, section 5 presents the conclusion and further works to be developed.

2 The First-Order Adaptive Automaton

Adaptive automaton belongs to the category of self-modifying devices. It is a computational model equivalent to Turing Machine [7] and has the non-deterministic finite state automaton as formulation basis. Its major characteristic is the ability to decide how to modify its own structure in response to some external input, without the interference of any external agent. The first appearance of the adaptive automata has some inconsistencies in its description, but this fact was corrected later in a complete formalization, performed using the automata transformations concept [10]. This formalization, developed in the present section, is known as **first-order adaptive automata** (FOAA). However, some introductory concepts are presented before for unequivocal understanding of the FOAA definition.

2.1 Notations and Technical Preliminaries

The main concepts used in this work, mostly concerning automata theory, as well the pertinent notation, are summarized in table 1.

$\mathbb{N} = \{0, 1, 2, \dots\}$	**The set of natural numbers**
$I = \{i_0, i_1, \dots, i_m\}$	**Finite arbitrary indexed set**
$rem(I, x) = \{I - \{x\} : x \in I\}$ with $(x \notin I)$	**Removal function**
$ins(I, x) = \{i_0, i_1, \dots, i_{m+1}\}$ with $i_{m+1} = x$	**Insertion function**
Σ	**An alphabet of symbols**
$\alpha \in \Sigma$	**A symbol of the alphabet**
$L \subseteq \Sigma^*$	**A language over Σ**
$t \in L$	**A string of L**
ε	**An empty string**
$\theta = (t_0, t_1, ...)$, with $t_k \in L$ for $k \in \mathbb{N}$	**A text of L**
$\theta[n] = (t_1, t_2, ...t_n)$, with $t_k \in L$ for $0 \leq k \leq n$	**the n-initial segment of θ**
$seq(\theta)$	**the family of all segments**
$M^0 = (Q, q_0, E, \Sigma, \partial)$	**A non-deterministic automaton**
$Q = \{q_1, \dots, q_n\}$	**The set of states of M^0**
$q_0 \in Q$	**The initial state of M^0**
$E \subseteq Q$	**The accepting states set of M^0**
$\partial \subseteq Q \times \{\Sigma \cup \{\varepsilon\}\} \times Q$	**The state transition relation**
$\partial = \{\delta_1, \delta_2, \dots, \delta_i\}$	**The transitions set of M^0**
$\delta = (q', \alpha, q'')$ with $\{q', q''\} \subseteq Q$ and $\alpha \in \Sigma$	**A transition of M^0**
$(q', t) \in Q \times \Sigma^*$ with $q' \in Q$	**A configuration of M^0**
(q_0, t)	**The initial configuration of M^0**

Table 1. Notation for technical preliminaries related to the automata theory.

A **scalar hierarchical structure**[9] is indicated by $\langle a_n \langle a_{n-1} \dots \langle a_1 \langle a_0 \rangle \rangle \rangle \rangle$, and is interpreted as follows: if a_{i+1} is a formal system defined by an ordered n-tuple, then a_i is a n-tuple element, for $0 \leq i \leq (n-1)$.

2.2 Automata transformations

Given the non-numerical set $\mathscr{M}^0$ of all non-deterministic finite state automaton under an alphabet Σ, for any element $M^0 \in \mathscr{M}^0$ and the state transition relation ∂ of M^0, a **proper transition** is defined as:

$$\delta_{pro} = \delta : \delta \in \partial \tag{1}$$

Otherwise, a **foreign transition** is defined as:

$$\delta_{for} = \delta : \delta \notin \partial \tag{2}$$

A sequence of proper transitions belong to M^0 and represented by

$$\lambda_{pro} = (\delta_{pro_1}, \ldots, \delta_{pro_m}) \tag{3}$$

is called a **positive sequence**. In turn, a transitions sequence

$$\lambda_{for} = (\delta_{for_1}, \ldots, \delta_{for_n}) \tag{4}$$

of foreign transition for M^0 is called a **negative sequence**.

Given a negative and a positive sequence for an automaton M^0, the sequence:

$$\phi = (\lambda_{for}, \lambda_{pro}) \tag{5}$$

is defined as a **first-order transformation pair**.

