Efficiency in the Identification in the Limit Learning Paradigm

Rémi Eyraud and Jeffrey Heinz and Ryo Yoshinaka

Abstract Two different paths exist when one is interested in validating an idea for a learning algorithm. On the one hand, the practical approach consists in using the available data to test the quality of the learning algorithm (for instance the widely used cross-validation technique). On the other hand, a theoretical approach is possible by using a learning paradigm, which is an attempt toformalize what learning means. These models provide a framework to study the behavior of a learning algorithm and allows to formally show the soundness of an approach. The mainly used learning paradigm in Grammatical Inference is the one of Identification in the limit. But its first definition is not restrictive enough in the sense that no efficiency bound is required. This chapter surveys the different refinements that have been developed and studied. Main results for each formalisation are given and comparisons are detailled.

1 Introduction

E. M. Gold [10] introduced in the sixties the notion of identification in the limit. The principle is to consider that the algorithm is fed with an infinite sequence of data corresponding to a target language. When a new element is given to the algorithm, it outputs an hypothesis. The algorithm identifies the language in the limit if for any possible sequence of data for this language, there exists a moment from when the algorithm does not change its hypothesis, and this hypothesis is a correct representation of the target language. When a whole class of languages is consid-

Name of First Author
Name, Address of Institute, e-mail: name@email.address

Name of Second Author
Name, Address of Institute e-mail: name@email.address

Name of Second Author
Name, Address of Institute e-mail: name@email.address

ered, the algorithm identifies the class in the limit if it can identify all languages of the class.

The fact that the convergence is required to hold whatever the sequence of data is makes this paradigm be adversarial. This worst-case scenario principle strengthens the value of any algorithmic idea that can yield to an identification in the limit result for a class of languages.

However, Gold's formulation can be of little help for practical purpose, when one wants to study an learning idea with the aim to apply it to real-world data. This is mainly due to the fact that no efficiency property is required and thus one can assume infinite time and space. This is the reason why several refinements of Gold's model have been developed, adding polynomial bounds to the requirements of the paradigm.

After a short section containing the needed definitions, Section 3 studies the limitations of the initial identification in the limit definition. In Section 4 requirements based on the running time of the studied algorithm are developed. Section **??** deals with efficiency requirements depending on the iterative behaviour of the algorithm while Section 6 is a bout a set driven refinement of Gold's paradigm. Then, Section **??** introduces two recent reformulation of the paradigm. Finally, Section 8 briefly discusses other paradigms.

2 Preliminary definitions

An *alphabet* Σ is a finite nonempty set of symbols called *letters*. A *string* w over Σ is a finite sequence $w = a_1a_2\dots a_n$ of letters. Let $|w|$ denote the length of w. Given a set of strings S, we denote $|S|$ its cardinality and $||S||$ its size, i.e. the sum of the length of the strings it contains. In the following, letters will be indicated by $a,b,c,\dots$, strings by $u,v,\dots,z$, and the empty string by λ. Let Σ^* be the set of all strings.

We assume a fixed but arbitrary total order $<$ on the letters of Σ. As usual, we extend $<$ to Σ^* by defining the *hierarchical order* [13], denoted by $\lhd$, as follows:

$$\forall w_1, w_2 \in \Sigma^*, w_1 \lhd w_2 \textit{ iff } \begin{cases} |w_1| < |w_2| \text{ or} \\ |w_1| = |w_2| \text{ and } \exists u, v_1, v_2 \in \Sigma^*, \exists a_1, a_2 \in \Sigma \\ \textit{s.t. } w_1 = ua_1v_1, w_2 = ua_2v_2 \text{ and } a_1 < a_2. \end{cases}$$

$\lhd$ is a total strict order over Σ^*, and if $\Sigma = \{a,b\}$ and $a < b$, then $\lambda \lhd a \lhd b \lhd aa \lhd ab \lhd ba \lhd bb \lhd aaa \lhd \dots$

We extend this order to non-empty finite sets of strings: $S_1 \lhd S_2$ iff $||S_1|| < ||S_2||$ or $||S_1|| = ||S_2||$ and $\exists w \in S_1$ such that $\forall w' \in S_2$ either $w' \in S_1$ or $w \lhd w'$.

