Inductive Inference and Language Learning

Thomas Zeugmann

Division of Computer Science,
Hokkaido University, Sapporo 060-0814, Japan
thomas@ist.hokudai.ac.jp

Abstract. The present paper is a short reflection concerning the role which inductive inference played and can play in language learning. We shortly recall some major insights obtained and outline some new directions based on own work and results recently presented in the literature.

1 Introduction

Humans are excellent learners. In particular, every normal child acquires its mother tongue, a grammatical system which is very complex as research in linguistics shows.

On the other hand, if we look fifty years back, science fiction had anticipated that computers will be able to communicate with humans like humans, i.e., by using any native language. So far, this goal has not been achieved. Thus, it is only natural to take a closer look at fundamental research in learning theory and to analyze the state of the art with respect to the ambitious goal of language learning. Within this extended abstract, we shall confine ourselves to inductive inference as the underlying framework for language learning.

Formal language learning may be characterized as the study of systems that map evidence on a language into hypotheses about it. Of special interest is the investigation of scenarios in which the sequence of hypotheses *stabilizes* to an *accurate* and *finite* description (a grammar) of the target language. Clearly, then some form of learning must have taken place. In his pioneering paper, Gold [7] gave precise definitions of the concepts "evidence," "stabilization," and "accuracy" resulting in the model of learning in the limit. During the last decades, Gold-style formal language learning has attracted a lot of attention by computer scientists (cf., e.g., Osherson, Stob and Weinstein [14], Jain *et al.* [10] as well as Zeugmann and Lange [22], and the references therein). Most of the work done in the field has been aimed at the following goals: showing what general collections of language classes are learnable, characterizing those collections of language classes that can be learned, studying the impact of several postulates on the behavior of learners to their learning power, and dealing with the influence of various parameters to the efficiency of learning.

Next, we specify the information from which the target languages have to be learned. A *text* of a language L is an infinite sequence of strings that eventually contains all strings of L. Texts may be considered as a first model of the information available to children when learning their native language.

An algorithmic learner, henceforth called *inductive inference machine* (abbr. IIM), takes as input initial segments of a text. Using this information, it computes and outputs hypotheses about the target language. The set $\mathcal{H}$ of all admissible hypotheses is called *hypothesis space.* Furthermore, the sequence of hypotheses has to converge to a hypothesis correctly describing the language to be learned, i.e., after some point, the IIM stabilizes to an accurate hypothesis. If there is an IIM that learns a language L from all texts for it, then L is said to be *learnable in the limit from text* with respect to the hypothesis space $\mathcal{H}$.

Finally, we call a class $\mathcal{L}$ of languages *learnable in the limit from text* if there are an IIM M and a hypothesis space $\mathcal{H}$ such that M learns every language $L \in \mathcal{L}$ in limit from text with respect to $\mathcal{H}$.

Since all natural languages have grammars, we may think of hypothesis spaces as of sets of formal grammars (cf. Hopcroft and Ullman [9]).

Having reached this point of precision, one may ask which language classes are learnable from text. The first result we would like to mention here, is due to Gold [7], who proved the following.

Theorem 1. *Let $\mathcal{L}$ be any class of languages containing all finite languages and at least one infinite language. Then $\mathcal{L}$ is not learnable in the limit from text.*

Consequently, neither the class of regular languages nor any superset thereof can be learned in the limit from text. Taking this into account, many researchers thought that there is no interesting class of languages at all that can be learned in the limit from text. As a result, the study of learning from text faced almost one decade of decline after Gold's [7] pioneering paper. The situation considerably changed when Angluin [2] proved the pattern languages to be learnable in the limit from text. Moreover, Angluin [3] provides a very nice characterization of language learning from text. A further major step has been done by Shinohara [17] who showed rich classes to be learnable in the limit from text.

Additionally, it should be noted that many linguists strongly believe that children are only prepared to learn any human native language, i.e., a rather small but distinguished class of languages (cf. [10] for a more detailed discussion).

Taking these insights into account, it seems already plausible that one has to look for particular language classes when trying to gain a better understanding of the power and limitations of language learning from text.

Within this paper, we would like to point to some directions that seem promising in this regard. These directions are concerned with the language classes studied, the information presentation, the efficiency, and the size of the underlying terminal alphabet (or vocabulary).

