On the Parameterised Complexity of Learning Patterns

Frank Stephan*,[1], Ryo Yoshinaka[2] and Thomas Zeugmann[3]

[1] Department of Mathematics and Department of Computer Science
National University of Singapore, Singapore 117417, Republic of Singapore
[2] ERATO Minato Project, Japan Science and Technology Agency
Hokkaido University, N-14, W-9, Sapporo 060-0814, Japan
[3] Division of Computer Science
Hokkaido University, N-14, W-9, Sapporo 060-0814, Japan

Abstract. Angluin (1980) showed that there is a consistent and conservative learner for the class of non-erasing pattern languages; however, most of these learners are *NP*-hard. In the current work, the complexity of consistent polynomial time learners for the class of non-erasing pattern languages is revisited, with the goal to close one gap left by Angluin, namely the question on what happens if the learner is not required to output each time a consistent pattern of maximum possible length. It is shown that consistent learners are non-uniformly $W[1]$-hard inside the fixed-parameter hierarchy of Downey and Fellows (1999), and that there is also a $W[1]$-complete such learner. Only when one requires that the learner is in addition both, conservative and class-preserving, then one can show that the learning task is *NP*-hard for certain alphabet-sizes.

1 Introduction

Angluin [1] introduced pattern languages as an example for an interesting class that is learnable in the limit from text. A *pattern* π is a finite string over a finite alphabet Σ and a countably infinite set X of variables, where $\Sigma \cap X \neq \emptyset$. The *language* $L(\pi)$ generated by π is the set of strings which can be obtained by substituting each variable in the pattern by a non-empty string over Σ.

Allowing the empty string as a possible substitution was also studied, yielding the *erasing pattern languages* (cf., e.g., [10, 12]) which are *not learnable from text* [10]. So we follow Angluin [1] and allow only non-empty substitutions. Such pattern languages are called *non-erasing pattern languages*.

We consider here the model of learning in the limit from text (see Section 3). Our learners are required to be *consistent* and/or *conservative* (cf., e.g., [9, 15]).

In [1, 2] the class of all non-erasing pattern languages was shown to be learnable in the limit from text and an easy modification of her learning algorithm is consistent and conservative (cf., e.g., [15]) but not polynomial-time computable if $P \neq NP$ (cf. [1], Theorem 3.6). Lange and Wiehagen [8] sacrificed the consistency requirement and provided a polynomial time learner. Zeugmann [14]

* Supported in part by NUS grant numbers R146-000-114-112 and R252-000-420-112.

studied the learning properties of their algorithm from a statistical perspective. Various approaches to learn pattern languages followed [6, 11, 13].

Note that there is no pattern learner which outputs only consistent patterns of maximum possible length unless $P = NP$ (cf. [1]). So what happens if we drop the maximum possible length condition? It is shown that the following holds:

– The existence of a polynomial-time consistent learner for the class of non-erasing pattern languages is equivalent to the condition that *CLIQUE* is non-uniformly fixed-parameter tractable, that is, $W[1] \subseteq FPT^n$;
– If there is a polynomial-time consistent and conservative learner using a hypothesis space where $H_i = L(\pi)$ for an index i and a pattern π is decidable, then *CLIQUE* is uniformly fixed-parameter tractable, i.e., $W[1] \subseteq FPT$.

There is an oracle A relative to which $FPT = W[Poly]$ and $P \neq NP$. This result gives some evidence that one cannot show that the existence of a polynomial time consistent and class-preserving pattern language learner would imply $P = NP$.

In [1] the word problem of the pattern languages was shown to be in $W[poly]$ which is the highest class of the W-hierarchy [5] but the exact level of that hierarchy was not located. Polynomial-time consistent and conservative class-comprising learning algorithms for classes where the membership-problem is uniformly polynomial-time computable and some mild other conditions hold were provided in [3, 4]. Below we use the algorithm of [8] instead of the work of [3, 4], but the algorithm of Theorem 4 can also be obtained by their methods.

2 The Complexity of the Pattern-Membership Problem

The *uniform* membership-problem $\{(\pi, w) \mid w \in L(\pi)\}$ is NP-complete (cf. [1]) but the membership-problem for a fixed pattern was not checked. So we ask what happens if the underlying numbering of all the pattern languages is *not* the default-numbering but given in a different way. Then certain information could be coded into the hypothesis space and the learner could gain power.