Employing the proper and foreign transition concepts, as well the definition of removal and insertion functions, it is possible to define transformation functions for all members of $\mathscr{M}^0$. Thus, the **δ-removal operation** and **δ-insertion operation** are defined, respectively, by:

$$f_k^- M^0 = f^-(\delta_{pro_k}, M^0) = (Q, q_0, E, \Sigma, \mathbf{rem}(\partial, \delta_{pro_k})) \tag{6}$$
$$f_k^+ M^0 = f^+(\delta_{for_k}, M^0) = (\mathbf{ins}(\mathbf{ins}(Q, q'), q''), q_0, E, \Sigma, \mathbf{ins}(\partial, \delta_{for_k})) \tag{7}$$

with $\delta_{pro_k} \in \lambda_{pro}$ and $\delta_{for_k} \in \lambda_{for}$.

Now, using this two operators, it is possible to introduce the concept of **first-order adaptive function**:

$$\mathbb{F}_\phi M^0 \triangleq \mathbb{F}(\phi, M^0) = F^-_{\lambda_{pro}} F^+_{\lambda_{for}} M^0 \tag{8}$$

in which

$$F^-_{\lambda_{pro}} M^0 \triangleq F^-(\lambda_{pro}, M^0) = (f_m^- \circ f_{m-1}^- \circ \ldots \circ f_2^- \circ f_1^-) M^0 \tag{9}$$

$$F^+_{\lambda_{for}} M^0 \triangleq F^+(\lambda_{for}, M^0) = (f_n^+ \circ f_{n-1}^+ \circ \ldots \circ f_2^+ \circ f_1^+) M^0 \tag{10}$$

are the **first-order removal transformation** and **first-order insertion transformation**.

Definition 1 (First-order Adaptive Automata) *A First-Order Adaptive Automata (FOAA) is the quadruple $M^1 = (M^0, \Phi, \phi^{\emptyset}, \partial^1)$, in which $M^0 \in \mathscr{M}^0$ is called **first-order subjacent device**. Set Φ of the first-order transformation pairs is called **adaptive behavior set**. The element $\phi^{\emptyset} \in \Phi$ is a void transformation pair called **null behavior**. Set ∂^1 is the **first-order adaptive transition relation**. Each element of ∂^1 takes the form $\delta^1_{i,k} = (\delta_i, \langle M^1 \langle \mathbb{F}_{\phi_k} M^0 \rangle \rangle)$, for $\phi_k \in \Phi$ and $\delta_i \in \partial$ in which ∂ is the first-order subjacent device state-transition relation.*

Any extension of the automaton concept implies a new expression for it. Hence, the traditional elements of the automata theory (step function, etc.) were brought into the FOAA model.

The **one step function** shows how the FOAA changes from one configuration to another:

$$(q', t) \vdash_{[\mathbb{F}_{\phi_k} M^0]} (q'', w) \Leftrightarrow \exists \alpha \in \Sigma : \alpha w = t \tag{11}$$

in which q'' is a state of $\mathbb{F}_{\phi_k} M^0$ and $((q', \alpha, q''), \langle M^1 \langle \mathbb{F}_{\phi_k}(M^0) \rangle \rangle) \in \partial^1$ for $\phi_k \in \Phi$. The **closure of the one step function** for a FOAA is defined as:

$$(q', t) \vdash^*_{[\mathbb{F}_{\phi_{k_j}} \dots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0]} (q'', w) \tag{12}$$

iff $(q' = q'')$ and $(w = t)$ or rules 1, 2 and 3 are all satisfied as defined below:

1. $t = a_0 a_1 \dots a_j w$ with $a_i \in \Sigma$ for $0 \leq i \leq j$
2. $\exists(\phi_{k_1}, \phi_{k_2}, \dots \phi_{k_{j+1}})$ with $\phi_{k_i} \in \Phi$ for $1 \leq i \leq j$
3. $\exists\ p_1, p_2 \dots p_j \in Q$ in which Q belongs to first-order subjacent device, such that,
   for $j \in \mathbb{N}$:

$$\begin{aligned}(q', t) &\vdash_{[\mathbb{F}_{\phi_{k_1}} M^0]} (p_1, a_1 a_2 a_3 \dots a_j w) \\ &\vdash_{[\mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0]} (p_2, a_2 a_3 \dots a_j w) \vdash_{[\mathbb{F}_{\phi_{k_3}} \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0]} \cdots \\ &\vdash_{[\mathbb{F}_{\phi_{k_j}} \dots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0]} (p_j, a_j w) \\ &\vdash_{[\mathbb{F}_{\phi_{k_{j+1}}} \mathbb{F}_{\phi_{k_j}} \dots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0]} (q'', w)\end{aligned}$$