We postpone the discussion of the first three items and discuss shortly the latter point here. Every natural language has a rather rich vocabulary as a short look into any dictionary confirms. So, it seems only natural to ask whether or not this fact may simplify or may complicate the underlying learning task. Research performed in the area of text classification may suggest that learning becomes more complicated (cf., e.g., Joachims [11]). On the other hand, there are some results obtained within the inductive inference paradigm pointing into the opposite direction (cf., e.g., Shinohara and Arikawa [18]). In particular, results surveyed in [18] suggest that learning is sometimes only possible if the underlying terminal alphabet is rather large.

Additionally, one may also ask to what extend the efficiency of learning algorithms does depend on the underlying terminal alphabet. When studying the learnability of pattern languages, we could prove that the number of examples necessary for successful learning *decreases* if the alphabet size *increases* (cf. [15, 16, 21]). However, so far we are not aware of any paper investigating the influence of the alphabet size systematically.

The paper is structured as follows. Section 2 presents preliminaries. Then we shortly recall some fundamental results concerning the learnability of languages from text. In Section 4 we outline some future directions.

2 Preliminaries

Unspecified notation follows Rogers [8]. By $\mathbb{N} = \{0, 1, 2, \ldots\}$ we denote the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. The cardinality of a set S is denoted by $|S|$. Let $\emptyset$, $\in$, $\subset$, $\subseteq$, $\supset$, and $\supseteq$ denote the empty set, element of, proper subset, subset, proper superset, and superset, respectively.

Let φ_0, φ_1, φ_2, $\ldots$ denote any fixed *acceptable programming system* for all (and only) the partial recursive functions over $\mathbb{N}$ (cf. Rogers [8]). Then φ_k is the partial recursive function computed by *program* k.

Gold's [7] model of learning in the limit allows one to formalize a rather general class of learning problems, i.e., learning from examples. For defining this model we assume any recursively enumerable set $\mathcal{X}$ and refer to it as the *learning domain.* By $\wp(\mathcal{X})$ we denote the power set of $\mathcal{X}$. Let $\mathcal{L} \subseteq \wp(\mathcal{X})$, and let $L \in \mathcal{L}$ be non-empty; then we refer to $\mathcal{L}$ and L as a *language class* and a *language*, respectively. Let L be a language, and let $t = (x_j)_{j\in\mathbb{N}}$ be any infinite sequence of elements $x_j \in L$ such that $\texttt{range(t)} := \{x_j \mid j \in \mathbb{N}\} = L$. Then t is said to be a *positive presentation* or, synonymously, a *text* for L. By *text*(L) we denote the set of all positive presentations for L. Moreover, let t be a positive presentation, and let $y \in \mathbb{N}$. Then, we set $t_y = x_0, \ldots, x_y$, i.e., t_y is the initial segment of t of length $y + 1$, and $t_y^+ := \{x_j \mid j \leq y\}$. We refer to t_y^+ as the *content* of t_y.

Furthermore, let $\sigma = x_0, \ldots, x_{n-1}$ be any finite sequence. Then we use $|\sigma|$ to denote the *length* n of σ, and let σ^+ denote the content of σ.

An *inductive inference machine* (abbr. IIM) is an algorithm that takes as input larger and larger initial segments of a text and outputs, after each input, a hypothesis from a prespecified *hypothesis space* $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$. The indices j are regarded as suitable finite encodings of the languages described by the hypotheses. A hypothesis h is said to describe a language L iff $L = h$.

A sequence $(j_n)_{n\in\mathbb{N}}$ of natural numbers is said to converge to number j if $j_n = j$ for all but finitely many $n \in \mathbb{N}$.

Definition 1. *Let $\mathcal{L}$ be any language class, and let $\mathcal{H} = (h_j)_{j\in\mathbb{N}}$ be a hypothesis space for it. $\mathcal{L}$ is called learnable in the limit from text with respect to $\mathcal{H}$ iff there is an IIM M such that for every $L \in \mathcal{L}$ and every text $t \in text(L)$,*

(1) *for all $n \in \mathbb{N}^+$, $M(t_n)$ is defined,*
(2) *there is a j such that $L = h_j$ and the sequence $(M(t_n))_{n\in\mathbb{N}}$ converges to j.*

The set of all language classes that are learnable in the limit with respect to $\mathcal{H}$ is denoted by $LimTxt_{\mathcal{H}}$. By $LimTxt$ we denote the collection of all language classes $\mathcal{L}$ for which there is a hypothesis space $\mathcal{H}$ such that $\mathcal{L}$ is learnable in the limit from text with respect to $\mathcal{H}$.