By a language L over Σ we mean every subset $L \subseteq \Sigma^*$. Many classes of languages were investigated in the literature. In general, the definition of a class $\mathbb{L}$ relies on a class $\mathbb{R}$ of abstract machines, here called *representations*, that characterize all and only the languages of $\mathbb{L}$. The relationship is given by the *naming function* $\mathscr{L} : \mathbb{R} \to \mathbb{L}$

such that: (1) $\forall R \in \mathbb{R}, \mathscr{L}(R) \in \mathbb{L}$ and (2) $\forall L \in \mathbb{L}, \exists R \in \mathbb{R}$ such that $\mathscr{L}(R) = L$. Two representations R_1 and R_2 are *equivalent iff* $\mathscr{L}(R_1) = \mathscr{L}(R_2)$.

The *size* of a representation R, denoted by $\|R\|$, is polynomially related to the size of its encoding.

A lot of different classes of representations have been studied in the litterature and it is behond the scope of this chapter to give an exhautive list of them. However, we introduce the following definition, that is a generalisation of some of the well-known classes of grammar. We will maily focus on the classes of representations whose characterisation can be done in this context.

Definition 1 (Generative grammar). $G =< \Sigma, N, P, I >$ where Σ is the alphabet of the language, N is a set of variables usually called non-terminals, $P \subset (N \cup \Sigma)^* \times (N \cup \Sigma)^*$ is the set of generative rules, I is the finite set of axioms, which are elements of $(\Sigma \cup N)^*$.

A generatives rule (α, β) is usually denoted $\alpha \rightarrow \beta$. It allows the rewritten of elements of $(\Sigma \cup N)^*$ into elements of $(\Sigma \cup N)^*$. Given $\gamma \in (\Sigma \cup N)^*$ we say that a rule $\alpha \rightarrow \beta$ applied to γ if it exists $\eta, \delta \in (\Sigma \cup N)^*$ such that $\gamma = \eta \alpha \delta$. The result of applying this rule on γ is $\eta \beta \delta$. We write $\gamma \Rightarrow \eta \beta \delta$. $\Rightarrow^*$ is the reflexive and transitive closure of $\Rightarrow$.

Definition 2 (Generated language). Let $G =< \Sigma, N, P, I >$ be a generative grammar. $\mathscr{L}(G) = \{w \in \Sigma^* : \exists \alpha \in I \; s.t \; \alpha \Rightarrow_P^* w\}$.

Example 1. The usual classes of the Chomsky hierarchy are classes of generative grammars. Regular grammars correspond to the restriction $P \subset N \times \Sigma N \cup \{\lambda\}$, or $P \subset N \times N\Sigma \cup \{\lambda\}$ by symetry. The context-free grammars are the ones where $P \subset N \times (\Sigma \cup N)^*$ while the context-sensitive grammars are the ones such that if $\alpha \rightarrow \beta \in P$ then $\exists (\gamma, \delta, \eta) \in (\Sigma \cup N)^*, A \in N$: $\alpha = \delta A \eta$ and $\beta = \delta \gamma \eta$. If no restriction is imposed to the rules of the grammar, then the considered class is the one of recursive grammars. All of these classes were formerly defined with a set of axioms reduced to one element of N.

Example 2. String Rewriting Systems (SRS) [5] are generative devices where $N = \emptyset$. A Rule corresponds to an element of Σ^* rewritten into an element of Σ^* and the set of axioms is made of elements of Σ^*. The language representing by a SRS is the set of strings that can be rewritten using the rules from an element of I.