The **language recognized by the FOAA** is

$$L(M^1) = \{t : (q_0, t) \vdash^*_{[\mathbb{F}_{\phi_{k_j}} \dots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0]} (q_f, \varepsilon)\} \tag{13}$$

Special case in which the behavior set is $\Phi = \{\phi^{\emptyset}\}$, the necessary condition for a string to be accepted by a FOAA assuming the form:

$$(q_0, t) \vdash^*_{[\mathbb{F}_{\phi^{\emptyset}} \dots \mathbb{F}_{\phi^{\emptyset}} \mathbb{F}_{\phi^{\emptyset}} M^0]} (q_f, \varepsilon) = (q_0, w) \vdash^*_{M^0} (q_f, \varepsilon)$$

3 The Second-Order Adaptive Automaton

Now, taking the set $\mathscr{M}^1$ of all first-order adaptive automata $M^1 = (M^0, \Phi, \phi^\emptyset, \partial^1)$ for a fixed Σ and applying the same method used for the set $\mathscr{M}^0$ (the definition of basic insertion and removal operations for FOAAs transitions), analogous to what occurred in the previous section, it is possible to obtain similar concepts, now to study the first-order adaptive automata set features under a set of operators. Therefore, the table below summarizes these concepts and their definitions:

$\delta^1_{pro} = \delta^1 : \delta^1 \in \partial^1$	δ^1**-proper transition**
$\delta^1_{for} = \delta^1 : \delta^1 \notin \partial^1$	δ^1**-foreign transition**
$\lambda^1_{pro} = (\delta^1_{pro_1}, \ldots, \delta^1_{pro_m})$	δ^1**-positive sequence**
$\lambda^1_{for} = (\delta^1_{for_1}, \ldots, \delta^1_{for_m})$	δ^1**-negative sequence**
$\psi \triangleq (\lambda^1_{for}, \lambda^1_{pro})$	δ^1**-transformation pair**
$g^-_k M^1 = g^-(\delta^1_{pro_k}, M^1) = (M^0, \Phi, \phi^\emptyset, \mathbf{rem}(\partial^1, \delta^1))$	δ^1**-removal operation**
$g^+_k M^1 = g^+(\delta^1_{for_k}, M^1) = (M^0, \mathbf{ins}(\Phi, \phi), \phi^\emptyset, \mathbf{ins}(\partial^1, \delta^1))$	δ^1**-insertion operation**

Table 2. Set of operations that structures the SOAA.

The **second-order adaptive function** is the operator

$$\mathbb{G}_\psi \triangleq \mathbb{G}(\psi, M^1) = G^-_{\lambda^1_{pro}} G^+_{\lambda^1_{for}} M^1 \tag{14}$$

in which

$$G^-_{\lambda^1_{pro}} M^1 \triangleq G^-(\lambda^1_{pro}, M^1) = (g^-_m \circ g^-_{m-1} \circ \ldots \circ g^-_2 \circ g^-_1) M^1 \tag{15}$$

$$G^+_{\lambda^1_{for}} M^1 \triangleq G^+(\lambda^1_{for}, M^1) = (g^+_n \circ g^+_{n-1} \circ \ldots \circ g^+_2 \circ g^+_1) M^1 \tag{16}$$

are the **second-order removal transformation** and **second-order insertion transformation**, respectively. Similar with the first-order case, the pair $\psi^\emptyset$ is called of **void** second-order characteristic pair. Thus, for an empty second-order characteristic pair, $\mathbb{G}_{\psi^\emptyset} M^1$, it is equal to M^1.