Note that instead of *LimTxt* sometimes *TxtEx* is used. In our notation, *Lim* stands for "limit." Since, by the definition of convergence, only finitely many data of L were seen by the IIM upto the (unknown) point of convergence, whenever an IIM identifies the language L, some form of learning must have taken place. For this reason, hereinafter the terms *infer*, *learn*, and *identify* are used interchangeably.

Note that Definition 1 does not contain any requirement concerning efficiency. We shall come back to this point later.

Many settings can be described by the scenario given in Definition 1. In particular, we can consider the special case that $\mathcal{X} = \mathbb{N}$ and let $\mathcal{L}$ be any subset of the collection of all recursively enumerable sets over $\mathbb{N}$. Let $W_k = \mathtt{domain}(\varphi_\mathtt{k})$, where φ_k is the partial recursive function computed by program k in the fixed acceptable programming system. Clearly, then W_k may be considered as a language. As a matter of fact, all W_k are recursively enumerable. In this case, $(W_k)_{k\in\mathbb{N}}$ is the most general hypothesis space. We use $\mathcal{E}$ to denote the set of all recursively enumerable languages.

Note that this setting has been used to study the general capabilities of different learning models which can be obtained by suitable modifications of Definition 1. There are numerous papers performing studies along this line of research (cf., e.g., [10, 14] and the references therein).

3 Learning Languages from Positive Data

Within this section, we shortly recall some fundamental insight concerning the learnability of language classes from text.

Based on Angluin [3] in Jain *et al.* [10] the following theorem is proved. Note that we neglect the computability of IIMs for a moment.

Theorem 2. *$\mathcal{L} \subseteq \mathcal{E}$ is identifiable if and only if for all $L \in \mathcal{L}$ there is a finite T_L such that for all $L' \in \mathcal{L}$, if $T_L \subseteq L'$ then $L' \nsubseteq L$.*

We are not going to repeat the proof of Theorem 2 here. But we like to point out the basic idea for showing the sufficiency. Let $L \in \mathcal{L}$ be the target language, let $t \in text(L)$, and let $n \in \mathbb{N}$.

Then the learner has to look for an index i such that

(a) i is an index for L; and
(b) $T_L \subseteq t_n^+ \subseteq L$.

The important part here is the topological structure of the language class to be learned which is expressed by the properties of the sets T_L. Clearly, this theorem directly implies Theorem 1.

In order to arrive at an IIM, one has to ensure that (a) and (b) can be handled algorithmically. So, if one would use the most general hypothesis space $(W_k)_{k\in\mathbb{N}}$ then Assertion (a) implies that one has to find an i such that $W_i = L$. Moreover, Assertion (b) requires a clever method for ensuring $T_L \subseteq t_n^+ \subseteq L$.

In her pioneering paper, Angluin [3] has proved this characterization theorem for *indexable language classes.* A language class is said to be an *indexable class* if it possesses an effective enumeration with uniformly decidable membership. Within the setting of indexable language classes she then showed the sets T_L to be recursively enumerable.

Moreover, when learning from text, a major problem one has to deal with is avoiding or detecting *overgeneralization.* An overgeneralization occurs if the learner guesses a proper superset of the target language. Using positive data alone, an overgeneralization cannot be detected. Nevertheless, as Angluin [3] has shown, overgeneralization is unavoidable if one wishes to exhaust the whole power of LimTxt, even within the setting of indexable language classes.

How can this happen? Assume an enumeration $(L_i)_{i\in\mathbb{N}}$ of the indexable language class, let L be the target and let i^* be the least index j such that $L = L_j$. That is, we have $L_{i^*} = L$ and $L \neq L_j$ for all $j < i^*$.

Looking at the characterization, one sees that overgeneralization may occur if some of the sets T_{L_j} with $j < i^*$ and $L \subset L_j$ are not yet completely enumerated.

IIMs that completely avoid overgeneralization are called *conservative.* Another way to look at conservative learning is to require that the IIM maintains its actual hypothesis at least as long at it has not seen data contradicting it.

Within the setting of indexable language classes, conservative learning can be characterized by posing a stronger requirement to the sets T_L, i.e., there must be uniform procedure g recursively generating all sets T_L for $L \in \mathcal{L}$ (cf. [23]). Here, by recursively generating we mean an algorithm that takes as input any index i (of the chosen enumeration) and outputs the complete set T_{L_i} and stops.

As we shall see below, if one aims at more realistic and efficient learning algorithms, it may be quite advantageous to have a conservative learner. The intuitive reason is that a conservative learner converges to its first correct guess in the sequence of all its guesses.

