Machine Learning 27, 125–138 (1997)

Characteristic Sets for Polynomial Grammatical Inference

COLIN DE LA HIGUERA delahiguera@lirmm.fr
Département d'Informatique Fondamentale (DIF) LIRMM, 161 rue Ada, 34 392 Montpellier Cedex 5, France
http ://www.lirmm.fr/∼cdlh

Editor: Wolfgang Maass

Abstract. When concerned about efficient grammatical inference two issues are relevant: the first one is to determine the quality of the result, and the second is to try to use polynomial time and space. A typical idea to deal with the first point is to say that an algorithm performs well if it infers *in the limit* the correct language. The second point has led to debate about how to define polynomial time: the main definitions of polynomial inference have been proposed by Pitt and Angluin. We return in this paper to a definition proposed by Gold that requires a characteristic set of strings to exist for each grammar, and this set to be polynomial in the size of the grammar or automaton that is to be learned, where the size of the sample is the sum of the lengths of all strings it includes. The learning algorithm must also infer correctly as soon as the characteristic set is included in the data. We first show that this definition corresponds to a notion of teachability as defined by Goldman and Mathias. By adapting their teacher/learner model to grammatical inference we prove that languages given by context-free grammars, simple deterministic grammars, linear grammars and nondeterministic finite automata are not identifiable in the limit from polynomial time and data.

Keywords: exact identification, grammatical inference, polynomial learning

1. Introduction and related work

The problem of describing polynomial paradigms for learning has received much attention in learning theory community. We focus our attention on those paradigms dealing with grammatical inference, i.e., the inference, from strings, of grammars or automata. As most of the literature deals with the inference of deterministic finite automata (DFA), the following general discussion concentrates on them.

In his seminal paper, Pitt (1989) discusses different possible ideas as to what polynomial complexity for the problem of exact identification of (DFA) should be. The model he analyses is Gold's classical model (1967): a presentation of the language is given, in which strings appear with a label (+ if it is a positive instance, and − if it is a negative instance). A presentation is required to be complete, that is each example appears at least once. In this model an algorithm is said to identify in the limit *iff* on input of any complete presentation of a language, the algorithm at some point converges to a correct representation of the language. Pitt discards the possibility of time being polynomial in the size of the

This work has been performed while the author was visiting U.P. Valencia, Spain.

representation to be learned, as we have no control over the presentation of the examples, so the very first example can be too big. He equally refuses polynomial update time (the complexity takes into account the sizes of the examples that have been seen so far) because the exhaustive search strategy can perform as follows: "when no time remains, delay the treatment of some examples for later". We note that using polynomial update time with Pitt's trick, the delaying of the treatment of the examples means that the algorithm has not inferred correctly; and at some point it has sufficient information to give the correct answer but does not.

Thus, Pitt proposes another measure of complexity: for an identification algorithm to be polynomial it must have polynomial update time, and also make a polynomial number of implicit errors (in the size of the automaton). An implicit error is made when the current hypothesis does not agree with a new example. This definition alas is shown (by Pitt) to be very restrictive, in the sense that even DFA do not comply with the conditions, so no superclasses of regular languages allow polynomial time inference.

A second model of learning has been proposed by Angluin (1987) and exhaustively studied since. The presentation of the language is not arbitrary, and can be somehow controlled by asking queries of an oracle. The two most important sorts of queries are the membership queries and equivalence queries. In a membership query a string is proposed to the oracle which returns its correct classification. An equivalence query consists in proposing a representation to the oracle, which either accepts it as a correct representation of the language to be inferred, or returns a counter-example, that is a string from the symmetric difference of the proposed language and the target one. This is known as the MAT model (Minimally Adequate Teacher). With time complexity dependent on the size of the automaton to be inferred and the length of the longest counter-example returned by the oracle, Angluin proves that DFA can be identified in polynomial time with membership queries and equivalence queries (to be exact, the time complexity must hold at any point of the inference procedure). Angluin also proves that both of these queries are necessary: neither membership queries alone, nor equivalence queries alone allow for polynomial inference. Following Angluin's definitions, further classes of grammars have been proven to be polynomially learnable with a MAT (Ishizaka, 1989).