Definition 2 (Second-Order Adaptive Automata) *A Second-Order Adaptive Automata (SOAA) is the quadruple $M^2 = (M^1, \Psi, \psi_0, \partial^2)$, in which $M^1 \in \mathscr{M}^1$ is called **second-order subjacent device**. The set $\Psi = \{\psi_0, \psi_1, \ldots, \psi_n\}$ of second-order transformation pairs is called **second-order adaptive behavior set**. The element ψ_0 is a void transformation pair called **null behavior**. In the **second-order adaptive transition relation** ∂^2, each element take the form $\delta^2_{i,k,j} = (\delta^1_{i,k}, \langle M^2 \langle \mathbb{G}_{\psi_j} M^1 \rangle \rangle)$, for $\psi_j \in \Psi$ and $\delta^1_{i,k} \in \partial^1$, in which ∂^1 is the second-order subjacent device state-transition relation.*

The **one step function** shows how the SOAA changes from one configuration to another and is defined below:

$$(q', t) \vdash_{[\langle \mathbb{G}_{\psi_j} M^1 \langle \mathbb{F}_{\phi_k} M^0 \rangle \rangle]} (q'', w) \Leftrightarrow \exists \alpha \in \Sigma : \alpha w = t \tag{17}$$

in which q'' is a state of $\mathbb{F}_{\phi_k} M^0$ for $\delta_{i,k,j} = (\delta_{i,k}, \langle M^2 \langle \mathbb{G}_{\psi_j} M^1 \rangle \rangle) \in \partial^2$, $\delta_{i,k} = (\delta_i, \langle M^1 \langle \mathbb{F}_{\phi_k} M^0 \rangle \rangle) \in \partial^1$ and $\delta_i = (q', \alpha, q'') \in \partial$ with $\phi_k \in \Phi$ and $\psi_j \in \Psi$.

For any $s \in \mathbb{N}$, the **closure of the one step function for a SOAA** is defined as:

$$(q', t) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_s}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1 \langle \mathbb{F}_{\phi_{k_s}} \ldots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0 \rangle \rangle]} (q'', w) \tag{18}$$

iff $(q' = q'')$ and $(w = t)$ or rules 1 to 4 are all satisfied as defined below:

1. $t = a_0 a_1 \ldots a_s w$ with $a_z \in \Sigma$ for $0 \leq z \leq s$
2. $\exists (\phi_{k_1}, \phi_{k_2}, \ldots \phi_{k_{s+1}})$ with $\phi_{k_z} \in \Phi$ for $1 \leq z \leq s$
3. $\exists (\psi_{j_1}, \psi_{j_2}, \ldots \psi_{j_{s+1}})$ with $\psi_{j_z} \in \Psi$ for $1 \leq z \leq s$
4. $\exists\ p_1, p_2 \ldots p_s \in Q$ with Q belongs to First-order subjacent device such that:

$$\begin{aligned}
(q', t) &\vdash_{[\langle \mathbb{G}_{\psi_{j_1}} M^1 \langle \mathbb{F}_{\phi_{k_1}} M^0 \rangle \rangle]} (p_1, a_1 a_2 a_3 \ldots a_s w) \\
&\vdash_{[\langle \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1 \langle \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0 \rangle]} (p_2, a_2 a_3 \ldots a_s w) \\
&\vdash_{[\langle \mathbb{G}_{\psi_{j_3}} \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} \langle \mathbb{F}_{\phi_{k_3}} \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0 \rangle \rangle]} \cdots \\
&\vdash_{[\langle \mathbb{G}_{\psi_{j_s}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1 \langle \mathbb{F}_{\phi_{k_s}} \ldots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0 \rangle \rangle]} (p_s, a_s w) \\
&\vdash_{[\langle \mathbb{G}_{\psi_{j_{s+1}}} \mathbb{G}_{\psi_{j_s}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1 \langle \mathbb{F}_{\phi_{k_{s+1}}} \mathbb{F}_{\phi_{k_s}} \ldots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0 \rangle \rangle]} (q'', w)
\end{aligned}$$

The language recognized by the SOAA is:

$$L(M^2) = \{t : (q_0, t) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_s}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1 \langle \mathbb{F}_{\phi_{k_j}} \ldots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0 \rangle \rangle]} (q_f, \varepsilon)\} \tag{19}$$