On the one hand, the results mentioned above are both beautiful and strong. They already provide a deep insight into the problem what can be learned from positive data.

On the other hand, they do not really contribute to the problem of how one can design practical learning algorithms. Even worse, they may suggest that one has to design learners along the line of testing something like Assertion (b) above. We therefore continue with some alternative approaches.

4 Towards more Realistic Learning Scenarios

The first approach we like to mention is learning from *good examples.* The idea of learning from good examples is to use finite sets of well selected examples instead of texts. The model of learning from good examples has been introduced by Freivalds, Kinber and Wiehagen [6] within the setting of learning recursive functions. Subsequently, Lange, Nessel and Wiehagen [12] have adopted this model to learning from positive examples of indexable concept classes.

Following [12], finite sets of good examples

1. are intended to be “important” ones,
2. are required to be computable from the languages to be learned,
3. are intended to be sufficient for learning rich classes of languages.

Then, instead of receiving growing initial sequences of a text, the learner receives any superset of the set of good examples for the target language. Furthermore, instead of converging in the limit to a correct hypothesis, now the learner is required to compute a *single* guess from the *finite* set it has received and to output a hypothesis which is correct for the possible infinite target language.

The resulting learning model is referred to as to *finite learning from good examples.* The requirement to learn from any superset of the set of good examples is introduced to avoid coding tricks. For example, if one has a given enumeration $(L_i)_{i\in\mathbb{N}}$ of the indexable target class, one could be tempted to provide just i examples to learn language L_i. So, such tricks are excluded.

Then, Lange, Nessel and Wiehagen [12] showed in particular that finite learning from good examples is exactly as powerful as conservative learning in the limit from text.

A prominent example known to be conservatively learnable in the limit from text is the class of all pattern languages.

Following Angluin [2] we define patterns and pattern languages as follows. Let $\mathcal{A} = \{0, 1, \dots\}$ be any finite alphabet containing at least two elements. Let $X = \{x_i \mid i \in \mathbb{N}\}$ be an infinite set of variables such that $\mathcal{A} \cap X = \emptyset$. *Patterns* are non-empty strings over $\mathcal{A} \cup X$, e.g., 01, $0x_0111$, $1x_0x_00x_1x_2x_0$ are patterns. The length of a string $s \in \mathcal{A}^*$ and of a pattern π is denoted by $|s|$ and $|\pi|$, respectively. A pattern π is in *canonical form* provided that if k is the number of different variables in π then the variables occurring in π are precisely $x_0, \dots, x_{k-1}$. Moreover, for every j with $0 \le j < k-1$, the leftmost occurrence of x_j in π is left to the leftmost occurrence of x_{j+1}. The examples given above are patterns in canonical form. In the sequel we assume, without loss of generality, that all patterns are in canonical form. By Pat we denote the set of all patterns in canonical form.

If k is the number of different variables in π then we refer to π as to a k*-variable pattern.* By Pat_k we denote the set of all k*-variable patterns.* Furthermore, let $\pi \in Pat_k$, and let $u_0, \dots, u_{k-1} \in \mathcal{A}^+$; then we denote by $\pi[x_0/u_0, \dots, x_{k-1}/u_{k-1}]$ the string $w \in \mathcal{A}^+$ obtained by substituting u_j for each occurrence of x_j, $j = 0, \dots, k-1$, in the pattern π. For example, let

$\pi = 0x_0 1x_1 x_0$. Then $\pi[x_0/10, x_1/01] = 01010110$. The tuple $(u_0, \dots, u_{k-1})$ is called a *substitution.* Furthermore, if $|u_0| = \dots = |u_{k-1}| = 1$, then we refer to $(u_0, \dots, u_{k-1})$ as to a *shortest substitution.* Let $\pi \in Pat_k$; we define the *language generated by pattern* π by

$$L(\pi) = \{\pi[x_0/u_0, \dots, x_{k-1}/u_{k-1}] \mid u_0, \dots, u_{k-1} \in \mathcal{A}^+\} .$$

By PAT_k we denote the set of all k*-variable pattern languages.* Finally, $PAT = \bigcup_{k \in \mathbb{N}} PAT_k$ denotes the set of all pattern languages over $\mathcal{A}$.

Note that deciding membership for the pattern languages is $\mathcal{NP}$–complete. Therefore, any learning algorithm testing membership will be infeasible in practice under the usual assumption that $\mathcal{P} \neq \mathcal{NP}$.