Some classes of representation that have been studied in grammatical inference are not coverted by Definition 1. This is the case for instance of the Multiple Context-Free Grammars and of the Paralel Context-Free Grammar. However, it is easy to generalise the definition in order to cover these classes: for the sake of simplicity we chose to consider this restrictive version.

3 The limits of Gold's paradigm

Introduce IIL, give general results (superfinite from text, all computably enumerable from informant)

We now detailled the formalisation of the identification in the limit paradigm.

A presentation P of a language L is an infinite sequence of data corresponding to L. We note $P[i]$ the i^{th} element of P and P_i the set of the i^{th} first elements of P. If the data contains only unstructured elements of L the presentation is called a *text* of language L. A text T is a complete presentation of L iff for all $w \in L$ it exists $n \in \mathbb{N}$ such that $T[n] = w$. If the data is made of couple (w,l), $w \in \Sigma^*$ such that l is a boolean valued true if $w \in L$ and false otherwise, then the presentation is called an *informant*. An informant I is a complete presentation of L iff for all $w \in \Sigma^*$ there exists $n \in \mathbb{N}$ such that $I[n] = (w,l)$. In the rest of the chapter, we will only consider complete presentations.

Definition 3 (Identification in the limit [10]). A class $\mathbb{L}$ of languages is *identifiable in the limit (IIL)* from text [resp. from informant] if and only if there exists an algorithm $\mathfrak{A}$ such that for all language $L \in \mathbb{L}$, for all text T [resp. informant I] of L,

- there exists an index N such that $\forall n \geq N$, $\mathfrak{A}(T_n) = \mathfrak{A}(T_N)$ [resp. $\mathfrak{A}(I_n) = \mathfrak{A}(I_N)$]
- $\mathscr{L}(\mathfrak{A}(T_N) = L$ [resp. $\mathscr{L}(\mathfrak{A}(I_N) = L$]

One of the main results in this paradigm is that no superfinite class of languages can be identified in the limit from text. Despite what the name can evocate, a class of languages is superfinite if it contains all finite languages and at least one infinite language (the class contains thus an infinite number of languages). The proof relies on the fact that given a presentation of an infinite language L, there does not exist any index N from which a learner can distinguish the finite language made of the strings seen so far and the infinite language: if it guesses the finite language it is making an error; but if its hypothesis corresponds to L, the presentation seen so far can also be the one of the finite language, which yields to an error if this language is the target one.

On the other hand, any computably enumerable class of languages is identifiable in the limit from informant. The learning algorithm is really naive: it enumerates the elements of the class until finding the first one consistent with the strings seen so far, that is to say the first language of the enumeration that accepts all positive example (labelled true) and rejects all negative ones (labelled false). If it is the correct hypothesis, the algorithm has converged. If not, then they will be a exemple later in the informant that is in contradiction with the current hypothesis and will make the algorithm to continue the enumeration to the next consistent language.

This second result, though of positive nature, is one of the main reason the identification in the limit paradigm was refined. Indeed, the briefly detailled learning algorithm is clearly not tractable and thus is of little use. This first formalisation is thus not enough restrictive to completely validate learning ideas.

ref to John Case chapter ?

4 Polynomial time

make it a subsection of the former one?

Given the limitations of IIL shown in Section 3, designing requirements to add to the pardigm is needed to strengthen the validity of learning idea. An intuitive way to deal with that, is to add a constraint on the computation time of the algorithm.

Taking into account the overall running time seems to be tricky since it would yield to consider only pathological presentations where the convergence happens late. A recent paper from Thomas Zeugmann [21] would be a great interest to a reader concerned by that standpoint.

A more consensual requirement is the update time efficiency. An algorithm is update time efficient if it outputs a new hypothesis in a time polynomial in the size of the data seen so far. This generate a new grammar, the amount of time used has to be a polynomial in the sum of the length of the strings available at that point.