In both of these models the results are mainly negative, i.e., even DFA can't be inferred in polynomial time (unless membership and equivalence queries are used). Nevertheless the needs of applications in several fields (speech, pattern recognition, automatic translation. . .) have led to the construction of heuristics to solve the learning problems. The natural question that follows is whether these heuristics are necessary and what can be done in the (usual) case of learning from given data. When working in this framework an obvious parameter for the complexity of a learning algorithm is the size of the data, where the size corresponds to the sum of the lengths of the strings in the learning set. We note that in the two models above, the justification for not using only the size of the target automaton as a measure of computational complexity is that we do not have enough control over the presentation of examples (in the first case) or the oracle (in the second one) to avoid receiving an unnecessarily long example. This leads to asking if such long examples are really necessary for the identification process. Moreover, how judicious is it to talk of polynomial learning if the counter-examples returned by the oracle are of unreasonable

length with respect to the concept to be learned? By unreasonable we will use the usual polynomial barrier, and ask the question in another way: suppose the concept to be learned is of size n, does the learner require a counter-example of length larger than any fixed polynomial in n? This question has been raised and discussed in the case of DFA (with a negative answer) (Angluin, 1987), but not for other classes of grammars. In fact, we prove in Section 4 that the class of simple deterministic grammars, although polynomially learnable with a MAT, needs strings of super-polynomaial length as counter-examples.

A theoretical framework to study these issues is provided by Gold: he presented (1978) a model for identification from given data, where a sample of labelled strings $(S+, S-)$, with $S+$ a set of positive instances, and $S-$ a set of negative instances, is presented to the inference algorithm which must return a representation compatible with $(S+, S-)$. The further conditions are: for each language there exists a characteristic sample with which the algorithm returns a correct representation, and this must be monotone in the sense that if correctly labelled examples are added to the characteristic set, then the algorithm infers the same language. These conditions insure identification, and it is easy to see that a class of representations is identifiable from given data if and only if it is identifiable in the limit from a complete presentation of examples. This model can be compared with the one refused by Pitt (1989): it consists in adding to the polynomial update time condition a second condition: as soon as all elements in the characteristic set have been presented, the algorithm must infer correctly; when this condition is met we say that the class is *identifiable in the limit from polynomial time and data* (Gold, 1978). Gold proved that deterministic finite automata are identifiable in the limit from polynomially time and data. It must be noticed that in the same paper Gold proved the NP-completeness of the "Minimum Inferred Finite State Automaton" problem (is there a DFA with less than n states consistent with the data?). This result yields that it is intractable to find the smallest automaton consistent with the data. The results are not contradictory because a characteristic set is not just any set, and thus, in this special case, what is inferred (in polynomial time) is the smallest compatible automaton. Further work in this model has contributed the following results: alternative algorithms have been proposed to infer DFA (Oncina & García, 1992), even linear grammars have been proved identifiable in the limit from polynomial time and data (Takada, 1988; Sempere & García, 1994); these techniques have been extended to universal linear grammars (Takada, 1994). Following the same idea, deterministic even linear grammars are polynomially identifiable from positive examples only (Koshiba et al., 1995). The same holds for total subsequential functions (Oncina et al., 1993). Algorithms provided in these papers have been implemented to deal with practical problems in the fields of speech (García et al., 1994), pattern recognition (García & Vidal, 1990) and Automatic Translation (Castellanos et al., 1994). On the other hand no hardness results have been proven within this model, leaving open the question of the triviality of the model. This paper deals with proving that this is not so, and that many classes of grammars can not be identified in the limit from polynomial time and data.

The notion of characteristic sets leads in a natural way to the associated problem of teaching (Shinohara & Miyano, 1991; Goldman & Kearns, 1995; Jackson & Tomkins, 1992): a teacher's goal is to help the learner (or the identification algorithm), by providing a "good" set of examples. Following the general work on Freivalds et al., on good examples (1989), different models of teaching have in recent years been proposed (Anthony et al., 1992;

Wiehagen, 1992; Mathias, 1995). The point of view that best fits to grammatical inference (from given data) is the one of Goldman & Mathias (1996): they define teachability of a class as the existence of a characteristic (to a learner) teaching set. Their point of view follows the trend introduced by Jackson & Tomkins (1992), of considering teacher/learner couples. We will prove that learnable in Gold's sense corresponds to teachable in the sense of Goldman & Mathias.

Alternative models for teacher/learner couples have been presented; one main difference lies in the fact that in the above model the teacher will provide the examples for a specific learner, whereas in other models (Goldman & Kearns, 1995, Jackson & Tomkins; 1992) the teacher must be able to teach any consistent learner, i.e., any learner capable of returning a hypothesis compatible with the examples.

In the following section we give the main definitions and results we need from formal languages theory. Section 3 deals with adapting the teacher/learner model for grammatical inference. The difficulties of this task are shown and the technical results we need are given. In Section 4 give our main results, i.e., that the following classes of representations do not admit identification in the limit from polynomial time and data:

- Context-free grammars.
- Linear grammars.
- Simple deterministic grammars.
- Nondeterministic finite automata.

