Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 38 (2005) 1332–1348
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A bibliographical study of grammatical inference

Colin de la Higuera*

EURISE, University of Saint-Etienne, France

Received 15 November 2004

Abstract

The field of grammatical inference (also known as grammar induction) is transversal to a number of research areas including machine learning, formal language theory, syntactic and structural pattern recognition, computational linguistics, computational biology and speech recognition. There is no uniform literature on the subject and one can find many papers with original definitions or points of view. This makes research in this subject very hard, mainly for a beginner or someone who does not wish to become a specialist but just to find the most suitable ideas for his own research activity. The goal of this paper is to introduce a certain number of papers related with grammatical inference. Some of these papers are essential and should constitute a common background to research in the area, whereas others are specialized on particular problems or techniques, but can be of great help on specific tasks.

Keywords: Grammatical inference; Grammar induction

1. Introduction

1.1. The field

Grammatical inference is transversal to a number of fields including machine learning, formal language theory, structural and syntactic pattern recognition, computational linguistics, computational biology and speech recognition.

In a broad sense a learner has access to some data which is sequential or structured (strings, words, trees, terms or limited forms of graphs) and is asked to return a grammar that should in some way explain these data. The learner is at least partially automatic, and can also be called an inference machine, or a learning algorithm. The induced (or inferred) grammar can then be used to classify unseen data, compress these data or provide some suitable model for these data. Typical features of the problem are:

- the data, usually composed from a finite alphabet; it is thus usually discrete, as opposed to numerical; but on the other hand the unbounded length of strings offers a higher complexity for classification tasks than with usual symbolic data;
- the sort of result: a grammar or an automaton, traditional objects much studied by computer scientists. Such objects have the added advantage of being understandable. One learns intelligible concepts, not black boxes; in fields where human experts need to be able to derive new knowledge from what the computer provides, this is undoubtedly a key feature;
- the hardness of even the easiest of problems. In usual machine learning settings, even the easiest of the problems are usually classified as hard;
- the variety of potential applications;
- the small number of industrial applications where grammar induction is successful.

* Corresponding author. EURISE, Faculte des Sciences et Techniques, Universite Jean Monnet, 23 rue Paul Michelon, Cedex 02 Saint-Etienne F-42023, France.
E-mail address: cdlh@univ-st-etienne.fr

0031-3203/$30.00
doi:10.1016/j.patcog.2005.01.003

We nevertheless strongly believe that the situation should change in the next few years: indication of this is given by the wide amount of applications where these techniques are at least partially used, and the recent successes of the field. The other grounds for this belief are that purely statistical methods have a natural bound, that can only be overcome by a close use of the structural nature of the data that is being manipulated. And the discovery of this structure is where grammatical inference can help.

1.2. Where should one start from?

The proceedings of the different International Colloquium on Grammatical Inference (ICGI), held at Alicante [1], Montpellier [2], Ames [3], Lisbon [4] and Amsterdam [5] are good places to find technical papers. The webpage of the grammatical inference community [6], and those of the related communities can be used to find most of the papers in the field: the *computational learning theory* (COLT) webpage [7] or the *algorithmic learning theory* (ALT) webpage [8] can provide good lists of earlier papers with the machine learning perspective. Important key papers setting the first definitions and providing important heuristics are those by Fu [9] or Fu and Booth [10]. The structural and syntactic pattern recognition approaches can be found for instance in Miclet's book [11] or in the survey by Bunke and Sanfeliu [12], with special interest in Miclet's chapter [13].

Surveys or introductions to the subject have been published over the years, some of which are those by Lee [14], Sakakibara [15], Honavar and de la Higuera [16], and de la Higuera [17].

Scientists in the area may need access to some good textbooks on related areas: on formal languages books by Harrison [18] or by Hopcroft and Ullman [19] give most of the elementary definitions and results. Parsing issues are discussed in Aho and Ullman's textbook [20]. On Machine Learning the books by Mitchell [21], Natarajan [22] or Kearns and Vazirani [23] all give elements that are of use to derive grammar induction results. Another place where structural pattern recognition issues are discussed is Gonzalez's book [24]. An early book with many important mathematical results on the subject of automata inference is that by Trakhtenbrot and Barzdin [25].

1.3. Organization of this paper

Because of the wide amount of subjects grammatical inference is related to, a technical paper would require many definitions, special notations and have to quote dozens of results from formal language theory, inductive inference, probability theory or grammar induction. We have chosen here to only introduce the subject and the papers from the field. Therefore hardly any formalism will be provided in this paper, and mathematics will be described through text only. This informality should not induce the reader to believe that the field of grammatical inference has no strong mathematical basis. On the contrary, formalization is a strong issue, and even in practice, algorithms with sound mathematical properties obtain better results than the heuristics as has been shown during the ABBADINGO competition [26]: a variety of open problems were set for the community to solve: they involved identifying a hidden automaton from raw data. A number of techniques were tested, with the better results obtained from algorithms based on characterizable methods [27].

We present in Section 2 the theory of the field with special interest in those results related to inductive inference (in Section 2.1) and on the tractability issues (in Section 2.2). The special case where one can question a teacher about the grammar to be learned has active learning as theoretical framework, and is described in Section 2.3. In Section 2.4 the main results concerning the distribution free PAC model for grammars are given, and in Section 2.5 we study the case where Kolmogorov complexity is used to define simple distributions, leading to models called *Simple*-PAC or PACS.

In Section 3 the algorithmics are described. Special attention is given to the case of automata learning (Section 3.1) and context-free grammar induction (Section 3.2). The later problem is closely linked with that of learning tree automata (Section 3.3). As most of these problems can be proved intractable in an associated combinatorial search setting, artificial intelligence techniques have been tested with varying degrees of success (Section 3.4). A special case is that of learning from positive examples only, which is a usual case in applications, moreover in those of pattern recognition: this is studied in Section 3.5. In the same sort of setting learning probabilistic automata or grammars has sometimes been successful and certainly is today an important open line of research; this is discussed in Section 3.6. In Section 3.7 some other formalisms (patterns, categories) that can be used to represent languages are described.

In Section 4 the main applications of grammar induction are described: robotics (4.1), pattern recognition (4.2), computational linguistics (4.3), speech recognition (4.4), automatic translation (4.5), computational biology (4.6) and a variety of other applications (4.7) including inductive logic programming, document management, compression, agents, time series and music. We will conclude with open questions and new trends of research in Section 5.

2. The theory

The main focus of research in the field of grammatical inference has been set on learning regular grammars or deterministic finite automata (DFA). Algorithms have been provided dealing with learning such grammars in a variety of settings (and will be discussed in more detail in Section 3.1): when both examples and counter-examples are provided or when the learning algorithm is allowed to question some teacher.

Reasons justifying that most attention has been focused on this class of grammars are that this problem may seem simple enough but theoretical results make it already too hard for usual *Machine Learning* settings (see Section 2.4); and for those learning paradigms in which DFA learning is possible, such as active learning or learning from polynomial time and data (see Section 2.2), the positive results do not seem to hold for the next level of the Chomsky hierarchy, the context-free grammars.

In the framework of *active learning* it is known since Angluin's results [28] that DFA can be inferred through a polynomial use of a *Minimal Adequate Teacher* (MAT). It is conjectured (also by Angluin [29]) not to be the case for richer classes. In the setting of learning from a bunch of examples, this is even clearer as DFA can be polynomially identified from time and data, but not context-free grammars nor non-deterministic finite automata [30].

2.1. Inductive inference

Inductive inference deals with the problem of identifying a function given some of its values. It can be set in a variety of manners, but the question ends up by being that of *identification* of some hidden function. Early research in this field is due to Solomonoff [31]. A first convincing model for the case of grammatical inference was introduced by Gold [32]: identification in the limit. The setting of this model is that of on-line, incremental learning. After each new example the learner (called *inductive machine* in this setting) must return some hypothesis. Identification is achieved when the learner returns a correct answer and does not change its mind afterwards. Of course, this must hold for any target function (grammar) in the class, and any admissible presentation of the examples. There are two traditional settings: *learning from text*, where only the positive instances are given to the learner, but each one of these must appear at some point or another of the presentation, and *learning from informant*, where examples are labelled as positive or negative and each possible string must appear. It is important to note at this point that there can be a difference between learning or identifying a language and identifying a grammar. Indeed, in the usual case where there may be an infinite number of equivalent grammars (two grammars are said to be equivalent if they generate the same language) one may have the possibility of identifying a language, but never a grammar, at least from raw data only.

Gold further developed his results [33] which mainly state that:

- if we are given examples and counter-examples of the language to be identified, and each individual string is sure of appearing, then at some point the inductive machine will return the correct hypothesis;
- if we are given only the examples of the target, then identification is impossible for any super-finite class of languages, i.e. a class containing all finite languages and at least one infinite language;
- Angluin [34] strengthens this result by proving that classes for which an infinite sequence of languages strictly included one into each other can be constructed (this has been introduced by Wright [35] and is called *infinite elasticity*) cannot be identified.

The field has continued to deal with important questions: what classes of functions can be identified? Can we modify the paradigm, for instance requiring to refute the class, stating that no function in the class is correct? Is it possible to learn in a monotonic way (converging slowly) [36]? Alternative models dealing with approximation issues have also been proposed (for instance by Wharton [37]).

A good survey of the field is due to Angluin and Smith [38]; COLT and ALT proceedings is where most results can be found. Both communities have their webpages [7,8].

2.2. Polynomial questions

Once admitted that DFA can be identified in the limit [33], a reasonable question is that of being able to do so in polynomial time, even if the meaning of this also needs exploring. A preliminary yet essential question is that of the measure of the data size. Pitt [39] discusses this point with care: the size of a set of strings (or trees) must be polynomially linked with the size of its encoding; for alphabet sizes larger than one this means that taking the size as the sum of the lengths of all strings in the set, or some function of the number of strings and the length of the longest string are two reasonable ways of counting.

