Phase Transitions within Grammatical Inference

Nicolas Pernot, Antoine Cornuéjols & Michèle Sebag
Laboratoire de Recherche en Informatique, CNRS UMR 8623
Bât.490, Université de Paris-Sud, Orsay
91405 Orsay Cedex (France)
antoine,sebag@lri.fr

Abstract

It is now well-known that the feasibility of inductive learning is ruled by statistical properties linking the empirical risk minimization principle and the "capacity" of the hypothesis space. The discovery, a few years ago, of a phase transition phenomenon in inductive logic programming proves that other fundamental characteristics of the learning problems may similarly affect the very possibility of learning under very general conditions.

Our work examines the case of grammatical inference. We show that while there is no phase transition when considering the whole hypothesis space, there is a much more severe "gap" phenomenon affecting the effective search space of standard grammatical induction algorithms for deterministic finite automata (DFA). Focusing on the search heuristics of the RPNI and RED-BLUE algorithms, we show that they overcome this problem to some extent, but that they are subject to overgeneralization. The paper last suggests some directions for new generalization operators, suited to this Phase Transition phenomenon.

1 Introduction

It is now well-known that the feasibility of inductive learning is ruled by statistical properties linking the empirical risk minimization principle and the "capacity" of the hypothesis space [Vapnik, 1995]. While this powerful framework leads to a much deeper understanding of machine learning and to many theoretical and applicative breakthroughs, it basically involves only statistical information on the learning search space, e.g. the so-called VC-dimension. The dynamics of the learning search is not considered.

Independently, a new combinatoric paradigm has been studied in the Constraint Satisfaction community since the early 90s, motivated by computational complexity concerns [Cheeseman *et al.*, 1991]: where are the really hard problems? Indeed, the worst case complexity analysis poorly accounts for the fact that, despite an exponential worst-case complexity, empirically, the complexity is low for most CSP instances. These remarks led to developing the so-called *phase transition framework* (PT) [Hogg *et al.*, 1996], which considers the satisfiability and the resolution complexity of CSP instances as random variables depending on order parameters of the problem instance (e.g. constraint density and tightness). This framework unveiled an interesting structure of the CSP landscape. Specifically, the landscape is divided into three regions: the `YES` region, corresponding to underconstrained problems, where the satisfiability probability is close to 1 and the average complexity is low; the `NO` region, corresponding to overconstrained problems, where the satisfiability probability is close to 0 and the average complexity is low too; last, a narrow region separating the `YES` and `NO` regions, referred to as *phase transition region*, where the satisfiability probability abruptly drops from 1 to 0 and which concentrates on average the computationally heaviest CSP instances.

The phase transition paradigm has been transported to relational machine learning and inductive logic programming (ILP) by [Giordana and Saitta, 2000], motivated by the fact that the covering test most used in ILP [Muggleton and Raedt, 1994] is equivalent to a CSP. As anticipated, a phase transition phenomenon appears in the framework of ILP: a wide `YES` (respectively `NO`) region includes all hypotheses which cover (resp. reject) all examples, and the hypotheses that can discriminate the examples lie in the narrow PT, where the average computational complexity of the covering test reaches its maximum.

Besides computational complexity, the PT phenomenon has far-reaching effects on the success of relational learning [Botta *et al.*, 2003]. For instance, a wide *Failure Region* is observed: for all target concepts/training sets in this region, no learning algorithms among the prominent ILP ones could find hypotheses better than random guessing [Botta *et al.*, 2003].

These negative results lead to a better understanding of the intrinsic limits of the existing ILP algorithms and search biases. Formally, consider a greedy specialization (top-down) search strategy: starting its exploration in the `YES` region, the system is almost bound to make random specialization choices, for all hypotheses in this region cover every example on average. The `YES` region constitutes a rugged plateau from a search perspective, and there is little chance that the algorithm ends in the right part of the PT region, where good hypotheses lie. A similar reasoning goes for algorithms that follow a greedy generalization strategy.

The phase transition paradigm thus provides another perspective on the pitfalls facing machine learning, focusing on the combinatoric search aspects while statistical learning focuses on the statistical aspects.

The main question studied in this paper is whether the PT phenomenon is limited to relational learning, or threatens the feasibility and tractability of other learning settings as well.

A learning setting with intermediate complexity between full relational learning and propositional learning is thus considered, that of grammatical inference (GI) [Pitt, 1989; Sakakibara, 1997]. Only the case of Finite-State Automata (FSA, section 2), will be considered through the paper. Specifically, the phase transition phenomenon will be investigated with respect to three distributions on the FSA space, incorporating gradually increasing knowledge on the syntactical and search biases of GI algorithms.

The first distribution incorporates no information on the algorithm and considers the whole space of FSA. Using a set of order parameters, the average coverage of automata is studied analytically and empirically.

The second one reflects the bias introduced by the generalization relations defined on the FSA space and exploited by GI algorithms. The vast majority of these algorithms first construct a least general generalization of the positive examples, or Prefix Tree Acceptor (PTA), and restrict the search to the generalizations of the PTA, or *generalization cone*[1].

The third distribution takes into account the heuristics used by GI algorithms, guiding the search trajectory in the generalization cone. Due to space limitations, the study is restricted to two prominent GI algorithms, namely RPNI [Oncina and Garcia, 1992] and RED-BLUE [Lang *et al.*, 1998].