2. Definitions

We only give here the main definitions from formal language theory. For more details and proofs, the reader can refer to a textbook on the subject, for instance (Harrison, 1978).

An alphabet is a finite, non-empty set of distinct symbols. For a given alphabet Σ, the set of all finite strings of symbols from Σ is denoted Σ^*. The empty string is denoted λ. For a string w, $|w|$ denotes the length of w. A language L over Σ is a subset of Σ^*.

A *nondeterministic finite automaton* (NFA) over Σ is a 5-tuple $N = (Q, \Sigma, \delta, I, F)$ where Q is a finite set of states, I and F two subsets of Q, denoting respectively the set of initial states and the set of final states of N; δ is the set of transitions, namely a finite subset of $Q \times \Sigma \to 2^Q$. We denote by $q' \in \delta(q, x)$ a transition labelled by x from q to q'. A NFA N accepts a string w *iff* either $w = \lambda$ and $I \cap F \neq \emptyset$, or $w = x_1 \cdots x_j$ and there exists a sequence of states $q_0, \ldots, q_j$ (possibly with repetitions) with $q_0 \in I, \forall i < jg_{i+1} \in \delta(q_i, x_{i+1})$, and $q_j \in F$. When the set I contains a unique state and δ is functional, i.e., $\forall q \in Q\ \forall x \in \Sigma\ |\delta(q, x)| \leq 1$, the automaton is a deterministic finite automaton (DFA). The language recognized by an automaton is the set of all strings accepted by the automaton. Two automata are equivalent *iff* they accept the same language. NFA and DFA have equal power of expression (they both recognize the regular languages). Nevertheless, for a given NFA, the number of states of an equivalent DFA can be exponentially larger.

A context-free grammar over Σ is a 4-tuple $G = (\Sigma, V, P, S_0)$ where V is a finite alphabet (of non-terminal symbols or variables), S_0 a special symbol in V called the start symbol and P a finite subset of $V \times (\Sigma \cup V)^*$ called the set of productions (or rules). A

rule in this set will be denoted $S \to \alpha$ and has intended meaning: non-terminal symbol S rewrites into α. A derivation is a sequence $S_0 \to \beta_1 \to \beta_2 \cdots \to \beta_n$ where β_{i+1} is obtained by substituting some occurrence of a nonterminal T in β_i by α where $(T \to \alpha) \in P$. The language generated by a context-free grammar G (denoted $L(G)$) is the set of all strings in Σ^* that can be obtained by derivation from S_0. A language is context-free *iff* there exists some context-free grammar $G = (\Sigma, V, P, S_0)$ generating it. Notice that non-terminal symbols are only used for the generation of strings; they do not appear in the strings of the language. Two context-free grammars are equivalent *iff* they generate the same language.

A linear grammar $G = (\Sigma, V, P, S_0)$ is a context-free grammar where all productions belong to $V \times (\Sigma^* \cup \Sigma^* V \Sigma^*)$. Thus, each rule has at most one non-terminal in its right-hand part. The following classes shall be used in this paper:

DFA(Σ): the class of deterministic finite automata over alphabet Σ.
NFA(Σ): the class of nondeterministic finite automata over alphabet Σ.
CFG(Σ): the class of context-free grammars over alphabet Σ.
LIN(Σ): the class of linear grammars over alphabet Σ.

The size of the alphabet can be considered a constant when working on some representation class $\boldsymbol{R}(\Sigma)$.

When considering these classes the size of a representation (denoted *size* (R)) will be some reasonable quantity: it must be polynomial in the number of bits needed to encode a representation. The following sizes are typically correct (for a constant alphabet Σ):

For DFA and NFA the number of states.
For CFGs and Linear Grammars the number of rules multiplied by the length of the longest rule.

We end this section with results concerning a specific problem on representations that is used in the sequel, and plays an important role for identification from given data:

The Equivalence problem $EQ(\boldsymbol{R}, \Sigma)$: For a class $\boldsymbol{R}(\Sigma)$, are two given representations equivalent? (i.e., do they represent the same language?)

The following results are well known:

Theorem 1 (Garey & Johnson 1979, Harrison 1978).
$EQ(\boldsymbol{DFA}, \Sigma) \in P$.
$EQ(\boldsymbol{NFA}, \Sigma)$ *is co-NP-complete*, *even when* $|\Sigma| = 1$.
$EQ(\boldsymbol{CFG}, \Sigma)$ *is undecidable.*
$EQ(\boldsymbol{LIN}, \Sigma)$ *is undecidable.*

3. Teaching and characteristic sets

To take into account the fact that the length of the examples must depend polynomially on the size of the concept to be learned we propose the following definition which is

just a generalisation of Gold's results (1978) and a natural restriction of the definition of polynomial update time (Pitt, 1989).