Negative results concerning the combinatorics of the problem have been given by Gold [33], who proves that the problem of finding the smallest DFA consistent with a given set of strings is NP-hard. Angluin [40] proves that this is the case even when the target automaton has only 2 states (this peculiar result is quoted by Pitt and Warmuth [41]), or even when only a very small fraction of all the strings up to length n, where n is the size of the target, is absent. Trakhtenbrot and Barzdin [25] had shown previously that when all such strings were present the problem was tractable.

This is not enough to obtain a direct proof that learning in polynomial time is impossible but Angluin proves the hardness of the task even using *membership queries* [42] (a string can be proposed to the oracle who must answer if the string belongs to the target language or not) or *equivalence queries* [43] (a grammar is proposed to the oracle who answers yes if the hypothesis is equivalent to the target, and provides a counter-example if not). In the second case, Angluin introduced the combinatorial notion of *approximate fingerprints* of independent interest: these correspond to a subset of hypothesis out of which only a small fraction can be excluded, given any counter-example, resulting in the necessity of using an exponential number of equivalence queries to isolate a single hypothesis. Gavaldà studies this notion with care in [44]. Pitt [39] uses this result to prove the intractability of the task of identifying DFA with a polynomial number of

mind changes only. The problem is proved to be hard [45], and by typical reduction techniques [46] is proved complete (hardest) in its class. But Pitt's model may be itself too demanding, as it is closely linked with Littlestone's learning model [47].

Based on teaching models [48,49], de la Higuera proposes the model of *identification in the limit from polynomial time and data* [30]. This model can be seen as intermediate between those of identification in the limit and PAC in the sense that harder classes such as context-free grammars or non-deterministic finite automata are not learnable whereas DFA are. For a class of grammars to be learnable in this setting it is required that each grammar in the class admits some robust characteristic set of polynomial size. The set is characteristic in the sense that from it the learning algorithm will return some equivalent grammar, and robust in the sense that this remains true whenever this characteristic set is included in any correctly labelled learning set.

Valiant's PAC model is a model considered indicative of how hard learning is for a specific class. In the case of grammar induction mainly negative results are known. The model can be made easier by limiting the classes of possible distributions. This is done by means of *Simple*-PAC or PACS settings. We will discuss these in Sections 2.4 and 2.5.

2.3. Active learning

Active learning is about learning with queries asked to an oracle. The model has been introduced by Angluin [50]. In a setting more general than that of grammar induction, bounds on the number of queries needed to learn can be found in Ref. [51], whereas Angluin [42] gives an overview of various query systems.

A recent survey paper of the field (also by Angluin) is Ref. [29], where the openness of the problems related to non-deterministic finite automata or context-free grammars is recalled. Membership and equivalence queries together form a *Minimal Adequate Teacher*. With this system Angluin proves that regular languages can be identified with only a polynomial amount of queries [28]: the proposed algorithm is called L_*. Balcazar et al. [52] study how to use more of one sort of queries or the other, but the tradeoff is that you need an exponential number of membership queries in order to simulate an equivalence query. In the case of context-free grammars the negative proofs by Angluin and Kharitonov [53] with MATS are related to cryptographic assumptions. On the other hand, if structural information is available, Sakakibara proves the learnability of the class of context-free grammars in this model [54].

Returning to the DFA case, it should be noticed that the oracle has no reason to return the counter-example the learner really needs. The case where this is not so, and the oracle returns the smallest counter-example is studied in Ref. [55].

Sufficiency of query systems is presented by Bshouty et al. [56]. Angluin presents a sufficient condition (approximate fingerprints) for languages to be learnable by equivalence queries in Ref. [43]. The condition is proved necessary by Gavaldá [44]. If a class does not have approximate fingerprints, then (unless P = NP) it can be learned in polynomial time through equivalence queries. Obviously, the result does not apply to any well-known language classes.

The model has received considerable attention and there are many papers on learning with different sorts of queries. Sakakibara [57] learns context-free grammars from queries; Yokomori [58] learns 2-tape automata from both queries and counter-examples, and in Ref. [59] non-deterministic finite automata from queries also in polynomial time, but depending on the size of the associated DFA; Vilar extends queries to translation tasks in Ref. [60], Maler and Pnueli [61] learn Büchi automata from queries on infinite strings.

2.4. PAC learning of languages

Exact learning has always been considered a hard to achieve goal. In a setting that is meant to represent a more realistic situation, Valiant [62] introduced the *probably approximately correct* (PAC) model, which has also been studied in the context of grammar induction (see Natarajan's book [22], with also a nice discussion about these issues and special interest to the DFA case).

An unknown distribution over all possible examples exists, and examples are sampled under this distribution. Learning is done from this sample, and the result is tested under the same distribution. It is required to be able to learn under any distribution, but, since one may be unlucky during the sampling processes, exactitude is not required: a small (ε) error is permitted, and one should not do worse than this error rate in more than very few cases (a fraction of all possible cases, bounded by δ). Number and size of examples should be polynomial in $1/\varepsilon$, $1/\delta$, and the size of the target; the run-time complexity needs to be polynomial in the same parameters, plus the length of the longest example that has been seen.

In this PAC setting few positive results (for grammatical inference) are known. One way to obtain positive results is by means of defining subclasses for which the VC-*dimension* (studied by Ishigami and Tani [63] for the case of DFA) can be made finite. This is done by Bhattacharyya and Nagaraja [64]: terminal distinguishable regular languages are defined as grammars that are backward deterministic and strongly unambiguous. Another positive PAC-type of result is given by Ron et al. [65], when restricting to stochastic acyclic DFA, with *distinguishable states* (for any 2 states there is a string whose difference in probability when using these states as initial is above a given threshold): see Section 3.6.1 for details.

Even for the case of DFA, most results are negative: Kearns and Valiant [66] linked the difficulty of learning DFA with that of solving cryptographic problems believed to be intractable (a nice proof is published in Kearns and Vazirani's book [23]).

2.5. Simple PAC

The PAC model introduced by Valiant [62] has also received criticism. For instance the obligation to learn under any distribution may seem too hard. It also may be the case that some concepts (those "hard" to encode) may not be learnable whereas the more "natural" ones may be learnable.

Based on Kolmogorov complexity [67], *simple* distributions are those where simple (admitting short descriptions) strings have high probability [68]. This is the setting for Simple-PAC learning. Denis et al. further restricted the model (it is called PACS) by allowing a benign teacher to select the learning sample, following a distribution that "knows the concept to be learned". Technically, it requires replacing the Kolmogorov complexity by the conditional Kolmogorov complexity. Parekh and Honavar [69] proved this was possible for the case of DFA.

3. The algorithmics

Once most negative results described in the previous section induced people to believe that nothing could be done, pragmatic considerations (problems that need solving, data from which some automaton or grammar has to be inferred) required new techniques for grammatical inference. Some of these were mere heuristics, based on good properties regular languages had, and that might be detected: Miclet describes a variety of these early heuristics in [11] or [13]. We will present these methods and algorithms by grouping them in classes corresponding to the problems they are intended to solve.

3.1. About DFA learning

The most well-known problem in grammatical inference is that of learning a deterministic finite automaton from both positive and negative data. The fact that the associated combinatorial problem (is there a DFA with at most n states consistent with this data?) is intractable was established by Gold [33] and Angluin [40] in 1978 and improved by Pitt and Warmuth [41] where it is shown that even finding a polynomially larger DFA than the minimum DFA, consistent with the data, is NP-hard. Gold [33] gave a first algorithm that works when the data is sufficient but refuses to generalize if it is not (even though no algorithm can decide in which of the two cases one stands). Trakhtenbrot and Barzdin proved that in the special case where all the data up to a certain length are presented there exists an algorithm that can identify DFA. But that amount of data is too large for the algorithm to be of practical use.

Algorithm RPNI, proposed by Oncina and García, is capable of generalizing (even if perhaps very badly in the worse cases), and identifying DFA. A nice lattice setting of the problem is presented by Dupont et al. [70]: nodes of the lattice correspond to all the different automata that can be obtained by merging states from the maximal canonical automaton (a star-like automaton that exactly recognizes the positive data from the sample). The number of nodes, and hence the size of the lattice, is exponential in the size of the initial automaton. Even if the lattice representation is of no practical use (full exploration of this lattice is doomed, due to its size), it does provide us with means to prove convergence of new algorithms. Lang [71] showed experimentally that depth-first techniques commonly used to cope with the size of the lattice do really badly until a certain (but exponential in the size of the target) quantity of data is given. Lang's results clearly depend strongly on his particular experimental setting, but even if taken as a worse-case setting, join the theoretical arguments against the possibility of being able to learn DFA efficiently in the general case.

In 1997, the ABBADINGO competition [26] restored interest in the problem of DFA inference, and although a neural network technique seemed to do well, at the end, an *evidence driven* technique developed by Price, based on classical state merging, won [27]. The idea was to try different merges but keep the one that had highest score. A (cheap) alternative to evidence driven heuristics is data driven heuristics [72], where the idea is to try merging those states through which most information is known. Some problems from that competition are still open.

Artificial intelligence techniques were used on this problem also, as for instance TABU search by Giordano [73] or genetic algorithms by Dupont [74]. Other artificial intelligence techniques are described in Section 3.4.

Another idea is to learn a non-deterministic automaton instead. In this line Denis et al. [75] offer an interesting approach, and Yokomori [59] learns non-deterministic finite automata, but from queries and counter-examples.

Work has continued since, with best results obtained to date (on large automata) by de Oliveira [76] and Lang [77]. Parallel versions of the basic algorithms have been studied by Balcazar et al. [78] but do not seem to have been tested on large problems.