So it is adequate to look at the parameterised complexity (cf. [5]). A parameterised set A is in FPT iff there is a recursive function f and a polynomial p such that the question whether $(k, x) \in A$ can be decided in $p(f(k) + |x|)$ time.

A parameterised set S is *fixed-parameter many-one reducible* to a parameterised set T (abbr. $S \leq_m^n T$) iff there is a polynomial p such that for each parameter k there is a parameter k', a factor d_k and a reduction ψ_k such that ψ_k translates every $\langle x, k\rangle$ in time $d_k \cdot p(k+|x|)$ into $\langle x', k'\rangle$ with $\langle x, k\rangle \in S \Leftrightarrow \langle x', k'\rangle \in T$. Moreover, S is *strongly uniformly fixed-parameter many-one reducible* to T (abbr. $S \leq_m^s T$) iff there is a recursive function computing k', the factor d_k and a program for ψ_k from k (cf. [5]). We write $S \equiv_m^s T$ iff $S \leq_m^s T$ and $T \leq_m^s S$.

Let *PMO* denote the membership problem of pattern languages with the parameter k being the number of occurrences of variables in the pattern π, e.g., let π be such that one variable occurs twice and two other variables occur just one time each in π, then $k = 4$. For *CLIQUE* the parameter is the size of the clique requested. The complexity class $W[1]$ can be characterized as those sets which are strongly uniformly fixed-parameter many-one reducible to *CLIQUE*.

Theorem 1. *The problem PMO is $W[1]$-complete for strongly uniform fixed-parameter many-one reducibility. In particular, PMO $\equiv^s_m$ CLIQUE.*

3 Learning Theory

First, we recall notions from learning theory and then apply Theorem 1 to it. The source of information are texts. A *text* t is an infinite sequence eventually containing all words of the target language L and possibly some pause symbols but no non-members. The learner is fed incrementally growing initial segments t_n of t and computes a hypothesis from its input, say i_n, if it has seen precisely the first n words of t. The hypotheses are interpreted with respect to a chosen *hypothesis space* $\{H_i \mid i \in I\}$. The sequence $(i_n)_{n\in\mathbb{N}}$ ($\mathbb{N}$ is the set of all natural numbers) of all created hypotheses has to *converge* to an $i \in I$ such that $H_i = L$.

A learner *learns* L from text if it learns L from all text for it. A learner learns a class C with respect to a hypothesis space $\{H_i \mid i \in I\}$ from text iff it learns every language from C from text. This model is called *learning in the limit* from text (cf. [7]). Since we consider this model only, we refer to it just as *learning*.

Our learners are also required to be consistent and conservative (cf. [1, 2]). A learner M is *consistent* iff for every input t_n it outputs an i_n such that $\mathit{range}(t_n) \subseteq H_{i_n}$; if no such i_n exists, M outputs a special no-conjecture symbol. We call M *conservative* iff for every two subsequent hypotheses i_n based on t_n and i_{n+k} based on t_{n+k}, $i_n \neq i_{n+k}$, there is an $x \in \mathit{range}(t_{n+k})$ with $x \notin H_{i_n}$.

A hypothesis space $\{H_i \mid i \in I\}$ is *class-preserving* (with respect to the target class C) iff $\{H_i \mid i \in I\} = C$. We call $\{H_i \mid i \in I\}$ *class-comprising* iff $\{H_i \mid i \in I\} \supseteq C$. Every hypothesis space must be class-comprising; ideally it should be class-preserving, but this restricts learnability sometimes (cf. [15]).

We relate our results to Angluin's [1, 2] which are based on descriptive patterns. By *sample* we mean $\mathit{range}(t_n)$. A pattern π is *descriptive* of a sample S iff $S \subseteq L(\pi)$ and for every pattern τ with $S \subseteq L(\tau)$, we must have $L(\tau) \not\subset L(\pi)$. Every learner producing on every input a descriptive pattern learns the class of all non-erasing pattern languages (abbr. *PAT*) and there is an algorithm computing on input any sample S a pattern that is descriptive of S (cf. [1, 2]). If we require learnability with respect to a class-preserving hypothesis space this is the only way to obtain a consistent and conservative learner for *PAT*.