In the special case in which the second-order behavior set is $\Psi = \{\psi^{\emptyset}\}$, the necessary condition for a string to be accepted by a SOAA assumes the form:

$$\begin{aligned}
&(q_0, t) \vdash^*_{[\langle \mathbb{G}_{\psi^{\emptyset}} \ldots \mathbb{G}_{\psi^{\emptyset}} \mathbb{G}_{\psi^{\emptyset}} M^1 \langle \mathbb{F}_{\phi_{k_j}} \ldots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0 \rangle \rangle]} (q_f, \varepsilon) = \\
&(q_0, t) \vdash^*_{[\mathbb{F}_{\phi_{k_j}} \ldots \mathbb{F}_{\phi_{k_2}} \mathbb{F}_{\phi_{k_1}} M^0]} (q_f, \varepsilon)
\end{aligned}$$

In this case, the recognition of string t is made by the initial second-order subjacent device. The SOAA assumes the behavior of the FOAA, which is equivalent to the Turing Machine.

4 Second-Order Adaptive Automata and Learning in the Limit

This section will show the advantage of using the SOAA as an identification in the limit Inductive Inference Machine for formal languages. By definition, SOAA transforms FOAAs by applying on them second-order adaptive actions.

Now, it is necessary to demonstrate how this behavior can be used to “recycle” the former hypothesis created in a learning in the limit process, as stated in the Introduction, and what kind of formal languages a SOAA can learn in the limit. Firstly, the main definitions related to the Gold identification in the limit are presented. Then, an illustrating example of learning in the limit using a SOAA is presented in subsection 4.1.

Definition 3 (Inductive Inference Machine) *Let a target formal language indexable class $\mathcal{L}$ and a hypothesis set $\mathcal{H}$ composed by an useful enumerable formal model class (grammars, Turing machines, recursive functions, etc) to represent the members of the target languages class. Given the family $seq(\theta)$ for $\theta \in text(L)$, in which L belongs to $\mathcal{L}$, an inductive inference machine (IIM in short) is defined as an effective procedure in which it computes any partial or total mapping $IIM \subseteq seq(\theta) \times \mathcal{H}$.*

The IIM **changes its mind** if two consecutives output hypotheses are different, i.e., $IIM(\theta[m]) \neq IIM(\theta[m+1])$ for $m \geq 0$.

The expression

$$IIM(\theta) \downarrow = h \Leftrightarrow \exists(n \in \mathbb{N})\exists(h \in \mathcal{H})(\forall m \geq n)[IIM(\theta[m])] = h \tag{20}$$

means that the inductive inference machine **converges**, i.e., the potential infinite sequence $[IIM(\theta[m])]_{m\in\mathbb{N}}$ of outputs converges on θ to $h \in \mathcal{H}$.

Definition 4 (Identification in the Limit) *Let $\mathcal{L}$ be an indexed family of languages, given a convenient hypothesis space $\mathcal{H}$. IIM **Lim-identifies** $\mathcal{L} \Leftrightarrow \forall(L \in \mathcal{L})\exists(h \in \mathcal{H} : L(h) = L)[IIM(\theta) \downarrow = h]$.*

The second-order adaptive function concept allows deriving definitions of language classes based only on the SOAA characteristics. One of these classes is shown below.

Definition 5 (Confined Adaptive Problem) *Given a FOAA M^1 and a language L in which $L \neq L(M^1)$, let $C_\propto = (\mathbb{G}_{\psi_i}, \ldots, \mathbb{G}_{\psi_j}, \ldots, \mathbb{G}_{\psi_k})$ be a sequence of second-order adaptive functions. If the language L can be expressed in terms of M^1 and $C_\propto$ as follows in equation 21:*

$$L = L((\mathbb{G}_{\psi_i} \ldots (\mathbb{G}_{\psi_j} \ldots (\mathbb{G}_{\psi_k}(M^1)) \ldots) \ldots)) \tag{21}$$

*then L is called a **confined adaptive problem**, sequence $C_\propto$ is called **metamorphosis sequence** and M^1 is a **seed** for L.*

Definition 6 (Linear Confined Adaptive Problem Class)

Given a finite second-order adaptive functions set $\mathcal{G}$, let the indexable class $\mathcal{L}_\mathcal{G} = \{L_n\}_{n\in\mathbb{N}}$ of confined adaptive problems, all based on the same seed L_0, in which the metamorphosis of L_i is a subsequence of the L_{i+1} metamorphosis.