3.2. The case of context-free languages

Learning the entire class of the context-free languages seems to be intractable whichever learning model you choose. Nevertheless, the class can of course be identified in the limit, and the question as to whether it can be identified with a polynomial number of queries to a MAT is still an open question, but widely believed to be also intractable [29]. One barrier is that of determinism, and the other that of linearity, motivating early studies for the class of linear languages [79]. But it should be noticed that in the setting of identifying from polynomial time and data this class is still not learnable [30].

One first option to obtain positive results is that of extending results from the class of the regular languages (when represented by DFA). This is the line followed for *even linear languages* for which a number of results is known: Takada

[80], Sempere and García [81] and Mäkinen [82] have all worked on this class and give similar results by different techniques. Even linear languages are generated by grammars where the rules are balanced: the right hands are composed either of terminal symbols only, or are of the form uTv where only T is non-terminal and u and v have identical length. Following this trend, other results concerning this class of languages (or similar) are an extension to a hierarchy of linear languages [83], the case where only positive strings are available [84] (but then only a sub-class is identifiable), and the case where the positive information is structural (if you know where the *center* of the strings is) [85]. Different surveys on the subject have been written by Yokomori [86], Lee [14] or Sakakibara [15]. In Ref. [87] a larger class, that of deterministic linear grammars, is proved by de la Higuera and Oncina to be identifiable from polynomial time and data. A general way of detecting if this is the case for other classes of grammars is given by the same authors in Ref. [88]. Giordano [89] proposes to see the problem as that of an exhaustive search in a lattice defined by the *Reynolds cover* over grammars in Chomsky normal form.

In a series of papers, Sakakibara gives techniques to learn context-free grammars from structured data [54], data containing positive structured data [90], unstructured data by genetic algorithms [91] and data containing some structure again by genetic algorithms [92]. There are also a number of very specific results that will be of interest to specialists: Ishizaka [93] learns another restricted class of context-free grammars, that of the simple deterministic grammars. It should be noticed that these grammars are not linear.

A special case should be made of Nevill-Manning and Witten's algorithm SEQUITUR [94]. Although it cannot be included into the class of grammar induction algorithms, as it has no generalization capacity, it is an elegant way of deducing a context-free grammar from just one (usually very long) sentence. This grammar can then generate just one string: the original one. Running in linear time and space, SEQUITUR is more than just a compression technique as it also explains the data it has to compress by giving its structural nature.

3.3. Tree automata

Tree automata are the direct extension of DFA and NFA for trees instead of strings. They also provide a smooth link between automata and context-free grammars. Learning tree automata has already been dealt with in Fu and Booth's survey [10].

There are very strong links between learning context-free grammars from bracketed data (or the actual skeletons or parse-trees without inner labels) and learning regular tree grammars [54].

The advantages are nevertheless that a deterministic case exists, allowing to re-use results from DFA learning in this setting. An extension of RPNI to deal with trees is provided by García and Oncina [95]; a tree version of the best known algorithm for stochastic DFA (ALERGIA: see Section 3.6.1) is proposed by Carrasco et al. [96]. For the case of learning from positive structural data only, Knuutila has presented a state of the art in Ref. [97]; for the same problem, Fernau [98] extends to tree automata results allowing to state when a class is identifiable from positive data only.

3.4. Artificial intelligence approaches

The number of possible automata or grammars that might be adequate with a given learning sample is such that artificial intelligence techniques might be an answer. The combinatorics of the problem of grammatical inference involve mainly the space in which things should be looked for. In the case of the regular languages, this is described by Dupont [74]. The VC-dimension of finite automata is studied by Ishigmai and Tani [63]. The search space for the context-free grammar problems has hardly been studied, but it has been seen as a version space by Vanlehn and Ball [99], described in Ref. [89] by Giordano, and also used by Langley and Stromsten [100] by means of a simplicity bias and a representation change. Dupont used genetic algorithms to deal with a population of DFA, in the partition lattice setting defined previously (Section 3.1). Using TABU search has also been looked into by Giordano [73]. In the case of context-free grammars, genetic algorithms (this time on the rules) were tried by Sakakibara and Kondo [91]. Experiments suggest that the knowledge of part of the structure (some parenthesis) may help and reduce the number of generations needed to identify [92]. A rough set approach is proposed by Yokomori and Kobayashi [101]. Oliveira and Silva [76] proposed algorithm BIC that attempts to merge states, starting from the prefix tree acceptor, but that can back-track intelligently, so as to avoid testing consistency for automata for which inconsistency should be derived from prior testing; this is done through conflict diagnosis. The domain they are concerned with is that of digital circuit design (synthesis of a finite state controller from descriptions of observed input/output signals). In this case, the problem is not so much a problem of approximately learning some good enough machine, as that of exactly discovering the correct smallest machine, and doing this from as little data as possible. Therefore, the benchmarks they use correspond to hard problems, but in their class. It should be noticed that due to the characteristics of the setting, the sizes of the inferred automata are necessarily limited: from 10 to 20 states.

3.5. Learning from positive data only

The problem of learning from positive data alone is probably the most practical of grammatical inference settings. There are many papers dating from the 1970s and the 1980s dealing with this subject, published mainly inside the pattern recognition community. A good survey is that of Knuutila [102], who classifies the methods into heuristic and characterizable ones. Characterizable methods for the entire

class of the regular languages are limited by Gold's result [32]. Angluin [103] defines reversible languages and gives an algorithm for this sub-class of the regular languages. Angluin gives other theoretical general results (as to sufficient conditions to be able to identify from positive examples only [34]).

García and Vidal [104] give another algorithm for the class of *k*-testable languages, also known as local languages and that correspond to an unnecessarily stochastic version of *n*-grams. In the same line of research even linear languages can be specialized in such a way as to be able to identify them from positive examples only [84]. Denis et al. [105] define sub-classes of languages that can be identified through non-deterministic finite automata by positive data only. An elegant generalization of these results is proposed by Fernau [106].

3.6. When we talk about probabilities

For a number of reasons the above models can be unsatisfying: incapacity of dealing with noisy data, hardness to learn from positive data only, true distributions that are not arbitrary but context dependent.

A classical idea is to introduce probabilities in the model. The most famous of such probabilistic finite state machines are *hidden Markov models* (HMMs). Alternative machines, and closer to objects from formal language theory are stochastic automata and stochastic context-free grammars. Stochastic context-free grammars are context-free grammars where to each rule is associated a probability, in such a way that the sum of the probabilities of all possible expansions of any non-terminal symbol is 1. A stochastic automaton is the graphical representation of a stochastic regular grammar, with determinism and non-determinism defined as usual.

Stochastic grammars and automata thus define a distribution over all strings. Specific parsing algorithms have been defined, and there has been quite a lot of work over the years around these models (see for instance Refs. [107,108]). There are lot of difficult issues involved with learning such models: the question of correctly estimating the probabilities is often hard, smoothing becomes an important question, and finally, one should not forget that the underlying hypothesis is not that the language is regular or context-free, but that the distribution is. Therefore, learning a stochastic automaton involves a modification of bias from what has been presented before: even if the underlying language is regular, the distribution may not be.

3.6.1. Stochastic automata

Stochastic finite state machines have been introduced more than 30 years ago [107,108]. They can be used in a variety of settings, and have seen independent theories developed because of that. Common features involve the algorithmics used for parsing with these machines: the VITERBI algorithm [109] computes for a given string the most probable parse. But finding the most probable string (in the non-deterministic case) is NP-hard [110]. Baum [111] gives techniques enabling to estimate the probabilities of a given model from a fixed bunch of data. Casacuberta [112] explores the relationship between probabilistic finite automata and HMMS.

Negative results concerning the possibility of approximating given distributions by means of stochastic automata can be found in Abe and Warmuth's paper [113]. Negative results concerning the inference of these automata (in a very general setting) are by Kearns et al. [114]. Description of the process of identification in the limit with probability 1, and useful tools can be found in Angluin's unpublished report [51].

The inference of these automata was made possible by different techniques: algorithm ALERGIA [115] learns them from a polynomial amount of data, but the proof in that paper is not convincing. For a special class of automata, those that are acyclic (i.e. the language is finite), Ron et al. [116] give an algorithm that provably converges and even PAC-learns if a specific condition called μ-distinguishability is met. Stolcke and Omohundro [117] follow a Bayesian approach in order to learn stochastic automata, but no proof of identification is given. Young-Lai and Tompa [118] use ALERGIA for document classification, improving the algorithm to take into account better its parameters, and avoiding those merges not substantiated by enough evidence. A data driven heuristic [72] based on ALERGIA is presented by Goan et al. [119]: the authors claim good experimental results. Basing oneself on a dynamic programming computation of the relative entropy [120], algorithm MDI [121] does better than ALERGIA on computational linguistics and speech tasks. The heuristic can also admit a data driven approach and then does even better [122]. Alternatively, one can use neural networks to infer stochastic automata [123]. Carrasco and Oncina [124] give a formal proof of the identification in the limit with probability one of a weaker version of ALERGIA, RLIPS, whereas de la Higuera and Thollard [125] extend this proof to ALERGIA (*Stern-Brocot* trees are used to identify the probabilities). The case where the strings in the learning sample are drawn without repetitions is studied by de la Higuera [126]. Finally, one important issue is that of smoothing [127]: the inferred automaton or grammar can give null probabilities to some unseen events, and these events might turn up: the effects may be disastrous. Knowing how to redistribute part of the mass of probabilities on such unseen events is of crucial importance; some techniques on finite automata are described by Dupont and Amengual [128] or by Thollard [122].