Definition 1. A representation class $\boldsymbol{R}$ is identifiable in the limit from polynomial time and data *iff* there exist two polynomials $p()$ and $q()$ and an algorithm $\boldsymbol{A}$ such that:

1) Given any sample $(S+, S-)$, of size m, $\boldsymbol{A}$ returns a representation R in $\boldsymbol{R}$ compatible with $(S+, S-)$ in $\boldsymbol{O}(p(m))$ time.
2) For each representation R of size n, there exists a characteristic sample $(CS+, CS-)$ of size less than $q(n)$ for which, if $S+ \supseteq CS+$, $S- \supseteq CS-$, $\boldsymbol{A}$ returns a representation R' equivalent with R.

By this definition algorithm $\boldsymbol{A}$ is a polynomial learner. Notice that Gold proved (1978) that identification in the limit from given data is equivalent to identification from a complete sequence (the on line protocol). With this definition Gold's 1978 result can be restated as follows:

Gold's theorem (1978). *DFA are identifiable in the limit from polynomial time and data.*

In fact his result is even stronger since for any DFA a characteristic set can also be computed in polynomial time (Oncina & García, 1992).

Goldman & Mathias (1996) present a model for teaching and learning that takes into account the quantity of information a good teacher has to provide to a learner. A first problem is to avoid collusion. Collusion (or cheating) occurs when the teacher can pass information to the learner about the representation of the concept and not the concept itself[1]. The teaching session is described as follows, where to avoid collusion, a third element is introduced, namely an adversary who can complicate the learner's task by introducing extra examples:

1) The adversary selects a target function and gives it to the teacher.
2) The teacher computes a set of examples sufficient to allow the learner to infer the target concept.
3) The adversary adds correctly labelled examples to this set, with the goal of causing the learner to fail.
4) On this augmented set the learner computes a function.

Goldman & Mathias prove that this model does not allow collusion (as defined by them), and define a class of functions as polynomially T/L-teachable *iff* the learner always infers the intended function and both teacher and learner work in polynomial time. The class is semi-poly T/L-teachable if the condition that the teachers computation takes polynomial time is abandoned.

It is obvious that this definition is related to one of the identification in the limit from polynomial time and data, thus, from Gold's result it follows that DFA are semi-poly T/L-teachable. In fact, in this case the characteristic set can be computed in polynomial time (Oncina & García, 1992) so the stronger result that DFA are polynomially T/L-teachable also holds.

It should be stressed that we have chosen here to adapt Goldman & Mathias' model for the case of grammatical inference by taking into account the length of the examples as a parameter. In their original setting this was unnecessary. When concerned with the inference of boolean functions all examples have the same size, the number of variables. In the framework of grammatical inference the number of strings is infinite, their length growing unbounded; thus, when considering the size of the teaching set the number of strings alone is insufficient. A teacher that needs only a small number of examples, but some of them of excessive length, will not allow for the class to be (semi-) polynomially teachable.

Formally:

Definition 2. A representation class $\boldsymbol{R}$ is semi-poly T/L teachable *iff* there exist 3 polynomials $p()$, $q()$, $r()$, a teacher, T and a learner L such that for any adversary ADV the following teaching session succeeds:

1) ADV selects a target function f of size n in $\boldsymbol{R}$ and gives it to T.
2) T computes a set of examples for L, with at most $p(n)$ examples, all of length at most $q(n)$.
3) ADV adds correctly labelled examples to this set, with the goal of causing L to fail. Let m be the size of the completed set.
4) On this augmented set L computes the function f in time less than $r(m)$.

This definition is not quite as strong as Definition 1. Indeed the learner has no obligation when the teacher fails to give him a good teaching set. Equivalence between both definitions holds only for classes for which the problem of finding a consistent function is easy. This is the case for all usual classes of grammars, where constructing a grammar that generates exactly all positive instances is straightforward. We call *consistency-easy* a class for which there exists a polynomial algorithm that given a set of labelled strings, returns a representation consistent with this set.