Theorem 2. *Let M be a consistent and conservative learner for PAT with respect to a class-preserving hypothesis space. Then M must output in every step a hypothesis that is descriptive for the content of the text seen so far.*

Theorem 3. *PAT has a consistent polynomial-time learner iff $W[1] \subseteq FPT^n$.*

The next result shows that for consistent and conservative learning with a minimum decidability requirement on the hypothesis space, polynomial-time learnability becomes linked to the more restrictive condition $W[1] \subseteq FPT$.

Theorem 4. *$W[1] \subseteq FPT$ iff there is a consistent and conservative polynomial-time learner for PAT which uses a hypothesis space $\{H_i \mid i \in I\}$ such that $\{(\pi, i) \mid L(\pi) = H_i\}$ is decidable.*

Next, we use an infinite alphabet Σ, e.g., Σ could be $\mathbb{N}$. Then it is *NP*-hard to make a consistent and conservative learner using a class-preserving hypothesis space. So the weakening from class-preserving to class-comprising hypotheses spaces is an important ingredient for the general polynomial-time learners obtained by [3, 4] for many uniformly polynomial-time decidable classes.

Theorem 5. *Let* $\Sigma = \{0, 1, 2, \ldots\}$. *Now* $P = NP$ *iff there is a consistent and conservative class-preserving polynomial-time learner for* PAT *over* Σ.

The same result holds if one works with finite alphabets and the learner has to learn the pattern plus the alphabet from the data. Then the learnability problem uniform over finite alphabets is *NP*-complete.

References

[1] Angluin, D.: Finding patterns common to a set of strings. J. of Comput. Syst. Sci. 21, 46–62 (1980)
[2] Angluin, D.: Inductive inference of formal languages from positive data. Inform. Control 45, 117–135 (1980)
[3] Case, J., Kötzing, T.: Dynamically delayed postdictive completeness and consistency in learning. In: Freund, Y., Györfi, L., Turán, G., Zeugmann, T. (eds.) ALT 2008. LNAI, vol. 5254, pp. 389–403. Springer, Berlin (2008)
[4] Case, J., Kötzing, T.: Difficulties in forcing fairness of polynomial time inductive inference. In: Gavaldà, R., Lugosi, G., Zeugmann, T., Zilles, S. (eds.) ALT 2009. LNAI, vol. 5809, pp. 263–277. Springer, Berlin (2009)
[5] Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science, Springer, Berlin (1999)
[6] Erlebach, T., Rossmanith, P., Stadtherr, H., Steger, A., Zeugmann, T.: Learning one-variable pattern languages very efficiently on average, in parallel, and by asking queries. Theoret. Comput. Sci. 261, 119–156 (2001)
[7] Gold, M.: Language identification in the limit. Inform. Control 10, 447–474 (1967)
[8] Lange, S., Wiehagen, R.: Polynomial-time inference of arbitrary pattern languages. New Generation Computing 8, 361–370 (1991)
[9] Lange, S., Zeugmann, T., Zilles, S.: Learning indexed families of recursive languages from positive data: A survey. Theoret. Comput. Sci. 397, 194–232 (2008)
[10] Reidenbach, D.: A non-learnable class of E-pattern languages. Theoret. Comput. Sci. 350, 91–102 (2006)
[11] Rossmanith, P., Zeugmann, T.: Stochastic finite learning of the pattern languages. Machine Learning 44, 67–91 (2001)
[12] Shinohara, T.: Polynomial time inference of extended regular pattern languages. In: RIMS Symposium on Software Science and Engineering, Proc. LNCS, vol. 147, pp. 115–127. Springer-Verlag, Berlin (1983)
[13] Shinohara, T., Arikawa, S.: Pattern inference. In: Algorithmic Learning for Knowledge-Based Systems, LNAI, vol. 961, pp. 259–291. Springer, Berlin (1995)
[14] Zeugmann, T.: Lange and Wiehagen's pattern language learning algorithm: An average-case analysis with respect to its total learning time. Annals of Mathematics and Artificial Intelligence 23, 117–145 (1998)
[15] Zeugmann, T., Lange, S.: A guided tour across the boundaries of learning recursive languages. In: Algorithmic Learning for Knowledge-Based Systems, LNAI, vol. 961, pp. 190–258. Springer, Berlin (1995)