3.6.2. Stochastic context-free grammars

The question of inferring probabilistic context-free grammars is going to prove even harder than that of doing the same with finite state machines. Yet the problem has been shown to be of interest in speech recognition [129] or in

computational biology [130] because these grammars can capture the long-term dependencies that can arise for instance in folding, and thus in secondary structure. A first problem that requires study is that of checking if a given grammar is consistent: it is easy to derive a set of rule probabilities for which there is a strictly positive probability that the derivations of the grammar do not halt: larger and larger derivation trees are constructed that (with probability one) never stop expanding. Booth and Thompson [131] give consistency conditions which are proved to hold if the probabilities are estimated from the data [132]. Another "elementary" problem is that of parsing with such a grammar [133]. There are two levels of learning problems:

- If you know the grammar rules you can try to estimate the probabilities that fit best. The usual algorithm in that case is the well-known *inside–outside algorithm* introduced by Baker [134] and studied by Lari and Young [135]. Alternative estimation techniques exist (as for example by Ra and Stockman [136] or Sakakibara et al. [130]).
- You can first learn the rules and then the probabilities. If you have additional information about the data, such as some of its structure, you can turn to adapting a tree-automaton learning algorithm (this is done by Sakakibara [54,90]) if this is not the case, it may be necessary to learn a simplified automaton, corresponding for instance to a local language, and then estimate the probabilities: this path is taken by Rico-Juan et al. [137]; the direct approach of inferring directly the context-free grammars is hard and seems to be attacked only by artificial intelligence techniques, such as genetic algorithms [138].

3.7. Other language representations

Obviously, grammars forming the Chomsky hierarchy correspond to the best-known way of describing languages but there have been other ways, including categorial grammars and patterns.

3.7.1. About pattern languages

Pattern languages have first been studied by Angluin [139]. They are defined by patterns which are sequences of letters, variables and wild-cards. Strings are in the language if they can match the pattern. Different authors have studied variants of pattern languages. Goldman and Kwek [140] provide a good picture of the situation and pointers to the field that has developed independently, with most work done inside the ALT and COLT communities. There is a lot of very specific research in the field, and a survey of these would require a whole article. Just to give the flavor of this research: typical pattern problems have been proved hard (even checking membership is NP-complete); the hardness of the combinatorics can be seen by considering the VC-dimension [141], so simplification of the task has been done by introducing types [142] or considering the case where there is only one pattern [143]. Alternative learning techniques have involved trying to learn pattern languages by means of case-based algorithms [144] or studying the stochastic case [145].

3.7.2. Categorial grammars

Computational linguists have long been interested in working on grammatical models that would not fit into Chomsky's hierarchy. Furthermore, their objective is to find suitable models for syntax and semantics to be interlinked, and provide a logic-based description language. Key ideas relating such models with the questions of language identification can be found in Kanazawa's book [146], and discussion relating this to the way children learn language can be found in papers by a variety of authors, as for instance Ref. [147]. The situation is still unclear, as positive results can only be obtained for special classes of grammars (see for instance Ref. [148]), whereas, here again, the corresponding combinatorial problems (for instance that of finding the smallest consistent grammar) appear to be intractable [149].

4. Applications

Applications based purely on grammatical inference are few, but many use grammatical inference ideas or techniques. As described in the introduction, there is still ample room to find a task where these techniques have done much better than other machine learning or pattern recognition programs. The applications are of interest also because they have often lead to thorough progress in the algorithms of the fields. Having to deal with large alphabets, noisy data, very long strings, scarcity issues and other practical matters has allowed to introduce new problems and to better some of the existing algorithms. We shall point out to these new ideas, when possible.

The early papers in pattern recognition (by Fu and Booth [10] or Miclet [13]) and Sakakibara's article [15] are some of the few places where survey work has taken place.

4.1. Robotics and control systems

Map learning is one potential application for grammatical inference. Dean et al. [150] consider the case where, whilst visiting an environment, the robot may perceive its observation with some possibility of error (but less than 0.5). Related work is by Rivest and Schapire [151]. As another example, Rieger [152] constructs a prefix tree from robot traces. This tree can of course be interpreted as an automaton. Because of sensor imprecision, an NFA, or better, a stochastic automaton describes the model best and can be used for further navigation. In control theory, Luzeaux has also used grammar induction techniques in the field of control theory [153].

4.2. Structural pattern recognition

Structural pattern recognition (for a general description see Bunke and Sanfeliu's book [12]) was an early application of grammatical inference. There are many publications, mainly from the 1970s. Miclet [11,13] and Fu and Booth [10] give details of some of the applications of grammatical inference to textures in images, fingerprints classification, dynamic systems or recognition of pictures of industrial objects. Two representative studies are those by Lucas et al. [154], where image contours are learned, and by Ney [155], for a general survey. Ron et al. [116] also describe a character recognition task by learning stochastic finite automata.

4.3. Computational linguistics

There has always been a lot of interest in relating grammatical inference with natural language. One direction has been taken by Adriaans through shallow grammars [156] (using categorial grammars (see Section 3.7.2): this theoretical work is the backbone of the EMILE prototype [157]. Another direction has been followed by Mohri [158]. Mohri argues that basically DFA, transducers or probabilistic automata are in fact the same object, and that only the output function changes. This leads to the construction of a system to manage these objects [159].

4.4. Speech

Speech technology makes use of language models which in turn require the capacity of parsing uttered sentences with respect to a language model. Typical language models are *n*-grams [160], but HMMs can also be used [161].

Finite automata have been considered as alternative language models for nearly 10 years [162]. Thollard et al. [121] use stochastic automata as an alternative model. Smoothing intervenes in this task also [128]. Amengual et al. [163] offer a nice survey of the use of grammar induction techniques for the task of constructing language models. Ye Wang and Acero [164] obtain reasonable results on the ATIS task [129] by means of learning context-free grammars.

4.5. Automatic translation

Oncina et al. [165] produced algorithm OSTIA to deal with learning subsequential transducers. These are deterministic finite automata with outputs both on the edges and the final states. The inputs for learning are in this case pairs of strings representing the input sentence and the associated output sentence. Improvements of the initial algorithm by Oncina and Varó [166] involve using specific domain knowledge in order to hope to be able to learn partial functions. A version working with translation queries was developed by Vilar [60], and Oncina again [167] uses a data driven approach in order to better OSTIA'S results. Further ideas to better the alignment between text and its translation can be found in Ref. [168], whereas the use of OSTIA in general translation tasks and projects is surveyed by Amengual et al. [169].

4.6. Applications in computational biology

Molecular biology has necessarily the data and the problems for grammatical inference scientists to work on. For the past 10 years this has been so, and even if the most successful methods in the field are not necessarily those using grammars or automata, there are sufficient features in language theory for work to continue. Brazma et al. [170] propose an overview of the situation, with special emphasis on pattern languages. Determining common patterns in DNA, RNA or protein sequences allows to build alignments, discriminate members of families from non-members, and the discovery of new members. Wang et al. [171] apply such techniques for DNA sequence classification tasks: patterns are induced through a learning stage, and used to score in a classification stage. Sakakibara et al. [130] learn stochastic context-free grammars from *t*RNA sequences. The fact that induced grammar is context-free allows to discover and model part of the secondary structure. In the same line of research, secondary structure prediction was detected also by context-free grammars by Abe and Mamitsuka [172]. An experimental result by Salvador and Benedí [173] is that a combination of context-free grammars and bi-grams obtains good results: they use Sakakibara's algorithm [90] on data that can present some very structured regions isolated and others that are not structured.

A lot of more general work concerns the study of hidden Markov models, their relationship with grammars. Lyngsø et al. study all typical distances between distributions in [174] and prove intractability results in [175]. The technique is improved to be able to also compare context-free stochastic grammars [176].

4.7. Other applications

4.7.1. ILP

Inductive logic programming [177] has several links with grammatical inference. It shares some of its objectives (when learning recursive rules) and sometimes its techniques. Boström's system MERLIN parses the data by the background knowledge and uses this information to learn a deterministic finite automaton [178], or a stochastic one [179]. System GIFT by Bernard et al. [180,181] improves on MERLIN, by learning directly tree automata, thus not needing to lose representation capacity by having to linearize the data.

4.7.2. Document management

There are a number of possible applications dealing with documents as data. They either involve constructing dictionaries [182], inferring the grammar generating the tags that have been used [118].

The rise of XML has lead to some new challenges for the field (Fernau points out some of these in [183]). Chiidlovski obtains some preliminary results by using context free grammar learning techniques [184]. For these reasons there has lately been increasing interest in tree automata (see Section 3.3) and tree patterns [185].

4.7.3. Compression

As described in Section 3.2, SEQUITUR [94] learns a grammar from just one string. The obtained grammar can then only generate the string SEQUITUR has learned from. Compression results with this method are comparable to the best compression methods. Moreover, SEQUITUR extracts the structure of the text.

N-grams have allowed to build good compression schemes on text. Using the same sort of ideas, Rico-Juan et al. [137] first learn a *k*-testable tree automaton, and then probabilize this. The obtained model is then used with very good compression rates on tree-like data (XML files, for instance).

4.7.4. Applications to agents

Intelligent agents should have the ability to learn. They are concerned with many learning problems. An original one is that of learning the strategy followed by another agent in a multi-agent world. This can be for negotiation or for collaboration. The interaction can be described as a repeated game, and the agent's strategy can be modelled as a DFA, at least when this strategy is rational: a rational strategy for an agent is just a Moore machine. After each move by its opponent, the agent changes state and acts deterministically following the state it is in. From just one "game" it is possible to learn the opponent's strategy as is shown by Carmel and Markovitch [186] who adapt Angluin's L_* algorithm. The same authors [187] study the related issues and, as in this case the learning is active, how to handle the risk involved with exploration to get hold of new learning data. The algorithms have been tested and have obtained good results in typical applications for the field, based on the iterated prisoner's dilemma.

4.7.5. Applications to time series

It would certainly seem that DFA or PFA could in some way be used as "next value" predictors for the case of discrete time series. Probably because other methods must do better there has been very little done in this context. One noticeable exception is the work by Giles et al. [188] who learn a stochastic finite automaton through neural network techniques, and then use it to predict if the value of a currency is to go up or down on the currency market. Results correspond to a 47% error rate, which in the case of currency prediction seems of interest.