*For all languages $L_i \in \mathcal{L}_{\mathcal{G}}$, if all elements of the sequence $C_{\propto_i}$, belongs to L_i, are elements of $C_{\mathcal{G}}$, then $\mathcal{L}_{\mathcal{G}}$ is called a **Linear Confined Adaptive Problem Class** and set $\mathcal{G}$ is called a **mutation set**.*

Theorem 1 *Given a Confined Adaptive Problem L and any text θ of L, there is a SOAA M^2 and a Natural number $n > 0$, for which:*

$$\begin{cases} L \neq L(M^2) & for\ \theta[n] \\ L = L(M^2) & for\ \theta[n+1] \end{cases}$$

Proof (by construction). Take a SOAA in which its subjacent device M^1 is a seed for a Confined Adaptive Problem L. Let Ψ be a behavior set in which all elements of $C_\propto$ are elements of Ψ, too. With the valid seed M^1 for L, it is possible to define a second-order adaptive transition relation in which the computation of text θ assumes the form:

$$\begin{array}{l}
(q_0, t_1) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_s}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1 \rangle]} (q', w') \\
(q_0, t_2) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_t}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)' \rangle]} (q'', w'') \\
\ldots \\
(q_0, t_{n-1}) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_u}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{(n-2)} \rangle]} (q^{n-1}, w^{n-1}) \\
(q_0, t_n) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_w}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{(n-1)} \rangle]} (q^n, w^n) \\
(q_0, t_{n+1}) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_p}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{n} \rangle]} (q_f, \epsilon) \\
(q_0, t_{n+2}) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_v}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{(n+1)} \rangle]} (q_f, \epsilon) \\
\ldots \\
(q_0, t_{n+k}) \vdash^*_{[\langle \mathbb{G}_{\psi_{j_l}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{(n+(k-1))} \rangle]} (q_f, \epsilon)
\end{array}$$

for $q', q'', \ldots, q^{n-1}, q^n$ different from q_f; $w', w'', \ldots, w^{n-1}, w^n$ different from ϵ and

$$\begin{array}{l}
(M^1)' = \mathbb{G}_{\psi_{j_s}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1 \\
(M^1)'' = \mathbb{G}_{\psi_{j_t}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)' \\
\ldots \\
(M^1)^{n-1} = \mathbb{G}_{\psi_{j_u}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{n-2} \\
(M^1)^{n} = \mathbb{G}_{\psi_{j_w}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{n-1} \\
(M^1)^{n+1} = \mathbb{G}_{\psi_{j_p}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} (M^1)^{n} \\
\ldots
\end{array}$$

such that the execution of adaptive transitions generates the sequence of the adaptive transformations below

$$L = L(\mathbb{G}_{\psi_{j_p}} \ldots \mathbb{G}_{\psi_{j_w}} \ldots \mathbb{G}_{\psi_{j_u}} \ldots \mathbb{G}_{\psi_{j_t}} \ldots \mathbb{G}_{\psi_{j_s}} \ldots \mathbb{G}_{\psi_{j_2}} \mathbb{G}_{\psi_{j_1}} M^1)$$

and, for any t_{n+k} in which $k > 1$, the following expression holds

$$(M^1)^{(n+(k-1))} = (M^1)^{(n+1)}$$

Thus, for $n > 0$, there is a SOAA for the Confined Adaptive Problem such that

$$\begin{cases} L \neq L(M^2) & for\ (t_1, t_2, \ldots, t_n) \\ L = L(M^2) & for\ (t_{n+1}, \ldots) \end{cases}$$

□

Theorem 2 *For $m \in \mathbb{N}$, and given a function $P(M^2, \theta[m]) = M^1$ that returns M^1, which is the subjacent device FOAA of the second-order M^2, then after processing the segment text $\theta[m]$, there is a SOAA M^2 in which the function $P(M^2, .)$ is an IIM and P* ***Lim-identifies*** *any language L_i of $\mathcal{L}_{C_\propto}$, for any text θ belonging to L_i.*

Proof. As seen in theorem 1, for any Confined Adaptive Problem L, it is possible to construct a SOAA M^2 such that:

$$\begin{cases} L \neq L(M^2) & for\ \theta[n] \\ L = L(M^2) & for\ \theta[n+1] \end{cases}$$

Thus, it is possible to claim that

$$\exists(n \in \mathbb{N})\exists(M^1 \in \mathscr{M}^1 : L(M^1) = L)(\forall m \geq n)[P(M^2, \theta[m]) = M^1]$$

in other words,

$$\exists(M^1 \in \mathscr{M}^1 : L(M^1) = L)[P(M^2, \theta[m]) \downarrow= M^1]$$

meaning that the function $P(M^2, .)$ **Lim-identifies** L.

