ARTICLE IN PRESS
YINCO:4281

Information and Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Automatic learning from positive data and negative counterexamples

Sanjay Jain [a,*,1], Efim Kinber [b,*], Frank Stephan [c,*,2]

[a] School of Computing, National University of Singapore, Singapore 117417, Republic of Singapore
[b] Department of Computer Science, Sacred Heart University, Fairfield, CT 06432-1000, USA
[c] Department of Mathematics and Department of Computer Science, National University of Singapore, Singapore 119076, Republic of Singapore

ARTICLE INFO

Article history:
Received 3 March 2016
Received in revised form 14 December 2016
Available online xxxx

Keywords:
Inductive inference
Automatic learning
Automatic classes
Negative counterexamples
Iterative learning

ABSTRACT

We introduce and study a model for learning in the limit by finite automata from positive data and negative counterexamples. The focus is on learning classes of languages with the membership problem computable by finite automata (so-called automatic classes). We show that, within the framework of our model, finite automata (automatic learners) can learn all automatic classes when memory of a learner is restricted by the size of the longest datum seen so far. We also study capabilities of automatic learners in our model with other restrictions on the memory and how the choice of negative counterexamples (arbitrary, or least, or the ones which are bounded by the largest positive datum seen so far) can impact automatic learnability.

1. Introduction

Inductive inference [1,16,23] studies learning as a limiting process of step by step improved conjectures depending on more and more incoming data. The field, due to its precise and multiple notions of convergence of hypotheses and data presentations, has applications and connections both inside and outside the field of learning:

- related areas within the field of learning are query learning [3,9] and grammatical inference [18–20];
- the learning of pattern languages [2,38] and elementary formal systems [45], originally studied in inductive inference, have been applied to other areas like molecular biology [4–7];
- areas which also apply notions from inductive inference outside the field of learning are the area of Weihrauch degrees [12,13], the study of limiting processes in recursion theory [17,25,26,42] and program synthesis [24,35].

1.1. Automatic inductive inference

Many inductive inference criteria consider learnability in the general framework of recursion theory and do not impose resource bounds. However, some works do consider limitations of the update time in response to a newly read datum

* Corresponding authors.
E-mail addresses: sanjay@comp.nus.edu.sg (S. Jain), kinbere@sacredheart.edu (E. Kinber), fstephan@comp.nus.edu.sg (F. Stephan).

[1] Supported in part by MOE Academic Research Fund Tier 1 through NUS grants R252-000-420-112 and R146-000-181-112 and NUS Provost Chair grant C252-000-087-001.
[2] Supported in part by MOE Academic Research Fund Tier 1 through NUS grants R252-000-420-112 and R146-000-181-112.

http://dx.doi.org/10.1016/j.ic.2017.05.002
0890-5401/© 2017 Elsevier Inc. All rights reserved.

[43] or the amount of memory permitted to store information about previously read data [21,36]. Jain, Luo and Stephan [32] introduced an "automatic" variant of inductive inference where these restrictions are formulated in the language of automata theory: the family of target languages is computable by a finite automaton (automatic family) and a learner, more precisely its update function, is a finite automaton itself (automatic learning). This approach is motivated from the fact that

> automatic functions coincide with those computed by a position-faithful one-tape Turing machine in linear time [14].

Therefore automatic learners are learners whose update function (of long-term memory and hypothesis) satisfies a very natural and restrictive resource bound. This stands in contrast to results of Case and Kötzing [15] and Pitt [43] — they showed that for many natural constraints one can slow down learners to polynomial or even linear time without losing learnability according to the given constraints.

Automatic families are the natural counterpart to automatic learners, as they form a natural type of hypothesis space, which is compatible with the notions of automatic learners. Here an automatic family is a family of target languages, which is defined by a regular index set and where the membership problem in these languages is regular in the sense that one finite automaton recognises a combination (so called "convolution") of an index and a word if and only if the word is in the language defined by the index.

Gulwani, Jha, Tiwari and Venkatesan [24] and Jha and Seshia [35] considered a scenario of automated program synthesis (learning), which, in some cases, can be easily modelled with automatic families: One wants to build an acceptor of a language or a program with a certain input-output behaviour (specified by some first-order formula), where the program can be synthesised from basic constraints using bounded number of operations, with parameters, which can be executed in an arbitrary order, but without loops. Here the number of operations is bounded by some prefixed constant c. In the case that all the base operations are from an automatic structure — for example, integers or dyadic rationals with addition and comparison and multiplication by a fixed list of constants, if-then-else statements and operations to manipulate a constant amount of variables — then the resulting programs or acceptors of languages can all be listed within an automatic family, where the choice of the family depends on the constant c and the permitted operations. This could be handled well from both ends: A teacher might use automatic functions to extract counterexamples or draw examples or check correctness of a proposed solution; a learner can access the automatic family in order to find an index (= solution) satisfying the constraints.

Though Gulwani, Jha, Tiwari and Venkatesan [24] themselves worked within a bit larger scope than automatic structures, their concepts can be implemented quite efficiently when restricted to such a scope. Note that besides finding the right order of the operations, the learner has also to find the right parameters and this makes the process more involved than just dealing with finitely many case distinctions. Furthermore, the oracles mentioned by Jha and Seshia [35] can also be handled efficiently in the case of automatic structures, as the first-order theory used to write down the constraints is decidable.

1.2. Memory limitations

Jain, Luo and Stephan [32], besides introducing the usage of automatic learners, also considered three different natural types of limits on the size of the (long-term) memory available to an automatic learner before outputting the next conjecture:

(a) memory is bounded by the size of the longest positive input datum seen so far (plus a constant);
(b) memory is bounded by the size of the current hypothesis (plus a constant);
(c) the learner can store in the memory the last hypothesis only (*iterative learning*, see [44]).

Jain, Luo and Stephan [32] established that automatic learners are much weaker than unrestricted recursive learners — even when learning automatic classes. In particular, the class of co-singletons, $\{\{0,1\}^* - \{\alpha\} : \alpha \in \{0,1\}^*\}$, is not automatically learnable. Moreover, they showed the following modification of Angluin's result from [1]: An automatic class is learnable by a recursive learner iff it satisfies Angluin's tell-tale condition.

Let $L_\epsilon = \{0^j : j > 0\}$ and $L_\alpha = \{0^j : \alpha(j) = 1\} \cup \{\epsilon\}$, where $\alpha \in \{0,1\}^+$. Jain, Luo and Stephan [32] showed that $\{L_\alpha : \alpha \in \{0,1\}^*\}$, separates automatic learning by learners having memory bounded by longest positive input datum seen so far from automatic learning by learners having memory bounded by the size of the current hypothesis. To see this, on the positive side, a learner can keep track of the words 0^j seen in the input by memorizing a word $\beta \in \{0,1\}^*$ such that $\beta(j) = 1$ iff 0^j has been seen in the input text. This information is sufficient to determine the input language in the limit (as $\beta(0) = 1$ implies that the input language is L_β for the limiting value of β; otherwise the input language is L_ϵ). On the other hand, a learner with memory bounded by hypothesis size, while getting input L_ϵ, would forget some of the input words, and thus would not be able to know the target language when ϵ appears in the input later (see [32] for details).

It is open at present if automatic learning by learners having memory bounded by the size of the current hypothesis is different from automatic learning by learners having only the last hypothesis as its memory. Similarly, it is open at present if bounding the memory by the size of the longest word seen so far is a restriction for automatic learners having no memory restriction (except that implicit due to the learner being automatic). Jain, Luo and Stephan [32] also considered how the