But, in a seminal paper [14], Leonard Pitt shows that this requirement is not sufficient to prove the validity of a learning approach. Indeed, using a method now known as Pitt's trick, he proves that any algorithm that identify a class in the limit can be transformed into an algorithm that keep the property of the previous and that a polynomial update time. Unformally the proof relies on the fact that, given a presentation P, if a learner converges to a correct hypothesis on the initial sequence P_i, a variant can delay the computation of any interesting hypothesis until having seen P_j such that the computation time of the initial learner on P_i is polynomial in $||P_j||$. The variant algorithm has then an efficient update time and fullfils the conditions of identification in the limit.

As a consequence, an algorithm might be able to efficiently output an hypothesis, but if the convergence can only occur if a non-reasonable amount of data is provided, the theoritecal result will not be usefull when real data are taken into account.

5 Implicit errors of prediction and mind changes

Despite the problem described in the previous section, the requirement of a polynomial update time is still desirable. Therefore the paradigm has to be enriched such that delaying tricks are not possible.

Most additional requirements are based on the same method: theylink the behavior of the algorithm to the size of a target representation. Indeed, though the identification of the target language is expected, the polynomiality cannot rely on the language itself: non-trivial classes of languages often contain an infinite number of infinite languages. In addition, this focuses the attention on the hypothesis space of an algorithm, which is relevant form a machine learning standpoint[1].

[1] We do not consider here that the class of possible hypothesis corresponds to the target class of languages: we just emphasize the fact that choosing a target class of representations is likely to have consequences on the hypothesis space of the algorithm.

The first attempts to formalize this notion of convergence from a reasonable amount of data rely on the number of mind changes [3, 1] or the number of implicit prediction errors [14]. The first one requires that the number of times the current hypothesis disagrees with the new data is bounded by a polynomial in the size of the target representation. However, if the presentation is a text, a unwanted trick can be used: the algorithm can output a large hypothesis, for instance Σ^*, that will never be in contradiction with the data. It can then wait to see an important number of example before returning a pertinent hypothesis. This thus do not avoid Pitt's trick in this case.

The second one states that the number of time the current hypothesis is in contradiction with the new example has to be polynomial in the size of the target representation. More formally:

Definition 4 (Identification in Polynomial Number of Implicit Errors).

- Given a presentation P, an algorithm $\mathfrak{A}$ makes an implicit error of prediction at step n if $\mathfrak{A}(P_n)$ is in contradiction with $P[n]$.
- A class $\mathbb{G}$ of representations is polynomial-time identifiable in the limit in Pitt's sense if $\mathbb{G}$ admits a polynomial time learning algorithm $\mathfrak{A}$ such that for any presentation of $\mathscr{L}(G)$ for $G \in \mathbb{G}$, $\mathfrak{A}$ makes implicit errors of prediction at most polynomial in $||Gk||$ [14].
- A class $\mathbb{G}$ of representations is polynomial-time identifiable in the limit in Yokomori's sense if $\mathbb{G}$ admits a polynomial time learning algorithm $\mathfrak{A}$ such that for any presentation P of $\mathscr{L}(G)$ for $G \in \mathbb{G}$, for any natural number n, the number fo implicit errors of prediction made by $\mathfrak{A}$ on the n^{th} first examples is bounded by a polynomial in $m \cdot ||G||$, where $m = max\{|P[1], \ldots, |P[n]|\}$ [19].

Notice that Yokomori's formulation is a relaxed version of the one of Pitt.

main results, comparison. Do we need to talk about conservatism, fairness, etc?

But both of these formulations suffer from the same drawback: they are mainly designed for incremental algorithms. Indeed, these paradigms give a lot of importance to the sequence of data, in particular as the parts of two sequences that contains the same elements in a different order might not correspond to the same number of implicit errors (or mind changes). This eventually yields to consider particularly malevolent sequences of data. However, in most practical framework, for instance in Natural Language Processing or Bio-Informatics, we are interested by algorithms that work from a finite *set* of data.