Proposition 1. *A consistency-easy class is identifiable in the limit from time and data iff it is semi-poly T/L teachable.*

Proof Sketch: If a class is identifiable in the limit from polynomial time and data, then for any target function (or representation) f, a polynomial characteristic set exists. This set meets the conditions to be the set of examples proposed by the teacher, and the monotonicity condition insures that no adversary can cause the learning algorithm to fail. Conversely, as the class is consistency-easy, a consistent function can always be returned in polynomial time, and the set of examples for L, with at most $p(n)$ examples, all of length at most $q(n)$ is a polynomial characteristic set. □

Thus, Goldman & Mathias' model is well adapted for grammatical inference. A natural question is to consider other teaching models: an interesting model with implications for grammatical inference yet to be studied is the interactive model (Mathias, 1995). Another trend of research considers unspecialized teachers, that should be able to adapt to any learner. Obviously the class of learners to consider is limited to the consistent learners, those who always return a solution consistent with the data (as defined for instance in (Goldman &

Kearns, 1995)). We prove that a finite teaching set does not exist for consistent language learners, even when the learners are restricted to work in polynomial time (i.e., that given a wide class of learners, there is no polynomial teaching set from which all learners can identify correctly the target function). The proof is similar in many ways to the proof of Lemma 1 in (Goldman & Kearns, 1995); the difference is that the result here applies even when the learners work in polynomial time.

We first note that any learner fulfilling the conditions of Definition 1 is a consistent learner: it always returns a solution consistent with the data.

Definition 3. Let $\boldsymbol{R}$ be a class of representations and $\boldsymbol{I}$ be a set of identification algorithms for $\boldsymbol{R}$. $\boldsymbol{R}$ is *polynomially characterisable* for $\boldsymbol{I}$ *iff* any representation R in $\boldsymbol{R}$ admits a characteristic set polynomial in size(R), that allows any algorithm in $\boldsymbol{I}$ to identify it.

Proposition 2. *Let $\boldsymbol{R}$ be a class of representations containing representations for all singleton languages (containing just one string) and the empty language. Let $\boldsymbol{I}$ be the set of all polynomial identification algorithms for $\boldsymbol{R}$. If $\boldsymbol{I}$ is not empty, $\boldsymbol{R}$ is not polynomially characterisable for $\boldsymbol{I}$.*

Proof: If $\boldsymbol{I}$ is not empty it contains at least one algorithm identifying in the limit from polynomial time and data; call it $\boldsymbol{A}_{\mathbf{basic}}$. Let a be some symbol in the reference alphabet Σ. We define a family $\{A_k\}$ (for all integers k) of learning algorithms as follows:

Algorithm A_k

If $\{a^k\}$ is compatible with the sample then return $\{a^k\}$.
If not use algorithm $\boldsymbol{A}_{\text{basic}}$.

It is straightforward to notice that each of these algorithms complies with the conditions of Definition 1, so they are polynomial algorithms for identification, even if the complexity of each A_k grows with k.

Suppose now that the target language is the empty set. Then to identify it an arbitrary learner A_k requires a^k as a negative instance. The number of learners is infinite, hence so is the size of the characteristic set. □

This proposition applies, for instance, for the regular languages when represented by DFA, as for any string of length k we can construct a DFA with $k+1$ states that recognizes only that string.

But the existence of such a universal teacher is not a necessary condition for identification in the limit from polynomial time and data to be possible. We now aim to give such a necessary condition:

Definition 4. A representation class $\boldsymbol{R}$ is polynomially characterisable *iff* there exists a polynomial $p()$, such that for each representation R of size n, there exists a characteristic sample $(CS+, CS-)$ of size less than $p(n)$ such that if another non equivalent representation R' is compatible with $(CS+, CS-)$, then R is incompatible with the characteristic sample for R'.

Proposition 3. *If **R** is identifiable in the limit from polynomial time and data, then **R** is polynomially characterisable.*

Proof[2]**:** If not there are two non equivalent representations R_1 and R_2 with respective characteristic samples $(S+_1, S-_1)$, $(S+_2, S-_2)$. By compatibility $(S+_1 \cup S+_2, S-_1 \cup S-_2)$ would be accepted by R_1 and R_2. But any algorithm can only infer one of the representations. □

In its negative form Proposition 3 provides us with a tool to prove that certain classes are not identifiable in the limit from polynomial time and data.

Hellerstein et al. (1995) use a similar technique in the context of polynomial-query learnability. A class has the *polynomial resilience property* when a concept in this class cannot be separated from other concepts through a polynomial number of polynomial strings (thus forbidding the use of interesting queries). As in Proposition 3 above, their definition leads to complexity-theoretical results.

4. Nonpolynomially identifiable grammars

In this section we turn to general classes of grammars and prove that they are not polynomially characterisable (and hence, by Proposition 3, not identifiable in the limit from polynomial time and data).