Fortunately, Lange and Wiehagen [13] have designed a pattern language learner for finitely learning from good examples which completely avoids membership tests. Also, Lange and Wiehagen [13] have shown that for every pattern π there is a set of good examples of cardinality linear in $|\pi|$.

In [21], we have dealt with the best-case, worst-case and average-case analysis of Lange and Wiehagen's [13] pattern language learning algorithm. The results obtained considerably improve Lange and Wiehagen's assertion concerning the minimal size of sets of good examples.

In particular, we proved the matching upper and lower bound of

$$\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1) \rfloor + 1$$

for the minimal size of sets of good examples for every k-variable pattern.

Note that this number *decreases* if the alphabet size *increases.* Thus, we have found a nice non-trivial example showing that a larger size of terminal (or constant symbols) does *facilitate* learning. Given that every natural language has a huge vocabulary, it may be worth to investigate the influence of the size of terminal symbols in a grammar to the complexity of learning. At a first step, this could be done within the setting of finite learning from good examples.

Using completely different ideas, we have also studied the learnability of one-variable pattern languages (cf. [15]). Though this has been done within the setting of learning in the limit from randomly generated texts, the results are in some sense similar. Our algorithm could be easily updated to finite learning from good examples. Then again, one easily sees that a larger alphabet size considerably reduces the minimal size of sets of good examples.

There is another point to be mentioned within this context. As a matter of fact, the algorithms sketched above are not *consistent.* Here consistency means that the intermediate hypotheses output by the learner do correctly reflect the data seen so far. Though consistency seems to be a very natural requirement at first glance, it is not as many results show. We refer the interested reader to Wiehagen and Zeugmann [19] for a detailed discussion.

In this context, we would also like to point the reader to the discussion concerning human languages and comparative grammar. As outlined in Jain *et al.* [10], theories of linguistic development are closely related to theories of comparative grammar. As far as natural languages are concerned, it is certain that children can master it in a few years on the basis of rather casual and unsystematic exposure to it. So, there must be some properties of natural languages making them particularly suited for humans to be learnable.

Though I am not a linguist, I have observed on my children the following. During the first two years, they somehow learned to distinguish words in any text spoken. Around the age of three, they had acquired a possibly simplified grammar allowing them to express themselves in simple sentences of three or four words. Then, from maybe three to six, they enlarged their vocabulary at an amazing speed on a daily basis. Interestingly, with their growing vocabulary they also went on to master more and more complex syntactical constructs. So, it would be very interesting to investigate to what extend the growing vocabulary is necessary to ensure the whole learning process.

The other point I have observed is that humans are not consistent learners.

Furthermore, humans are for sure not good in learning their mother tongue from every text for it. Instead, we may assume that humans learn from randomly generated text. Adopting this idea, we studied the learnability of the pattern languages from randomly generated text for a large class of probability distributions. In a first step, we analyzed the *expected* number of examples needed until successful learning. Our learner is both conservative and *rearrangement independent*. A learner is said to be rearrangement independent iff its output depends only on the content and length of its input. For such learners we could show that the probability to deviate from the expected number of examples until convergence is *exponentially shrinking*. Finally, a bit of additional domain knowledge concerning the underlying probability distributions allows one to arrive at a *stochastic finite learner*. A stochastic finite learner is fed randomly generated strings from the target pattern language. Additionally, it takes a confidence parameter δ as input. But in contrast to learning in the limit, the stochastic finite learner decides itself how many examples it wishes to read. Then it computes a hypothesis, outputs it and stops. The hypothesis output is correct for the target with probability at least $1-\delta$. We refer the interested reader to [16] for the details. As a matter of fact, stochastic finite learning incorporates the requirements concerning efficiency that have been missing in Gold's [7] model of learning in the limit. And it inherits the property stated above that the number of examples needed decreases if the alphabet size increases.

Last but not least, we would like to point the reader to a direction of research that deserves attention, i.e., the design and analysis of algorithms learning subclasses of context-free grammars in the limit from text. As already stated in Theorem 1, the whole class of context-free grammars is not learnable in the limit from text. So, one has to look for suitable subsets. While subsets of regular languages have attracted considerable attention within the grammatical inference community, so far not too much work has been done for subclasses of context-free grammars (cf., e.g., Adriaans *et al.* [1], Yokomori [20]).

Recently, Clark and Eyraud [4] presented a learning algorithm for a subclass of context-free languages which they called *substitutable* languages. Roughly speaking, substitutable languages are those context-free languages L which satisfy the condition that $lur \in L$ if and only $lvr \in L$ for pairs of strings u, v. Intuitively, if u and v appear in the same context, there should be a non-terminal generating both of them.