6 Characteristic Sample

The most widely used definition of data efficiency relies on the notion of characteristic sample, that is to say a set of data that ensures the correct convergence of the algorithm as soon as it is present in the set of data seen by the algorithm. In this paradigm [7], it is required that the algorithm needs a characteristic sample whose size[2] is polynomial in the size of the target representation.

This refinement diverges from the usual identification in the limit approach as the iterative process in not any more the core of the paradigm. Indeed, the fact that it relies on a characteristic sample allows a set-driven definition that we are developping here[3]. We thus need first the two following definitions that concerns non-iterative learners.

Definition 5. Let $\mathbb{L}$ be a class of languages represented by some class $\mathbb{R}$ of representations.

1. A *sample* S for a language $L \in \mathbb{L}$ is a finite set of data corresponding to L. A *positive sample* for L is made only of elements of L. A positive and negative sample for L is made of couples (w,l), where l is a boolean such that $l = \text{TRUE}$ if $w \in L$ and $l = \text{FALSE}$ otherwise. The *size* of a sample S is the sum of the size of all its elements.
2. An $(\mathbb{L},\mathbb{R})$-learning algorithm $\mathfrak{A}$ is a program that takes as input a sample for a language $L \in \mathbb{L}$ and outputs a representation from $\mathbb{R}$.

Notice that these definitions do not specified whether the data are strings or structural data like trees, skeletons or graphs.

The underlying idea of the paradigm is that it is impossible to learn if the data seen so far does not contain enough information about the target. This paradigm also deals with the relevance of the class of representations, as the characteristic sample is usually described using the target representation, different classes of representation for the same class of laguages might not obtain the same learning result.

We first need to define the following notion:

Definition 6 (Characteristic sample). Given a $(\mathbb{L},\mathbb{R})$-learning algorithm $\mathfrak{A}$, we say that a sample CS is a *characteristic sample* of a language $L \in \mathbb{L}$ if for all samples S such that $CS \subseteq S$, $\mathfrak{A}$ returns a representation R such that $\mathscr{L}(R) = L$.

The set-driven version of the paradigm can now be given:

Definition 7 (Set-driven identification in polynomial time and data [7]). A class $\mathbb{L}$ of languages is *identifiable in polynomial time and data (IPTD)* from a class $\mathbb{R}$ of representations if and only if there exist an $(\mathbb{L},\mathbb{R})$-learning algorithm $\mathfrak{A}$ and two polynomials $p()$ and $q()$ such that:

1. Given a sample S for $L \in \mathbb{L}$ of size m, $\mathfrak{A}$ returns a hypothesis $H \in \mathbb{R}$ in $\mathcal{O}(p(m))$ time ;
2. For each representation R of size k of a language $L \in \mathbb{L}$, there exists a characteristic sample of L of size at most $\mathcal{O}(q(k))$.

[2] The size of a sample is the sum of the length of its elements: it has been shown [14] that its cardinality is not a relevant feature when efficiency is considered, as it creates a risk of collusion.**Remi: collusion=Pitt's trick? Is CS enough or do we need constistent to get rid of the trick? I think it is, but we might need to say it**

[3] Notice that it can also be defined in an iterative way.

Several reasons explain why we chose this unusual way to formalize identification. First, the aim here is to formalize learning when a set of data is available, which corresponds to the most common framework when real-world data is considered. This is the reason why, real iterative algorithms, that is to say algorithms that only used their previous hypothesis and the new data to generate a new hypothesis, though of great interest, are not of central in this chapter. However, notice that, an algorithm defined within the paradigm of Definition 7 can easily be studied in the incremental one: it suffices to define a new algorithm that for each new data launches the first one on the set of data seen so far.