Theorem 2. *For any alphabet Σ of size at least 2, the following classes are not polynomially characterisable:*
CFG(Σ)*, the class of context-free grammars*
LIN(Σ)*, the class of linear grammars.*

Proof: If equivalence is undecidable for a class ***R***, then for every $p()$ and every n (sufficiently large) we can find two representations R_1 and R_2 with size bounded by n, representing different languages, and inseparable by any string of length smaller than $p(n)$. If not, testing all strings up to that size would be a computable equivalence test of both representations. Thus ***R*** is not polynomially characterisable. This, by Theorem 1, applies to classes of grammars **CFG**(Σ) and **LIN**(Σ). □

Because of the undecidability of the equivalence problem (Theorem 1), this result can be extended to the above classes in any computable normal form (Chomsky normal form, Greibach normal form. . .)(Harrison, 1978):

Corollary 1. *For any alphabet Σ of size at least 2, the following classes are not polynomially characterisable:*
NCFG(Σ)*, the class of context-free grammars in some computable normal form.*
NLIN(Σ)*, the class of linear grammars in some computable normal form.*

Even when the equivalence problem is decidable, if the separating strings are too long, then inference cannot be obtained through characteristic samples of polynomial length.

The class of simple deterministic grammars has been proven polynomially inferable (with queries) by Ishizaka (1989). A context-free grammar is simple deterministic (in 2-normal form) if the rules are of the following form:

$$S \to \alpha \text{ with } \alpha \in \Sigma \cup \Sigma V \cup \Sigma V^2,$$
$$\text{and if } \forall x \in \Sigma\, [A \to x\alpha] \in P \text{ and } [A \to x\beta] \in P \Rightarrow \alpha = \beta.$$

As simple deterministic grammars generalize regular grammars, the consistency problem can be solved in polynomial time.

Theorem 3. *For any alphabet Σ, the class of simple deterministic grammars over Σ is not polynomially characterisable.*

Proof: Take the following (indexed) simple deterministic grammar:

$$G_k = < \{a\}, \{S_i : i \in \{0..k\}\}, P, S_0 >$$
$$\text{and } P = \{ \quad S_i \to aS_{i+1}S_{i+1} \forall\, i < k$$
$$S_n \to a \quad \}$$

The size of each grammar G_k is obviously polynomial in k. Yet, each grammar G_k generates only one string (of exponential length): $a^{2^{k+1}-1}$.

Thus, $L(G_k)$ cannot be separated from the empty language by any subset of strings of polynomial length. And as above, the result follows. □

This result does not constitute a contradiction with Goldman & Mathias' theorem that "any class learnable in deterministic polynomial time using example-based queries is semi-poly T/L teachable" (Goldman & Mathias, Theorem 2, 1996). Indeed we have the following apparently contradictory facts:

- Ishizaka has proven that simple deterministic grammars could be inferred in polynomial time with equivalence and membership queries (Ishizaka, 1989).
- From Goldman & Mathias' result it follows that simple deterministic grammars are semi-poly *T/L* teachable.
- From Proposition 1 it follows that for simple deterministic grammars, "semi-poly T/L teachable" is equivalent to "identifiable in the limit from polynomially time and data", as simple deterministic grammars are consistency-easy.
- Theorem 3 states that simple deterministic grammars are not polynomially characterisable, hence not identifiable in the limit from polynomial time and data.

The contradiction depends on the role of the length of examples and counter-examples:

- For Ishizaka, the oracle is independent, so the length of counter-examples is a parameter.
- Goldman & Mathias only consider boolean functions. The length is a constant.

We believe this length must depend polynomially of the size of the target grammar.

These considerations explain that Goldman & Mathias' theorem is no longer true for the definition of teachability we have adapted to the case of grammatical inference. A second corollary of Theorem 3 is that to infer a simple deterministic grammar with a MAT the length of the counter-examples to be expected can be bounded *a priori* by no polynomial.

Learning NFA is difficult, even with a MAT (see e.g. (Yokomori, 1993), or (Angluin & Kharitonov, 1995)), this remains true for our criterion:

Theorem 4. *If $P \neq NP$ for any non empty alphabet Σ, the class $\boldsymbol{NFA}(\Sigma)$ is not polynomially characterisable*:

Proof: In the case where the input alphabet has only one letter, the equivalence problem is co-NP-complete (Garey & Johnson, 1979). Thus there is no polynomial $p()$ that given two NFA of size smaller than n can solve the equivalence problem by testing chains of length less than $p(n)$: otherwise the number of such chains is precisely $p(n)$, and the equivalence problem would be in P. Hence the result. In the appendix we give a construction of this fact. □

Corollary 2. *The following classes are not identifiable in the limit from polynomial time and data:*
$\boldsymbol{CFG}(\Sigma)$, *the class of context-free grammars over* Σ, *when* $|\Sigma| > 1$
$\boldsymbol{LING}(\Sigma)$, *the class of linear grammars over* Σ, *when* $|\Sigma| > 1$
$\boldsymbol{SDG}(\Sigma)$, *the class of simple deterministic grammars over* Σ, *when* $|\Sigma| > 0$
$\boldsymbol{NFA}(\Sigma)$, *the class of nondeterministic finite automata over* Σ, *when* $|\Sigma| > 0$

The results follow by applying Proposition 2 to the above results. The fourth result depends on the assumption $P \neq NP$.