4.7.6. Data mining

There have been recent applications in the emerging fields related with the World Wide Web. We only give 2 examples here of problems on which grammatical inference techniques have been tested:

- Levene and Borges [189] intend to learn user behaviors from their navigation patterns;
- Chidlovskii et al. [190] generate wrappers for meta-search engines semi-automatically through learning a simplified transducer: given a web page returned by a search engine, where only the first query is labelled, the grammar for the entire page is constructed. This can then be used to parse new pages. In Ref. [191] Chidlovskii uses *k*-testable languages for this purpose.

4.7.7. Music

When dealing with music, representation issues acquire a specific importance: both pitch and length have to be encoded, and polyphony is clearly going to be a hard question. Cruz and Vidal [192] use stochastic automata to model music styles (Ragtime, Bach, etc.); the automaton is inferred from a number of pieces of music and can then be used to classify a new piece or even to produce a melody.

5. New trends, open problems

As a conclusion to this survey we would like to emphasize certain research directions that should be of importance in the next few years.

- *Context-free grammars*: As pointed out in Section 3.2, most non-trivial theoretic results concerning the learnability of these models are negative. But the challenge remains open. The main directions that have been followed consist in using some form of simplicity bias (such a line has been followed by Adriaans and Vervoort [157] or Langley and Stromsten [100]), or by exploring the set of all possible rules (by Sakakibara and Kondo [91] or Giordano [89]). On the other hand, extensions of standard algorithms for DFA have been proposed for certain classes of linear deterministic grammars. The advantage of this alternative is that of obtaining characterizable methods, and not heuristics. It is hard to tell which line will win on the long run, but trying to extend the linear deterministic grammars to a non-linear setting would be a good step in the right direction.
- *Dealing with noisy data*: Experimental work suggests that grammar induction algorithms are not robust to noise, whatever its source. Yet for many applications it is necessary to be able to cope with a certain quantity of noise. One usual way out is to deal with stochastic grammars which can deal with noise, but makes the implicit assumption that the distribution (not the just language) is regular. There have been few lines of investigation followed so far to deal with noisy or erroneous examples. Sakakibara cites some theoretical ideas in his review of the field [15]. Sebban and Janodet use distances over ex-

amples to eliminate examples that may be noisy [193]. Practically, only algorithms dealing with stochastic automata or grammars are proving to be robust and have thus been used in applications. Building noise tolerant algorithms is clearly one big issue of the field.

- Combining grammar induction with other techniques or prior knowledge: In most applications the knowledge from which one wants to learn is not just strings. More information is available that cannot be introduced into the learning framework as things stand. Kermorvant and de la Higuera [194] introduced type automata to model external knowledge into grammar induction, and Cano et al. [195] chose to work on forbidden configurations in the target language. A similar point was made by McAllester and Schapire [196]: they proposed "seeding the search with sufficient initial regularities".

References

[1] R.C. Carrasco, J. Oncina (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lecture Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994.

[2] L. Miclet, C. de la Higuera (Eds.), Proceedings of ICGI '96, Lecture Notes in Artificial Intelligence, vol. 1147, Springer, Berlin, Heidelberg, 1996.

[3] V. Honavar, G. Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lecture Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998.

[4] A. de Oliveira (Ed.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 1891, Springer, Berlin, Heidelberg, 2000.

[5] P. Adriaans, H. Fernau, M. van Zaannen (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002.

[6] M. van Zaanen, The grammatical inference homepage, 2003. URL http://eurise.univ-st-etienne.fr/gi/gi.html.

[7] S.S. Kwerk, Colt: Computational learning theory, 1999. URL http://www.learningtheory.org.

[8] T. Zeugmann, Alt series home page, 1999. URL http://www.tcs.mu-luebeck.de/pages/thomas/WALT/waltn.jhtml.

[9] K.S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.

[10] K.S. Fu, T.L. Booth, Grammatical inference: Introduction and survey. Part I and II, IEEE Trans. Syst. Man. Cybernet. 5 (1975) 59–72 and 409–423.

[11] L. Miclet, Structural Methods in Pattern Recognition, Chapman & Hall, New York, 1986.

[12] H. Bunke, A. Sanfeliu (Eds.), Syntactic and Structural Pattern Recognition, Theory and Applications, Series in Computer Science, vol. 7, World Scientific, Singapore, New Jersey, London, Hong Kong, 1990.

[13] L. Miclet, Grammatical inference, in: Syntactic and Structural Pattern Recognition, Theory and Applications, World Scientific, Singapore, 1990, pp. 237–290.

[14] S. Lee, Learning of context-free languages: a survey of the literature, Technical Report TR-12-96, Center for Research in Computing Technology, Harvard University, Cambridge, MA, 1996.

[15] Y. Sakakibara, Recent advances of grammatical inference, Theoret. Comput. Sci. 185 (1997) 15–45.

[16] V. Honavar, C. de la Higuera, Introduction, Mach. Learning J. 44 (1) (2001) 5–7.

[17] C. de la Higuera, Current trends in grammatical inference, in: F.J. Ferri, et al. (Eds.), Advances in Pattern Recognition, Joint IAPR International Workshops SSPR+SPR 2000, Lecture Notes in Computer Science, vol. 1876, Springer, Berlin, 2000, pp. 28–31.

[18] M.H. Harrison, Introduction to Formal Language Theory, Addison-Wesley Publishing Company, Inc., Reading, MA, 1978.

[19] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 1979.

[20] A. Aho, J.D. Ullman, The Theory of Parsing, Translation and Compiling, Parsing, vol. 1, Prentice-Hall, Englewood Cliffs, NJ, 1972.

[21] T.M. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.

[22] B.L. Natarajan, Machine Learning: a Theoretical Approach, Morgan Kauffman Pub., San Mateo, CA, 1991.

[23] M. Kearns, U. Vazirani, An Introduction to Computational Learning Theory, MIT Press, Cambridge, MA, 1994.

[24] R. Gonzalez, M. Thomason, Syntactic Pattern Recognition: an Introduction, Addison-Wesley, Reading, MA, 1978.

[25] B. Trakhtenbrot, Y. Barzdin, Finite Automata: Behavior and Synthesis, North-Holland, Amsterdam, 1973.

[26] K. Lang, B.A. Pearlmutter, The Abbadingo one DFA learning competition, 1997, URL http://abbadingo.cs.unm.edu/.

[27] K.J. Lang, B.A. Pearlmutter, R.A. Price, Results of the Abbadingo one DFA learning competition and a new evidence-driven state merging algorithm, in: Honavar and Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lecture Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998, pp. 1–12.

[28] D. Angluin, Learning regular sets from queries and counterexamples, Inform. and Control 39 (1987) 337–350.

[29] D. Angluin, Queries revisited, in: Abe, et al. (Eds.), Proceedings of ALT 2001, Lecture Notes in Computer Science, vol. 2225, Springer, Berlin, Heidelberg, 2001, pp. 12–31.

[30] C. de la Higuera, Characteristic sets for polynomial grammatical inference, Mach. Learning J. 27 (1997) 125–138.

[31] R. Solomonoff, A formal theory of inductive inference, Inform. and Control 7 (1964) 224–254.

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

[33] E.M. Gold, Complexity of automaton identification from given data, Inform. and Control 37 (1978) 302–320.

[34] D. Angluin, Inductive inference of formal languages from positive data, Inform. and Control 45 (1980) 117–135.

[35] K. Wright, Identification of unions of languages drawn from an identifiable class, in: Proceedings of Workshop on Computational Learning Theory, Morgan Kaufmann Publishers, Los Altos, CA, 1989, pp. 328–333.

[36] S. Lange, T. Zeugmann, Monotonic versus non-monotonic language learning, in: Proceedings of the Second International Workshop on Nonmonotonic and Inductive