By forcing the algorithm to converge to a correct hypothesis whenever a characteristic sample of reasonable size has been seen, this paradigm tackles the risk of collusion by forbidding Pitt's delaying trick. For a recent and detail discussion on this problematics, see for instance [6].

main results, comparison with the previous ones

6.1 Limitations

The identification in polynomial time and data suffers from one main drawback: it has been thought in the context of regular languages learning, and thus is not shaped for more complex class of languages. Example 3 proves that context-free languages cannot be learned under this criterion using context-free grammars. Indeed, the characteristic sample has to contain the only string in the language, but the length of this string is exponentially greater than the grammar.

Example 3. Let n be a fixed natural number and let G_1 be the context-free grammar whose production rules are $N_i \rightarrow N_{i+1}N_{i+1}$, for $0 \leq i < n$, and $N_n \rightarrow a$. The language of this grammar is the singleton $L(G_1) = \{a^{2^n}\}$.

The reason why this example is not learnable does not come from the hardness of the language: it is made of only on string. But the use of any class of representation that contains this example is not identifiable in the limit. It seems that in this case the problem comes from the definition of what learning means, that is to say from the learning criterion, rather than the properties of the language. Hence, from an information theory point of view, it is obviously interesting to have an algorithm that is able to find a model explaining the data it is fed with that is exponentially smaller than these data. This is actually a desired property in many fields of machine learning (see [9] for instance).

The trouble here comes thus from the learning paradigm.

7 Thickness and Structurally complete

definitions of the formalisms, comparisons, limitations.

7.1 Thickness

In a recent paper [20], Ryo Yoshinaka introduced the identification from an characteristic sample whose size is a polynomial in the size of the target grammar and of a measure called the thickness of the grammar.

Definition 8 (Thickness). Let $G = \langle \Sigma, N, P, I \rangle$ be a generative grammar. The *thickness* of G is $\tau_G = max\{|\omega(\alpha)| : \exists\beta, \alpha \to \beta \in P\}$ where $\omega(\alpha) = min\{w \in \Sigma^* : \alpha \Rightarrow_G^* w\}$.

This definition is an extended version of the one that was first introduced for context-free grammars in the context of model complexity [18]. Notice that it has nothing to do with the usual notion of thickness in learning theory.

Definition 9 (Identification with Thickness Polynomiality [20]). A class $\mathbb{L}$ of languages is *identifiable in polynomial time and thick data (IPTtD)* for a class $\mathbb{R}$ of representations if and only if there exists an algorithm $\mathfrak{A}$ and two polynomials $p()$ and $q()$ such that:

1. Given a sample S for $L \in \mathbb{L}$ of size m, $\mathfrak{A}$ returns a hypothesis $H \in \mathbb{R}$ in $\mathcal{O}(p(m))$ time ;
2. For each representation R of a language $L \in \mathbb{L}$ of size k, there exists a *characteristic sample CS* whose size is in $\mathcal{O}(q(k, \tau_R))$.

7.2 Structurally complete set

We first introduced the following definition:

Definition 10 (Structually Complete Set). Given a generative grammar G, a structurally complete set (SCS) for G is a set of data SC such that for each production $\alpha \to \beta$, there exists an element $x \in SC$, an element $\gamma \in I$ and two elements $\eta, \tau \in (\Sigma \cup N)^*$ such that $\gamma \Rightarrow^* \eta\alpha\tau \Rightarrow \eta\beta\tau \Rightarrow^* x$. The smallest structurally complete set SC for a grammar G is the sample such that for all SCS SC' for G, $SC \lhd SC'$.

A notion of structurally complete sample has already been defined in the context of regular language learning [8]. However, this former definition relied on a particular representation, namely the finite state automaton, and it considered only the case of positive and negative examples. Definition 10 is more general as it does not depend on a particular representation and does not consider a particular type of data.