5. Conclusion

The framework of identification in the limit from polynomial time and data has so far provided the grammatical inference community with many positive results. This framework is implicitly defined in Gold's 1978 article, linked with Pitt's propositions (1989) and corresponds to Goldman & Mathias' teaching model (1996) when the length of the examples is taken as a variable. We have proven that a number of important classes are not identifiable in the limit from polynomial time and data. Nevertheless as this setting does not guarantee polynomial induction, work remains to be done: in the positive cases (*DFA*. . .) how simple can the characteristic set be? As different algorithms will admit different characteristic sets, does this give us a quality measure of an inference algorithm (the smaller the characteristic set the better). These issues are important ones: grammatical inference algorithms are used today in different fields, and theoretical results comparing existing algorithms and justifying new ones can be of considerable help.

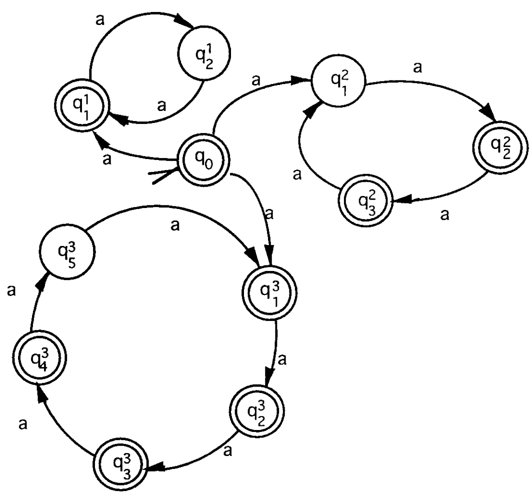

Figure 1.

Appendix

The proof of Theorem 4 is non-constructive and actually finding non deterministic automata for which the equivalence problem requires looking at non polynomial chains is not straightforward. We give the following procedure:

Construct two automata:
Automaton A recognises a^*.

Automaton B is formed by one starting state q_0, and for every prime number p_i in a finite set PRIME (included in the set of all primes) states $q_1^i, \ldots, q_{p_i}^i$. The transitions are:

- $\forall p_i \in PR\, q_1^i, \in \delta(q_0, a)$,
- $\forall p_i \in PR\, \forall_j \in \{1, \ldots, p_i - 1\} q_{j+1}^i \in \delta\big(q_j^i, a\big)$
- $\forall p_i \in PR\, q_1^i \in \delta\big(q_{p_i}^i, a\big)$

All states are final except each q_{pi}^i . The automaton B for the set PRIME $= \{2, 3, 5\}$ is drawn in Fig. 1. Initial state is q_0, all states are final except q_2^1, q_3^2 and q_5^3.

The language recognised by B for a given set PRIME is $a^* - (a^l)^+$ with $l = \prod_{p_i \in \text{PRIME}} p_i$.

The smallest string separating this language from a^* is a^l.

Thus, the number of states is $\sum_{P_i \in \text{PRIME}} p_i$.

And the smallest separating word is of length $\prod_{p_i \in \text{PRIME}} p_i$.

Now for each integer m there exists (Heath-Brown & Iwaniec, 1979) a prime in the interval $[m, m + m^{11/20}]$. We can thus deduce that $\forall j \geq 1, \exists p \in \text{PRIME} \cap [2^j, 2^{j+1}[$, and by choosing for each of these intervals one prime we have for each j a set PRIME(j) containing j primes each included in a different interval $[2^i, 2^{i+1}[$ $(\forall i < j)$.

Automaton B has set of states Q. If we denote by $\min l(j)$ the length of the smallest string in the symmetric difference of B and A, we have

$$|Q| \leq \sum_{i=1}^{j} 2^{i+1} \leq 2^{j+2}. \qquad \min l(j) \geq \prod_{i=1}^{j} 2^{i} \geq 2^{j^2/2}.$$

It follows that for any k we can find j such that $|Q|^k < \min l(j)$.

Acknowledgment

The author wishes to thank researchers from the Universidad Politécnica de Valencia for many fruitful discussions. He is more particularly indebted to Jose-Maria Sempere for ideas in Proposition 2, and other important suggestions, and to Pierre Dupont for a careful reading of the manuscript.