Logic, Lecture Notes in Artificial Intelligence, vol. 659, Springer, Berlin, 1993, pp. 254–269.
[37] R.M. Wharton, Approximate language identification, Inform. and Control 26 (1974) 236–255.
[38] D. Angluin, C. Smith, Inductive inference: theory and methods, ACM Comput. Surveys 15 (3) (1983) 237–269.
[39] L. Pitt, Inductive inference, DFA's, and computational complexity, in: Analogical and Inductive Inference, Lecture Notes in Artificial Intelligence, vol. 397, Springer, Berlin, Heidelberg, 1989, pp. 18–44.
[40] D. Angluin, On the complexity of minimum inference of regular sets, Inform. and Control 39 (1978) 337–350.
[41] L. Pitt, M. Warmuth, The minimum consistent DFA problem cannot be approximated within any polynomial, J. Assoc. Comput. Mach. 40 (1) (1993) 95–142.
[42] D. Angluin, Queries and concept learning, Mach. Learning J. 2 (1987) 319–342.
[43] D. Angluin, Negative results for equivalence queries, Mach. Learning J. 5 (1990) 121–150.
[44] R. Gavaldà, On the power of equivalence queries, in: Proceedings of the First European Conference on Computational Learning Theory, The Institute of Mathematics and its Applications Conference Series, New Series, vol. 53, Oxford University Press, Oxford, 1993, pp. 193–203.
[45] L. Pitt, M. Warmuth, Reductions among prediction problems: on the difficulty of predicting automata, in: Third Conference on Structure in Complexity Theory, 1988, pp. 60–69.
[46] M. Warmuth, Towards representation independence in *pac*-learning, in: K.P. Jantke (Ed.), Proceedings of AII'89, Lecture Notes in Artificial Intelligence, vol. 397, Springer, Berlin, 1989, pp. 78–103.
[47] N. Littlestone, Learning quickly when irrelevant attributes abound: a new linear threshold, Mach. Learning J. 2 (1987) 285–318.
[48] S.A. Goldman, M. Kearns, On the complexity of teaching, J. Comput. Syst. Sci. 50 (1) (1995) 20–31.
[49] S.A. Goldman, H. Mathias, Teaching a smarter learner, J. Comput. Syst. Sci. 52 (2) (1996) 255–267.
[50] D. Angluin, A note on the number of queries needed to identify regular languages, Inform. and Control 51 (1981) 76–87.
[51] D. Angluin, Identifying languages from stochastic examples, Technical Report YALEU/DCS/RR-614, Yale University, March 1988.
[52] J.L. Balcazar, J. Diaz, R. Gavaldà, O. Watanabe, The query complexity of learning DFA, New Generat. Comput. 12 (1994) 337–358.
[53] D. Angluin, M. Kharitonov, When won't membership queries help?, in: Proceedings of 24th ACM Symposium on Theory of Computing, ACM Press, New York, 1991, pp. 444–454.
[54] Y. Sakakibara, Learning context-free grammars from structural data in polynomial time, Theoret. Comput. Sci. 76 (1990) 223–242.
[55] A. Birkendorf, A. Boeker, H.U. Simon, Learning deterministic finite automata from smallest counterexamples, SIAM J. Discrete Math. 13 (4) (2000) 465–491.
[56] N.H. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, C. Tamon, Oracles and queries that are sufficient for exact learning, J. Comput. Syst. Sci. 52 (1996) 421–433.
[57] Y. Sakakibara, Inferring parsers of context-free languages from structural examples, Technical Report 81, Fujitsu Limited, International Institute for Advanced Study of Social Information Science, Numazu, Japan, 1987.
[58] T. Yokomori, Learning two-tape automata from queries and counterexamples, Math. Syst. Theory (1996) 259–270.
[59] T. Yokomori, Learning non-deterministic finite automata from queries and counterexamples, Mach. Intell. 13 (1994) 169–189.
[60] J.M. Vilar, Query learning of subsequential transducers, in: L. Miclet, C. de la Higuera (Eds.), Proceedings of ICGI '96, Lecture Notes in Artificial Intelligence, vol. 1147, Springer, Berlin, Heidelberg, 1996, pp. 72–83.
[61] O. Maler, A. Pnueli, On the learnability of infinitary regular sets, in: Proceedings of COLT, Morgan-Kauffman, San Mateo, 1991, pp. 128–136.
[62] L.G. Valiant, A theory of the learnable, Commun. Assoc. Comput. Mach. 27 (11) (1984) 1134–1142.
[63] Y. Ishigami, S. Tani, VC-dimensions of finite automata and commutative finite automata with k letters and n states, Discrete Appl. Math. 74 (1997) 123–134.
[64] P. Bhattacharyya, G. Nagaraja, Learning a class of regular languages in the probably approximately correct learnability framework of Valiant, in: Grammatical Inference: Theory, Applications and Alternatives, First Colloquium on Grammatical Inference, Essex, U.K., IEE Digest no. 1993/092, London, IEE, Essex, UK, 1993.
[65] D. Ron, Y. Singer, N. Tishby, Learning probabilistic automata with variable memory length, in: Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory, ACM Press, New Brunswick, NJ, 1994, pp. 35–46.
[66] M. Kearns, L. Valiant, Cryptographic limitations on learning boolean formulae and finite automata, in: 21st ACM Symposium on Theory of Computing, 1989, pp. 433–444.
[67] M. Li, P. Vitanyi, An Introduction to Kolmogorov Complexity and its Applications, Springer, Berlin, 1993.
[68] M. Li, P. Vitanyi, Learning simple concepts under simple distributions, SIAM J. Comput. 20 (1991) 911–935.
[69] R.J. Parekh, V. Honavar, Learning DFA from simple examples, in: Workshop on Automata Induction, Grammatical Inference, and Language Acquisition, ICML-97, 1997.
[70] P. Dupont, L. Miclet, E. Vidal, What is the search space of the regular inference?, in: R.C. Carrasco, J. Oncina, (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lecture Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 25–37.
[71] K. Lang, Random DFA's can be approximately learned from sparse uniform examples, in: Proceedings of COLT 1992, 1992, pp. 45–52.
[72] C. de la Higuera, J. Oncina, E. Vidal, Identification of DFA: data-dependent versus data-independent algorithm, in: L. Miclet, C. de la Higuera (Eds.), Proceedings of ICGI '96, Lecture Notes in Artificial Intelligence, vol. 1147, Springer, Berlin, Heidelberg, 1996, pp. 313–325.
[73] J.Y. Giordano, Grammatical inference using tabu search, in: L. Miclet, C. de la Higuera (Eds.), Proceedings of ICGI '96, Lecture Notes in Artificial Intelligence, vol. 1147, Springer, Berlin, Heidelberg, 1996, pp. 292–300.
[74] P. Dupont, Regular grammatical inference from positive and negative samples by genetic search: the GIG method, in: R.C. Carrasco, J. Oncina (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lecture Notes in

Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 236–245.
[75] F. Denis, A. Lemay, A. Terlutte, Learning regular languages using RFSA, in: N. Abe, et al. (Eds.), Proceedings of ALT 2001, Lecture Notes in Computer Science, vol. 2225, Springer, Berlin, Heidelberg, 2001, pp. 348–363.
[76] A.L. de Oliveira, J.P.M. Silva, Efficient algorithms for the inference of minimum size DFAs, Mach. Learning J. 44 (1) (2001) 93–119.
[77] K. Lang, Faster algorithms for finding minimal consistent DFAs, Technical Report, NEC Research Institute, 1999.
[78] J.L. Balcazar, J. Diaz, R. Gavaldà, O. Watanabe, An optimal parallel algorithm for learning DFA, in: Proceedings of the Seventh COLT, ACM Press, New York, 1994, pp. 208–217.
[79] A. Biermann, A grammatical inference program for linear languages, in: Fourth Hawaii International Conference on System Sciences, 1971, pp. 121–123.
[80] Y. Takada, Grammatical inference for even linear languages based on control sets, Inform. Process. Lett. 28 (4) (1988) 193–199.
[81] J.M. Sempere, P. García, A characterisation of even linear languages and its application to the learning problem, in: R.C. Carrasco, J. Oncina (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lectures Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 38–44.
[82] E. Mäkinen, A note on the grammatical inference problem for even linear languages, Fundam. Inf. 25 (2) (1996) 175–182.
[83] Y. Takada, A hierarchy of language families learnable by regular language learners, in: R.C. Carrasco, J. Oncina (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lectures Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 16–24.
[84] T. Koshiba, E. Mäkinen, Y. Takada, Learning deterministic even linear languages from positive examples, Theoret. Comput. Sci. 185 (1) (1997) 63–79.
[85] J.M. Sempere, G. Nagaraja, Learning a subclass of linear languages from positive structural information, in: V. Honavar, G. Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lectures Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998, pp. 162–174.
[86] T. Yokomori, Learning context-free languages efficiently: a report on recent results in Japan, in: K.P. Jantke (Ed.), Analogical and Inductive Inference: Proceedings of the International Workshop AII'89, Springer, Berlin, Heidelberg, 1989, pp. 104–123.
[87] C. de la Higuera, J. Oncina, Learning deterministic linear languages, in: J. Kivinen, R.H. Sloan (Eds.), Proceedings of COLT 2002, Lectures Notes in Artificial Intelligence, vol. 2375, Springer, Berlin, Heidelberg, 2002, pp. 185–200.
[88] C. de la Higuera, J. Oncina, On sufficient conditions to identify in the limit classes of grammars from polynomial time and data, in: P. Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lectures Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 134–148.
[89] J.Y. Giordano, Inference of context-free grammars by enumeration: structural containment as an ordering bias, in: R.C. Carrasco, J. Oncina (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lectures Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 212–221.
[90] Y. Sakakibara, Efficient learning of context-free grammars from positive structural examples, Inform. and Comput. 97 (1992) 23–60.
[91] Y. Sakakibara, M. Kondo, Ga-based learning of context-free grammars using tabular representations, in: Proceedings of 16th International Conference on Machine Learning (ICML-99), 1999, pp. 354–360.
[92] Y. Sakakibara, H. Muramatsu, Learning context-free grammars from partially structured examples, in: A. de Oliveira (Ed.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lectures Notes in Artificial Intelligence, vol. 1891, Springer, Berlin, Heidelberg, 2000, pp. 229–240.
[93] I. Ishizaka, Learning simple deterministic languages, in: Proceedings of COLT 89, 1989.
[94] C. Nevill-Manning, I. Witten, Identifying hierarchical structure in sequences: a linear-time algorithm, J. Artif. Intell. Res. 7 (1997) 67–82.
[95] P. García, J. Oncina, Inference of recognizable tree sets, Technical Report DSIC-II/47/93, Departamento de Lenguajes y Sistemas Informáticos, Universidad Politécnica de Valencia, Spain, 1993.
[96] R.C. Carrasco, J. Oncina, J. Calera-Rubio, Stochastic inference of regular tree languages, Mach. Learning J. 44 (1) (2001) 185–197.
[97] T. Knuutila, M. Steinby, Inference of tree languages from a finite sample: an algebraic approach, Theoret. Comput. Sci. 129 (1994) 337–367.
[98] H. Fernau, Learning tree languages from text, in: J. Kivinen, R.H. Sloan (Eds.), Proceedings of COLT 2002, Lectures Notes in Artificial Intelligence, vol. 2375, Springer, Berlin, Heidelberg, 2002, pp. 153–168.
[99] K. Vanlehn, W. Ball, A version space approach to learning context-free grammars, Mach. Learning J. 2 (1987) 39–74.
[100] P. Langley, S. Stromsten, Learning context-free grammars with a simplicity bias, in: Proceedings of ECML 2000, 11th European Conference on Machine Learning, Lecture Notes in Computer Science, vol. 1810, Springer, Berlin, 2000, pp. 220–228.
[101] T. Yokomori, S. Kobayashi, Inductive learning of regular sets from examples: a rough set approach, in: Proceedings of International Workshop on Rough Sets and Soft Computing, 1994.
[102] T. Knuutila, Inductive inference from positive data: from heuristic to characterising methods, in: L. Miclet, C. de la Higuera (Eds.), Proceedings of ICGI '96, Lectures Notes in Artificial Intelligence, vol. 1147, Springer, Berlin, Heidelberg, 1996, pp. 22–47.
[103] D. Angluin, Inference of reversible languages, J. Assoc. Comput. Mach. 29 (3) (1982) 741–765.
[104] P. García, E. Vidal, Inference of K-testable languages in the strict sense and applications to syntactic pattern recognition, IEEE Trans. Pattern Anal. Mach. Intell. 12 (9) (1990) 920–925.
[105] F. Denis, A. Lemay, A. Terlutte, Some classes of regular languages identifiable in the limit from positive data, in: Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lectures Notes

on Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 63–76.
[106] H. Fernau, Identification of function distinguishable languages, in: H. Arimura, S. Jain, A. Sharma (Eds.), Proceedings of the 11th International Conference on Algorithmic Learning Theory (ALT 2000), Lecture Notes in Computer Science, vol. 1968, Springer, Berlin, Heidelberg, 2000, pp. 116–130.
[107] M.O. Rabin, Probabilistic automata, Inform. and Control 6 (1966) 230–245.
[108] A. Paz, Introduction to Probabilistic Automata, Academic Press, New York, 1971.
[109] G.D. Forney, The Viterbi algorithm, in: IEEE Proceedings, vol. 3, 1973, pp. 268–278.
[110] F. Casacuberta, C. de la Higuera, Computational complexity of problems on probabilistic grammars and transducers, in: A. de Oliveira (Ed.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lectures Notes in Artificial Intelligence, vol. 1891, Springer, Berlin, Heidelberg, 2000, pp. 15–24.
[111] L.E. Baum, An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, Inequalities 3 (1972) 1–8.
[112] F. Casacuberta, Some relations among stochastic finite state networks used in automatic speech recognition, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 691–695.
[113] N. Abe, M. Warmuth, On the computational complexity of approximating distributions by probabilistic automata, Mach. Learning J. 9 (1992) 205–260.
[114] M.J. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R.E. Schapire, L. Sellie, On the learnability of discrete distributions, in: Proceedings of the 25th Annual ACM Symposium on Theory of Computing, 1994, pp. 273–282.
[115] R.C. Carrasco, J. Oncina, Learning stochastic regular grammars by means of a state merging method, in: R.C. Carrasco, J. Oncina, ICGI'94 (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lecture Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 139–150.
[116] D. Ron, Y. Singer, N. Tishby, On the learnability and usage of acyclic probabilistic finite automata, in: Proceedings of COLT 1995, 1995, pp. 31–40.
[117] A. Stolcke, S. Omohundro, Inducing probabilistic grammars by bayesian model merging, in: R.C. Carrasco, J. Oncina, (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lecture Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 106–118.
[118] M. Young-Lai, F.W. Tompa, Stochastic grammatical inference of text database structure, Mach. Learning J. 40 (2) (2000) 111–137.
[119] T. Goan, N. Benson, O. Etzioni, A grammar inference algorithm for the world wide web, in: Proceedings of AAAI Spring Symposium on Machine Learning in Information Access, AAAI Press, Stanford, CA, 1996.
[120] R.C. Carrasco, Accurate computation of the relative entropy between stochastic regular grammars, RAIRO, Theoret. Informatics Appl. 31 (5) (1997) 437–444.
[121] F. Thollard, P. Dupont, C. de la Higuera, Probabilistic DFA inference using kullback-leibler divergence and minimality, in: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 975–982.
[122] F. Thollard, Improving probabilistic grammatical inference core algorithms with post-processing techniques, in: Eighth International Conference on Machine Learning, Morgan Kauffman, Los Altos, CA, 2001, pp. 561–568.
[123] R.C. Carrasco, M. Forcada, L. Santamaria, Inferring stochastic regular grammars with recurrent neural networks, in: L. Miclet, C. de la Higuera (Eds.), Proceedings of ICGI '96, Lecture Notes in Artificial Intelligence, vol. 1147, Springer, Berlin, Heidelberg, 1996, pp. 274–281.
[124] R.C. Carrasco, J. Oncina, Learning deterministic regular grammars from stochastic samples in polynomial time, RAIRO, Theoret. Informatics Appl. 33 (1) (1999) 1–20.
[125] C. de la Higuera, F. Thollard, Identification in the limit with probability one of stochastic deterministic finite automata, in: A. de Oliveira (Ed.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 1891, Springer, Berlin, Heidelberg, 2000, pp. 15–24.
[126] C. de la Higuera, Learning stochastic finite automata from experts, in: V. Honavar, G. Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lecture Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998, pp. 79–89.
[127] I.H. Witten, T.C. Bell, The zero frequency problem: Estimating the probabilities of novel events in adaptive test compression, IEEE Trans. IT-37 (4) (1991) 1085–1094.
[128] P. Dupont, J.-C. Amengual, Smoothing probabilistic automata: an error-correcting approach, in: A. de Oliveira (Ed.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 1891, Springer, Berlin, Heidelberg, 2000, pp. 51–62.
[129] Y. Wang, A. Acero, Evaluation of spoken language grammar learning in the ATIS domain, in: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 2002.
[130] Y. Sakakibara, M. Brown, R. Hughley, I. Mian, K. Sjolander, R. Underwood, D. Haussler, Stochastic context-free grammars for tRNA modeling, Nucleic Acids Res. 22 (1994) 5112–5120.
[131] T.L. Booth, R.A. Thompson, Applying probability measures to abstract languages, IEEE Trans. Comput. C-22 (5) (1973) 442–450.
[132] J.A. Sánchez, J.M. Benedí, Consistency of stochastic context-free grammars from probabilistic estimation based on growth transformation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (9) (1997) 1052–1055.
[133] A. Stolcke, An efficient probabilistic context-free parsing algorithm that computes prefix probabilities, Comput. Linguist. 21 (2) (1995) 165–201.
[134] J.K. Baker, Trainable grammars for speech recognition, in: D.H. Klatt, J.J. Wolf (Eds.), Speech Communication Papers for the 97th Meeting of the Acoustical Society of America, 1979, pp. 547–550.
[135] K. Lari, S.J. Young, The estimation of stochastic context free grammars using the inside–outside algorithm, Comput. Speech Lang. 4 (1990) 35–56.
[136] D.-Y. Ra, G.C. Stockman, A new one pass algorithm for estimating stochastic context-free grammars, Inf. Process. Lett. 72 (1999) 37–45.

[137] J.R. Rico-Juan, J. Calera-Rubio, R.C. Carrasco, Stochastic *k*-testable tree languages and applications, in: P. Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 199–212.

[138] T. Kammeyer, R.K. Belew, Stochastic context-free grammar induction with a genetic algorithm using local search, in: R.K. Belew, M. Vose (Eds.), Foundations of Genetic Algorithms IV, Morgan Kaufmann, University of San Diego, CA, USA, 1996.

[139] D. Angluin, Finding patterns common to a set of strings, in: Conference Record of the 11th Annual ACM Symposium on Theory of Computing, ACM Press, New York, NY, USA, 1979, pp. 130–141.

[140] S.A. Goldman, S.S. Kwek, On learning unions of pattern languages and tree patterns, in: O. Watanabe, T. Yokomori (Eds.), Proceedings of the 10th International Conference on Algorithmic Learning Theory (ALT 1999), Lecture Notes in Computer Science, vol. 1720, Springer, Berlin, 1999, pp. 347–363.

[141] A. Mitchell, T. Scheffer, A. Sharma, F. Stephan, The VC-dimension of subclasses of pattern languages, in: O. Watanabe, T. Yokomori (Eds.), Proceedings of the 10th International Conference on Algorithmic Learning Theory (ALT 1999), Lecture Notes in Computer Science, vol. 1720, Springer, Berlin, 1999, pp. 93–105.

[142] T. Koshiba, Typed pattern languages and their learnability, Proceedings of Euro COLT '95, Lecture Notes in Artificial Intelligence, vol. 904, Springer, Berlin, 1995, pp. 367–379.

[143] T. Erlebach, P. Rossmanith, H. Stadtherr, A. Steger, T. Zeugmann, Learning one-variable pattern languages very efficiently on average, in parallel, and by asking queries, in: M. Li, A. Maruoka (Eds.), Proceedings of ALT '97, Lecture Notes in Computer Science, vol. 1316, Springer, Berlin, Heidelberg, 1997, pp. 260–276.

[144] K.P. Jantke, S. Lange, Case-based representation and learning of pattern languages, Theoret. Comput. Sci. 137 (1) (1995) 25–51.

[145] P. Rossmanith, T. Zeugmann, Stochastic finite learning of the pattern languages, Mach. Learning J. 44 (1) (2001) 67–91.

[146] M. Kanazawa, Learnable Classes of Categorial Grammars, CSLI Publications, Stanford, CA, 1998.

[147] I. Tellier, Meaning helps learning syntax, in: V. Honavar, G. Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lecture Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998, pp. 25–36.

[148] A. Foret, Y.L. Nir, On limit points for some variants of rigid Lambek grammars, in: P. Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 106–119.

[149] C.C Florêncio, Consistent identification in the limit of rigid grammars from strings is NP-hard, in: P. Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 49–62.

[150] T. Dean, K. Basye, L. Kaelbling, E. Kokkevis, O. Maron, D. Angluin, S. Engelson, Inferring finite automata with stochastic output functions and an application to map learning, in: W. Swartout (Ed.), Proceedings of the 10th National Conference on Artificial Intelligence, MIT Press, San Jose, CA, 1992, pp. 208–214.

[151] R.L. Rivest, R.E. Schapire, Inference of finite automata using homing sequences, Inform. and Comput. 103 (1993) 299–347.

[152] A. Rieger, Inferring probabilistic automata from sensor data for robot navigation, in: M. Kaiser (Ed.), Proceedings of the MLnet Familiarization Workshop and Third European Workshop on Learning Robots, 1995, pp. 65–74.