Definition 11 (Polynomial Structuraly Complete Identification). A class $\mathbb{L}$ of languages is *identifiable in polynomial time and structurally complete data (ITscD)* for a class $\mathbb{R}$ of representations if and only if there exist an algorithm $\mathfrak{A}$ and two polynomials $p()$ and $q()$ such that:

1. Given a sample S for $L \in \mathbb{L}$ of size m, $\mathfrak{A}$ returns a hypothesis $H \in \mathbb{R}$ in $\mathcal{O}(p(m))$ time ;
2. For each representation R of a language $L \in \mathbb{L}$, there exists a *characteristic sample CS* whose size is in $\mathcal{O}(q(k))$, where k is the size of the smallest structurally complete set for R.

Notice that in the case where negative data is also available, the size of the characteristic sample has to be polynomial in the size of a SCS which contains only positive examples. This implies that the amount of negative evidence has to be polynomialy related to the one of the positive evidence.

7.2.1 Comparison with IPTD

Consider the class of languages $\mathscr{L}_1 = \cup_{n\in\mathbb{N}} \{a^i : 0 \leq i \leq 2^n\}$. This class is identifiable in polynomial time and data from positive data only using the class of representations $\mathscr{G}_1 = \cup_{n\in\mathbb{N}} \langle\{a\}, \{S,A\}, \{S \to A^{2^n}, A \to a|\lambda\}, \{S\}\rangle$. Indeed, given a target language, the simple algorithm that returns the only grammar consistent with a sample admit the characterisric sample $\{a^{2^n}\}$ which is linear in the size of the target. However, the smallest structurally complete set of any target grammar is $\{\lambda, a\}$ which is of size 2. As the size of the smallest SCS is constant and the class of languages infinite, $\mathscr{L}_1$ is not identifiable in polynomial time and structurally complete data.

On the other hand, let consider the class of languages $\mathscr{L}_2 = \cup_{n\in\mathbb{N}} \{a^{2^n}\}$ and its class of representations $\mathscr{G}_2 = \cup_{n\in\mathbb{N}} \langle\{a\}, N_n, P_n, \{N_0\}\rangle$, with $P_n = \{N_n \to a\} \cup_{0\leq i<n} \{N_i \to N_{i+1}N_{i+1}\}$. Given n, the characteristic sample is $\{a^{2^n}\}$ which is also the smallest structurally complete set for the target grammar. However, this sample is not polynomial in the size of the target grammar. Therfore $\mathscr{L}_2$ is identifiable in the limit in polynomial time and structurally complete data using $\mathscr{G}_2$ but not in polynomial time and data.

This show that these two paradigms are thus non-comparable. In the next section we show that the main language classes studied under the former paradigm are identifiable in polynomial time and structurally complete data. Moreover, some classes that were not learnable in de la Higuera sense are shown to be identifiable in the new paradigm.

7.3 *Comparison of the two refinements*

Ryo's counter-example shows that a class can be IPTtD but not IPTscD. Is the converse true?

8 Other learning paradigms in GI

PAC: why it is not adapted to GI, main results in restrictive version. The most used paradigm in machine learning is the Probably Approximately Correct (PAC) criterium [16] and its refinements [12, 15]. However, these paradigms are usually considered as not adapted to formal languages learning, as even very simple and well characterized classes of languages are not PAC-learnable [2].

Several theoretical reasons explain this inadequacy, one of the main ones being that the VC-dimension of even the simplest models of language representations, namely the finite state automata, is not bounded [11] which make them not learnable in the PAC sense [4]. This is closely related to the fact that the learning principle of empirical risk minimization [17], inherent in most approaches studied under the PAC framework, is of little use when formal languages are considered. Indeed, the number of representations consistent to a given set of data of a target language, that is to say representations that correctly explain all the data, is often non finite. It is then useless to reduce the hypothesis space to the one that minimize the error on a given set of data.

Another reason is that a representation of a formal language is not only a classifier, that is to say a device that defines what is in the language and what is not, but it gives also structural information about the elements of the language.