According to definition 6, the metamorphosis sequence of any L_i (with $i \geq 0$) of $\mathcal{L}_{C_\propto}$ is a subsequence of the metamorphosis sequence of L_z, with $z \geq i$. Thus, using theorem 2, the following assertions holds:

for L_0 $\exists(P(M_0^2, .) : P(M_0^2, .)$ **Lim-identifies** (L_0)
for L_1 $\exists(P(M_1^2, .) : P(M_1^2, .)$ **Lim-identifies** (L_0, L_1)
for L_2 $\exists(P(M_2^2, .) : P(M_2^2, .)$ **Lim-identifies** (L_0, L_1, L_2)
...
for L_i $\exists(P(M_i^2, .) : P(M_i^2, .)$ **Lim-identifies** $(L_0, L_1, L_2, \ldots, L_i)$
...
for L_n $\exists(P(M_n^2, .) : P(M_n^2, .)$ **Lim-identifies** $\mathcal{L}_{C_\propto}$

□

An important consequence of Theorem 2 has an immediate impact on the choice relation over the hypotheses space.

Corollary 1 *Set $\mathscr{M}^1$ is an admissible hypotheses space.*

4.1 Illustrating example

Let $\Sigma = \{a, b, c\}$ be an alphabet, and $t = abc$ a string over Σ^*. Based on the string t, it is possible to define the class of formal languages below:

$$I = \{L_0 = a^n b^n c, L_1 = a^n b c^n, L_2 = ab^n c^n\}$$

Now, consider the following situation: there is a text θ that belongs to an unknown language X. The only information about language X is the fact that it belongs to class I. The question is: would it be possible to obtain a SOAA that identifies the language X represented by sequence θ? If I is a Confined Adaptive Problem Class, then the response is yes. Thus, to answer the question, it is necessary to verify whether I is a Confined Adaptive Problem Class or not.

Proof (I is a Confined Adaptive Problem Class).
The proof that I is a Confined Adaptive Problem Class is lengthy. For space limitation reasons, only the proof sketch will be given here. All elements of I can surely be represented by FOAAs. Let M_0^1 represents the FOAA for the language L_0 of I. One possible adaptive function $\mathbb{F}_0$ for this FOAA is "for a number n of symbols 'a' recognized, transform the M^0 to accept the same number of symbols 'b' in the string".

The next step is to verify that all language members of I are Confined Adaptive Problems. Thus, it is necessary to verify if there are metamorphosis sequences for L_1 and L_2. If there are such metamorphosis sequences, then the first sequence is

$$C_{\propto_2} = (\mathbb{G}_2, \mathbb{G}_1)$$

and it performs the transformation sequence below:

$$L_2 = L((\mathbb{G}_2(\mathbb{G}_1(M_0^1))))$$

in which the two second-order adaptive functions $\mathbb{G}_2$ and $\mathbb{G}_2$ must have the following characteristics:

- $\mathbb{G}_1$: Replace the second symbol of $\mathbb{F}_0$ with the symbol 'c' and transform $\mathbb{F}_0$ in $\mathbb{F}_1$.
- $\mathbb{G}_2$: Replace the first symbol of $\mathbb{F}_1$ with the symbol 'b' and transform $\mathbb{F}_1$ in $\mathbb{F}_2$.

And the second sequence is

$$C_{\propto_1} = (\mathbb{G}_1)$$

that performs the transformation sequence below:

$$L_1 = L(\mathbb{G}_1(M_0^1))$$

But sequence $C_{\propto_1}$ is a subsequence of $C_{\propto_2}$ and generates the language L_1. Considering that the second-order adaptive functions set used to create the transformation sequences is finite, then it is a mutation set. Thus, I is a Confined Adaptive Problem.

□

5 Conclusion

As stated in the Introduction it is possible to define many ways to the hypothesis choice and each one, in practice, determines a whole new learning model; the main ones are the probabilistic approach and the enumerations strategies. The approach used in this work is closely related to Wallace's Minimum Message Length (MML) approach, but was inspired by Solomonoff's paper [11].