[153] D. Luzeaux, Machine learning applied to the control of complex systems, in: Proceedings of the Eighth International Conference on Artificial Intelligence and Expert Systems Applications, Paris, France, 1996.

[154] S. Lucas, E. Vidal, A. Amari, S. Hanlon, J.C. Amengual, A comparison of syntactic and statistical techniques for off-line OCR, in: R.C. Carrasco, J. Oncina (Eds.), Grammatical Inference and Applications, Proceedings of ICGI '94, Lecture Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 168–179.

[155] H. Ney, Stochastic grammars and pattern recognition, in: P. Laface, R.D. Mori (Eds.), Proceedings of the NATO Advanced Study Institute, Springer, Berlin, 1992, pp. 313–344.

[156] P. Adriaans, Language learning from a categorical perspective, Ph.D. Thesis, Universiteit van Amsterdam, 1992.

[157] P. Adriaans, M. Vervoort, The EMILE 4.1 grammar induction toolbox, in: P. Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 293–295.

[158] M. Mohri, Finite-state transducers in language and speech processing, Comput. Linguist. 23 (3) (1997) 269–311.

[159] M. Mohri, F.C.N. Pereira, M. Riley, The design principles of a weighted finite-state transducer library, Theoret. Comput. Sci. 231 (1) (2000) 17–32.

[160] F. Jelinek, Statistical Methods for Speech Recognition, The MIT Press, Cambridge, MA, 1998.

[161] N. Morgan, H. Bourlard, Continuous speech recognition: an introduction to the hybrid HMM/connectionnist approach, IEEE Signal Process. Mag. 12.

[162] P. García, E. Segarra, E. Vidal, I. Galiano, On the use of the morphic generator grammatical inference (mggi) methodology in automatic speech recognition, Int. J. Pattern Recogn. Artif. Intell. 4 (1994) 667–685.

[163] J.-C. Amengual, A. Sanchis, E. Vidal, J.-M. Benedí, Language simplification through error-correcting and grammatical inference techniques, Mach. Learning J. 44 (1) (2001) 143–159.

[164] Y. Wang, A. Acero, Grammar learning for spoken language understanding, in: Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, Madonna di Campiglio, 2001.

[165] J. Oncina, P. García, E. Vidal, Learning subsequential transducers for pattern recognition interpretation tasks, IEEE Trans. Pattern Anal. Mach. Intell. 15 (5) (1993) 448–458.

[166] J. Oncina, M.A. Varó, Using domain information during the learning of a subsequential transducer, in: L. Miclet, C. de la Higuera (Eds.), Proceedings of ICGI '96, Lecture Notes in Artificial Intelligence, vol. 1147, Springer, Berlin, Heidelberg, 1996, pp. 301–312.

[167] J. Oncina, The data driven approach applied to the OSTIA algorithm, in: V. Honavar, G. Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lecture Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998, pp. 50–56.

[168] J.M. Vilar, Improve the learning of subsequential transducers by using alignments and dictionaries, in: A. de Oliveira (Ed.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 1891, Springer, Berlin, Heidelberg, 2000, pp. 298–312.

[169] J.C. Amengual, J.M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, V.M. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, J.M. Vilar, The EuTrans-I speech translation system, Mach. Translation 15 (1) (2001) 75–103.

[170] A. Brazma, I. Jonassen, J. Vilo, E. Ukkonen, Pattern discovery in biosequences, in: V. Honavar, G. Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lecture Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998, pp. 257–270.

[171] J.T.-L. Wang, S. Rozen, B.A. Shapiro, D. Shasha, Z. Wang, M. Yin, New techniques for DNA sequence classification, J. Comput. Biol. 6 (2) (1999) 209–218.

[172] N. Abe, H. Mamitsuka, Predicting protein secondary structure using stochastic tree grammars, Mach. Learning J. 29 (1997) 275–301.

[173] I. Salvador, J.-M. Benedí, RNA modeling by combining stochastic context-free grammars and *n*-gram models, Int. J. Pattern Recogn. Artif. Intell. 16 (3) (2002) 309–316.

[174] R.B. Lyngsø, C.N.S. Pedersen, H. Nielsen, Metrics and similarity measures for hidden Markov models, in: Proceedings of ISMB'99, 1999, pp. 178–186.

[175] R.B. Lyngsø, C.N.S. Pedersen, Complexity of comparing hidden Markov models, in: Proceedings of ISAAC '01, Lecture Notes in Computer Science, vol. 2223, Springer, Berlin, Heidelberg, 2001, pp. 416–428.

[176] A. Jagota, R. B. Lyngsø, C.N.S. Pedersen, Comparing a hidden Markov model and a stochastic context-free grammar, in: Proceedings of WABI '01, Lecture Notes in Computer Science, vol. 2149, Springer, Berlin, Heidelberg, 2001, pp. 69–74.

[177] S. Muggleton, Inductive Logic Programming, in: The MIT Encyclopedia of the Cognitive Sciences (MITECS), MIT Press, Cambridge, MA, 1999.

[178] H. Boström, Theory-guided induction of logic programs by inference of regular languages, in: 13th International Conference on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1996.

[179] H. Boström, Predicate invention and learning from positive examples only, in: C. Nédellec, C. Rouveirol (Eds.), 10th European Conference on Machine Learning, Lecture Notes in Artificial Intelligence, vol. 1398, Springer, Berlin, 1998, pp. 226–237.

[180] M. Bernard, C. de la Higuera, Apprentissage de programmes logiques par inférence grammaticale, Rev. d'Intell. Artif. 14 (3) (2001) 375–396.

[181] A. Habrard, M. Bernard, F. Jacquenet, Generalized stochastic tree automata for multi-relational data mining, in: P. Adriaans, et al., (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 120–133.

[182] H. Ahonen, H. Mannila, E. Nikunen, Forming grammars for structured documents: An application of grammatical inference, in: R.C. Carrasco, J. Oncina (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '94, Lecture Notes in Artificial Intelligence, vol. 862, Springer, Berlin, Heidelberg, 1994, pp. 153–167.

[183] H. Fernau, Learning XML grammars, in: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition MLDM'01, Lecture Notes in Computer Science, vol. 2123, Springer, Berlin, 2001, pp. 73–87.

[184] B. Chidlovskii, Schema extraction from xml: A grammatical inference approach, in: M. Lenzerini, D. Nardi, W. Nutt, D. Suciu (Eds.), Proceedings of the Eighth International Workshop on Knowledge Representation meets Databases (KRDB 2001), vol. 45 CEUR Workshop Proceedings, 2001.

[185] H. Arimura, H. Sakamoto, S. Arikawa, Efficient learning of semi-structured data from queries, in: N. Abe, et al. (Eds.), Proceedings of ALT 2001, Lecture Notes in Computer Science, vol. 2225, Springer, Berlin, Heidelberg, 2001, pp. 315–331.

[186] D. Carmel, S. Markovitch, Model-based learning of interaction strategies in multi-agent systems, J. Exp. Theoret. Artif. Intell. 10 (3) (1998) 309–332.

[187] D. Carmel, S. Markovitch, Exploration strategies for model-based learning in multiagent systems, Auton. Agents Multi-agent Syst. 2 (2) (1999) 141–172.

[188] C.L. Giles, S. Lawrence, A. Tsoi, Noisy time series prediction using recurrent neural networks and grammatical inference, Mach. Learning J. 44 (1) (2001) 161–183.

[189] J. Borges, M. Levene, Data mining of user navigation patterns, in: B. Masand, M. Spiliopoulou (Eds.), Web Usage Mining and User Profiling, Lecture Notes in Computer Science, vol. 1836, Springer, Berlin, 2000, pp. 92–111.

[190] B. Chidlovskii, J. Ragetli, M. de Rijke, Wrapper generation via grammar induction, in: Machine Learning: ECML 2000, 11th European Conference on Machine Learning, vol. 1810, Springer, Berlin, 2000, pp. 96–108.

[191] B. Chidlovskii, Wrapper generation by *k*-reversible grammar induction, in: Proceedings of the Workshop on Machine Learning and Information Extraction, 2000.

[192] P. Cruz, E. Vidal, Learning regular grammars to model musical style: Comparing different coding schemes, in: V. Honavar, G. Slutski (Eds.), Grammatical Inference, Proceedings of ICGI '98, Lecture Notes in Artificial Intelligence, vol. 1433, Springer, Berlin, Heidelberg, 1998, pp. 211–222.

[193] M. Sebban, J.-C. Janodet, On state merging in grammatical inference: a statistical approach for dealing with noisy data, in: Proceedings of ICML, 2003.

[194] C. Kermorvant, C. de la Higuera, Learning languages with help, in: P. Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 161–173.

[195] A. Cano, J. Ruiz, P. García, Inferring subclasses of regular languages faster using RPNI and forbidden configurations, in: P. Adriaans, et al. (Eds.), Grammatical Inference: Algorithms and Applications, Proceedings of ICGI '00, Lecture Notes in Artificial Intelligence, vol. 2484, Springer, Berlin, Heidelberg, 2002, pp. 28–36.

[196] D. McAllester, R. Schapire, Learning theory and language modeling, Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann, Los Altos, CA, 2002.

Further Reading

[197] N. Abe, R. Khardon, T. Zeugmann (Eds.), Proceedings of ALT 2001, Lecture Notes in Computer Science, vol. 2225, Springer, Berlin, Heidelberg, 2001.

[198] J. Kivinen, R.H. Sloan (Eds.), Proceedings of COLT 2002, Lecture Notes in Artificial Intelligence, vol. 2375, Springer, Berlin, Heidelberg, 2002.

[199] O. Watanabe, T. Yokomori (Eds.), Proceedings of the 10th International Conference on Algorithmic Learning Theory (ALT 1999), Lecture Notes in Computer Science, vol. 1720, Springer, Berlin, 1999.