Notes

1. For more about collusion, see (Mathias, 1995).
2. This proof uses a similar technique to the proof of Theorem 1 in (Goldman & Mathias, 1996), where unions of teaching sets are considered to prove that their method is not collusion.

References

Angluin, D. (1987). Queries and concept learning. *Machine Learning*, 2:319–342.

Angluin, D., & Kharitonov, M. (1995). When won't membership queries help? *Journal of Computer and System Sciences*, 50(1):336–355.

Anthony, M., Brightwell, G., Cohen, D., & Shawe-Taylor, J. (1992). On exact specification by examples. *Proceedings of COLT*'92 (pp. 311–318). A.C.M.

Castellanos, A., Galiano I., & Vidal, E. (1994). Application of OSTIA to machine translation tasks. *Proceedings of the International Colloquium on Grammatical Inference ICGI-94* (pp. 93–105). Lecture Notes in Artificial Intelligence (Vol. 862). Springer-Verlag.

Freivalds, R., Kinber, E.B., & Wiehagen, R. (1989). Inductive inference from good examples. *Proceedings of the International Workshop on Analogical and Inductive Inference* (pp. 1–17). Lecture Notes in Artificial Intelligence (Vol. 397). Springer-Verlag.

García, P., Segarra, E., Vidal, E., & Galiano, I. (1994). On the use of the morphic generator grammatical inference (MGGI) methodology in automatic speech recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 4:667–685.

García, P., & Vidal, E. (1990). Inference of K-testable languages in the strict sense and applications to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):920–925.

Garey, M.R., & Johnson, D.S. (1979). *Computers and Intractability: A guide to the Theory of NP-Completeness*. San Francisco: W.H. Freeman.

Gold, E.M. (1967). Language identification in the limit. *Inform. & Control*, 10:447–474.

Gold, E.M. (1978). Complexity of automaton identification from given data. *Information and Control*, 37:302–320.

Goldman, S.A., & Kearns, M.J. (1995). On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31.

Goldman, S.A., & Mathias, H.D. (1996). Teaching a smarter learner. *Journal of Computer and System Sciences*, 50(2):255–267.

Harrison, M.A. (1978). *Introduction to Formal Language Theory*. Reading: Addison-Wesley.

Heath-Brown, D., & Iwaniec, H. (1979). *Invent. Math.*, 55:49–69.

Ishizaka, H. (1989). Learning simple deterministic languages. *Proceedings of COLT'89* (pp. 162–174). A.C.M.

Jackson, J., & Tomkins, A. (1992). A computational model of teaching. *Proceedings of COLT'92* (pp. 319–326). A.C.M.

Koshiba, T., Makinen, E., & Takada, Y. (1995). Learning deterministic even linear languages from positive examples. *Proceedings of ALT'95*, Lecture Notes in Artificial Intelligence (Vol. 997). Springer-Verlag.

Mathias, H.D. (1995). If you can't learn 'em teach 'em. In *Proceedings of COLT'95*.

Oncina, J., & García, P. (1992). Inferring regular languages in polynomial time. *In Pattern Recognition and Image Analysis, World Scientific.*

Oncina, J., García, P., & Vidal E. (1993). Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15:448–458.

Pitt, L. (1989). Inductive inference, dfas and computational complexity. *Proceedings of the International Workshop on Analogical and Inductive Inference* (pp. 18–44). Lecture Notes in Artificial Intelligence (Vol. 397). Springer-Verlag.

Sempere, J.M., & García, P. (1994). A characterisation of even linear languages and its application to the learning problem. *Proceedings of the International Colloquium on Grammatical Inference ICGI-94* (pp. 38–44). Lecture Notes in Artificial Intelligence (Vol. 862). Springer-Verlag.

Shinohara, A., & Miyano, S. (1991). Teachability in computational learning. *New Generation Computing* 8:337–347.

Takada, Y. (1988). Grammatical inference for even linear languages based on control sets. *Information Processing Letters*, 28:193–199.

Takada, Y. (1994). A hierarchy of language families learnable by regular language learners. *Proceedings of the International Colloquium on Grammatical Inference ICGI-94* (pp. 16–24). Lecture Notes in Artificial Intelligence (Vol. 862). Springer-Verlag.

Wiehagen, R. (1992). From inductive inference to algorithmic learning theory. *Proceedings of ALT'92* (pp. 13–24). Lecture Notes in Artificial Intelligence (Vol. 743). Springer-Verlag.

Yokomori, T. (1993). Learning nondeterministic finite automata from queries and counterexamples. *Machine Intelligence*, vol. 13. Furukawa, Michie & Muggleton (Eds.), Oxford Univ. Press.

Received December 19, 1995
Accepted August 21, 1996
Final Manuscript September 16, 1996