Another particularity of language learning is that a lot of algorithms use only positive examples of a target concept, while the usual machine learning framework relies on labelled data. In addition, the PAC paradigm is particularly pertinent in the case of statistical models where the probability of making a mistake can be evaluated using the hypothesis, but it is of less interest for non stochastic model learning.

On the other hand, the PAC paradigm does not suffer from the main drawback of identification in the limit that it is of being asymptotic: no guarantee is provided about the quality of the hypothesis before the convergence happens. But this drawback seems to be inherent of the kind of representations for the learning targets considered: even if two generative grammars have 99.9% of their rules in common, the languages of these two grammars can be as far apart as one wishes. This problem is inherent to the nature of formal languages and their representations and this "Gestalt-like" property is unavoidable in the formalization of learning: the whole grammar is more than the sum of its rules. In our view, this mainly justifies the use of identification in the limit in the context of grammar learning.

Zeugmann's stochastic finite learning? others?

References

1. A. Ambainis, S. Jain, and A. Sharma. Ordinal mind change complexity of language identification. pages 323–343, 1999.
2. D. Angluin, J. Aspnes, and A. Kontorovich. On the learnability of shuffle ideals. In *Algorithmic Learning Theory, 23rd International Conference, ALT 2012, Lyon, France, October*

29-31, 2012. Proceedings, volume 7568 of *Lecture Notes in Computer Science*, pages 111–123. Springer-Verlag, 2012.

3. L. E. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28(2):125–155, 1975.
4. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.
5. R. Book and F. Otto. *String-Rewriting Systems*. Springer Verlag, 1993.
6. J. Case and T. Kötzing. Difficulties in forcing fairness of polynomial time inductive inference. In *Proceedings of ALT'09*, volume 5809 of *LNCS*, pages 263–277, 2009.
7. C. de la Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning Journal*, 27:125–138, 1997.
8. P. Dupont, L. Miclet, and E. Vidal. What is the search space of the regular inference? In R. C. Carrasco and J. Oncina, editors, *Proceedings of ICGI'94*, number 862 in LNAI, pages 25–37. Springer-Verlag, 1994.
9. F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Comput.*, 10(6):1455–1480, 1998.
10. E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
11. Y. Ishigami and S. Tani. VC-dimensions of finite automata and commutative finite automata with *k* letters and *n* states. *Discrete Applied Mathematics*, 74:123–134, 1997.
12. M. Li and P. Vitanyi. Learning simple concepts under simple distributions. *Siam Journal of Computing*, 20:911–935, 1991.
13. J. Oncina and P. García. Identifying regular languages in polynomial time. In *Advances in Structural and Syntactic Pattern Recognition*, volume 5 of *Series in Machine Perception and Artificial Intelligence*, pages 99–108. 1992.
14. L. Pitt. Inductive inference, DFA's, and computational complexity. In *Analogical and Inductive Inference*, number 397 in LNAI, pages 18–44. Springer-Verlag, 1989.
15. J. Shawe-Taylor and R. Williamson. A pac analysis of a bayesian estimator. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 2–9, 1997.
16. L. G. Valiant. A theory of the learnable. *Communications of the Association for Computing Machinery*, 27(11):1134–1142, 1984.
17. V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
18. M. Wakatsuki and E. Tomita. A fast algorithm for checking the inclusion for very simple deterministic pushdown automata. *IEICE TRANSACTIONS on Information and Systems*, VE76-D(10):1224–1233, 1993.
19. T. Yokomori. On polynomial-time learnability in the limit of strictly deterministic automata. *Machine Learning Journal*, 19:153–179, 1995.
20. R. Yoshinaka. Identification in the limit of k, l-substitutable context-free languages. In *ICGI*, pages 266–279, 2008.
21. T. Zeugmann. Can learning in the limit be done efficiently? In *Proceedings of ALT'03*, pages 17–38, 2003.