The learning problem is then considered in the setting of identification in the limit from text with polynomial time and data introduced by de la Higuera [5]. For the sake of better readability we recall the definition here in the form used by Clark and Eyraud [4]. Within this definition, $L(R)$ denotes the languages described by representation R.

Definition 2. *A representation class $\mathbb{R}$ is identifiable in the limit from positive data with polynomial time and data iff there exist two polynomials $p(), q()$ and an algorithm A such that*

(1) *Given a positive sample S of size m A returns a representation $R \in \mathbb{R}$ in time $p(m)$.*
(2) *For each representation R of size n there exists a characteristic set CS of size less than $q(n)$ such that if $CS \subseteq S$, A returns a representation R' such that $L(R) = L(R')$.*

As far as the characteristic sets are concerned, it is intuitively sufficient to think of them as sets of "good examples." Once the learner has seen a super set of the characteristic set, it converges.

The point I found most interesting in the approach made by Clark and Eyraud [4] is that they looked for a property of context-free languages that facilitates learning, i.e., substitutability.

References

[1] P. W. Adriaans, M. Trautwein, and M. Vervoort. Towards high speed grammar induction on large text corpora. In *SOFSEM 2000: Theory and Practice of Informatics, 27th Conference on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic, November 25 - December 2, 2000, Proceedings*, pages 173–186, 2000.

[2] D. Angluin. Finding patterns common to a set of strings. *J. of Comput. Syst. Sci.*, 21(1):46–62, 1980.

[3] D. Angluin. Inductive inference of formal languages from positive data. *Inform. Control*, 45(2):117–135, May 1980.

[4] A. Clark and R. Eyraud. Identification in the limit of substitutable context-free languages. In *Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 2005, Proceedings*, volume 3734 of *Lecture Notes in Artificial Intelligence*, pages 283–296. Springer, Oct. 2005.

[5] C. de la Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning*, 27:125–138, 1997.

[6] R. Freivalds, E. B. Kinber, and R. Wiehagen. On the power of inductive inference from good examples. *Theoret. Comput. Sci.*, 110(1):131–144, 1993.

[7] E. M. Gold. Language identification in the limit. *Inform. Control*, 10(5):447–474, 1967.

[8] J. H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw–Hill, New York, 1967.

[9] J. Hopcroft and J. Ullman. *Formal Languages and their Relation to Automata.* Addison-Wesley, Reading, Massachusetts, 1969.

[10] S. Jain, D. Osherson, J. S. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory, second edition.* MIT Press, Cambridge, Massachusetts, 1999.

[11] T. Joachims. *Learning to Classify Text using Support Vector Machines: Methods, Theory, and Algorithms.* Kluwer Academic Publishers, Dordrecht, 2002.

[12] S. Lange, J. Nessel, and R. Wiehagen. Learning recursive languages from good examples. *Annals of Mathematics and Artificial Intelligence*, 23(1/2):27–52, 1998.

[13] S. Lange and R. Wiehagen. Polynomial-time inference of arbitrary pattern languages. *New Generation Computing*, 8(4):361–370, 1991.

[14] D. N. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Massachusetts, 1986.

[15] R. Reischuk and T. Zeugmann. An average-case optimal one-variable pattern language learner. *J. Comput. Syst. Sci.*, 60(2):302–335, 2000.

[16] P. Rossmanith and T. Zeugmann. Stochastic finite learning of the pattern languages. *Machine Learning*, 44(1/2):67–91, 2001.

[17] T. Shinohara. Rich classes inferable from positive data: Length-bounded elementary formal systems. *Inform. Comput.*, 108(2):175–186, 1994.

[18] T. Shinohara and S. Arikawa. Pattern inference. In *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 259–291. Springer, 1995.

[19] R. Wiehagen and T. Zeugmann. Learning and consistency. In *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 1–24. Springer, 1995.

[20] T. Yokomori. Polynomial-time identification of very simple grammars from positive data. *Theoret. Comput. Sci.*, 298(1):179–206, 2003.

[21] T. Zeugmann. 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.

[22] T. Zeugmann and S. Lange. A guided tour across the boundaries of learning recursive languages. In *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 190–258. Springer, 1995.

[23] T. Zeugmann, S. Lange, and S. Kapur. Characterizations of monotonic and dual monotonic language learning. *Inform. Comput.*, 120(2):155–173, 1995.