A strong connection between learning in the limit and the SOAA was shown by Theorems 1 and 2. The connection is established using Solomonoff's approach to mutating programs. The purpose is to represent a learning process using the SOAA, and this learner acts as a learning in the limit one. Thus, from this point of view, any hypothesis can be used to start a learning process, and, following a step-by-step transformation of that hypothesis by a SOAA, produces a correct final model, when computational learning can be effective.

Hence, inductive inference can be envisioned in a new and different way using this kind of learner. The SOAA can be used as a learner for formal languages, as illustrated by the example in section 4.1. There are many applications for the learning process defined in this paper; one clearly comes from on-line learning. Since it is a non-stop process, it is suitable for continuous learning, and as the previous hypothesis can be used to produce new ones, the second-order adaptive automaton seems to be an appropriate choice for diverse environments.

5.1 Future work

As a future work, some of the applications mentioned here will be implemented, to run them in order to generate a benchmark comparing the second-order adaptive approach to some others. A lot of work need to be done before a product ready to be used can be generated, but the path to be followed has been established.

It is necessary to define all the limitations of the computational model and then define the learning limitations of this adaptive learning process. Some constraints and limits of the adaptive automata hierarchy have to be formally defined. For the purposes of this work, the second-order was sufficient. Another task to be carried out is investigate the necessity of third or higher order.

References

1. Nick Chater and Paul Vitányi. "Ideal learning" of natural language: positive results about learning from positive evidence. *Journal of Mathematical Psychology*, 51(3):135–163, 2007.
2. D. L. Dowe. *Handbook of the Philosophy of Science - (HPS Volume 7) Philosophy of Statistics*, chapter MML, hybrid Bayesian network graphical models, statistical consistency, invariance and uniqueness, pages 901–982. Elsevier, 2011.
3. E. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
4. Ming Li and Paul Vitányi. *Computer Science Today*, volume 1000 of *Lecture Notes in Computer Science*, chapter Computational Machine Learning in Theory and Praxis, pages 518–535. Springer, 1995.
5. Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer Publishing Company, Incorporated, third edition, 2008.
6. W.J. Paul and R.J. Solomonoff. Autonomous theory building systems. Technical Report D 6600, Computer Science Department; University of Saarbruecken; Germany, 1990.
7. Ricardo Luis de Azevedo da Rocha and João José Neto. Adaptive automaton, limits and complexity compared to the Turing machine - in Portuguese Autômato Adaptativo, Limites e Complexidade em Comparação com Máquina de Turing. In *Proceedings of the I Congress of Logic Applied to Technology - LAPTEC 2000*, pages 33–48, São Paulo, Oct 2000. Faculdade SENAC de Ciências Exatas e Tecnologia.
8. Roy S. Rubinstein and John N. Shutt. Self-modifying finite automata. In B. Pehrson and I. Simon, editors, *Proceedings of the 13th IFIP World Computer Congress*, volume I of *Technology and Foundations: Information Processing '94*, pages 493–498, Amsterdam, 1994. North-Holland.
9. Stanley Salthe and Koichiro Matsuno. Self-organization in hierarchical systems. *Journal of Social and Evolutionary Systems*, 18(4):327–338, 1995.
10. Reginaldo Inojosa da Silva Filho and Ricardo Luis de Azevedo da Rocha. *International Conference on Adaptive and Natural Computing Algorithms, ICANNGA 2011, Ljubljana, Slovenia, April 14-16, 2011, Revised Papers*, volume 6594 of *Lecture Notes in Computer Science*, chapter Adaptive Finite Automaton: a New Algebraic Approach, pages 275–284. Springer-Verlag, 2011.
11. R. J. Solomonoff. A formal theory of inductive inference. parts I and II. *Information and Control*, 7(2):224–254, 1964.
12. R. J. Solomonoff. Does algorithmic probability solve the problem of induction? In D. L. Dowe, K. B. Korb, and J. J Oliver, editors, *Proceedings of the Information, Statistics and Induction in Science (ISIS) Conference*, pages 7–8, Melbourne, Australia, August 1996. World Scientific.
13. C. S. Wallace. *Statistical and Inductive Inference by Mininum Message Length*. Springer, 2005.