This paper is organized as follows. Section 2 briefly introduces the domain of Grammatical Inference, the principles of the inference algorithms and defines the order parameters used in the rest of the paper. Section 3 investigates the existence and potential implications of phase transition phenomenons in the whole FSA space (section 3.1) and in the generalization cone (section 3.2). Section 4 focuses on the actual landscape explored by GI algorithms, specifically considering the search trajectories of RPNI and RED-BLUE. Section 5 discusses the scope of the presented study and lays out some perspectives for future research.

2 Grammatical inference

After introducing general notations and definitions, this section briefly discusses the state of the art and introduces the order parameters used in the rest of the paper.

2.1 Notations and definitions

Grammatical inference is concerned with inferring grammars from positive (and possibly negative) examples. Only regular grammars are considered in this paper. They form the bottom class of the hierarchy of formal grammars as defined by Chomsky, yet are sufficiently rich to express many interesting sequential structures. Their identification in the limit from positive examples only is known to be impossible, while it is feasible with a complete set of examples [Gold, 1967].

It is known that any regular language can be produced by a finite-state automaton (FSA), and that any FSA generates a regular language. In the remaining of the paper, we will mostly use the terminology of finite-state automata. A FSA is a 5-tuple $A = \langle \Sigma, \mathcal{Q}, \mathcal{Q}_0, F, \delta \rangle$ where Σ is a finite alphabet, $\mathcal{Q}$ is a finite set of states, $\mathcal{Q}_0 \subseteq \mathcal{Q}$ is the set of initial states, $F \subseteq \mathcal{Q}$ is the set of final states, δ is the transition function defined from $\mathcal{Q} \times \Sigma$ to $2^{\mathcal{Q}}$.

A positive example of a FSA is a string on Σ, produced by following any path in the graph linking one initial state q_0 to any accepting state.

A finite state-automaton (FSA) is deterministic (DFA) if $\mathcal{Q}_0$ contains exactly one element q_0 and if $\forall q \in \mathcal{Q}, \forall x \in \Sigma, Card(\delta(q, x)) \leq 1$. Otherwise it is non-deterministic (NFA). Every NFA can be translated into an equivalent DFA, but at the price of being possibly exponentially more complex in terms of number of states. Given any FSA A', there exists a minimum state DFA (also called *canonical DFA*) A such that $L(A) = L(A')$ (where $L(A)$ denotes the set of strings accepted by A). Without loss of generality, it can be assumed that the target automaton being learned is a canonical DFA.

A set S^+ is said to be *structurally complete* with respect to a DFA A if S^+ covers each transition of A and uses every element of the set of final states of A as an accepting state. Clearly, $L(PTA(S^+)) = S^+$.

Given a FSA A and a partition π on the set of states $\mathcal{Q}$ of A, the *quotient automaton* is obtained by merging the states of A that belong to the same block in partition π (see [Dupont *et al.*, 1994] for more details). Note that a quotient automaton of a DFA might be a NFA and vice versa. The set of all quotient automata obtained by systematically merging the states of a DFA A represents a lattice of FSAs. This lattice is ordered by the *grammar cover* relation $\preceq$. The transitive closure of $\preceq$ is denoted by $\ll$. We say that $A_{\pi_i} \ll A_{\pi_j}$ *iff* $L(A_{\pi_i}) \subseteq L(A_{\pi_j})$. Given a canonical DFA A and a set S^+ that is structurally complete with respect to A, the lattice derived from $PTA(S^+)$ is guaranteed to contain A.

From these assumptions, follows the paradigmatic approach of most grammatical inference algorithms (see, e.g., [Coste, 1999; Dupont *et al.*, 1994; Pitt, 1989; Sakakibara, 1997]), which equates generalization with state merging operations starting from the PTA.

2.2 Learning biases in grammatical inference

The core task of GI algorithms is thus to select iteratively a pair of states to be merged. The differences among algorithms is related to the choice of: (i) the search criterion (which merge is the best one); (ii) the search strategy (how is the search space explored); and (iii) the stopping criterion.

We shall consider here the setting of learning FSAs from positive and negative examples, and describe the algorithms studied in section 3. In this setting, the stopping criterion is determined from the negative examples: generalization proceeds as long as the candidate solutions remain correct, not

[1] More precisely, the Prefix Tree Acceptor is obtained by merging the states that share the same prefix in the Maximal Canonical Automaton (MCA), which represents the whole positive learning set as an automaton. A PTA is therefore a DFA with a tree-like structure.

covering any negative example[2]

The RPNI algorithm [Oncina and Garcia, 1992] uses a depth first search strategy with some backtracking ability, favoring the pair of states which is closest to the start state, such that their generalization (FSA obtained by merging the two states and subsequently applying the determinisation operator) does not cover any negative example.

The RED-BLUE algorithm (also known as BLUE-FRINGE) [Lang *et al.*, 1998] uses a beam search from a candidate list, selecting the pair of states after the *Evidence-Driven* State Merging (EDSM) criterion, i.e. such that their generalization involves a minimal number of final states. RED-BLUE thus also performs a search with limited backtracking, based on a more complex criterion and a wider search width than RPNI.

2.3 Order Parameters

Following the methodology introduced in [Giordana and Saitta, 2000], the PT phenomenon is investigated along so-called *order parameters* chosen in accordance with the parameters used in the *Abbaddingo* challenge [Lang *et al.*, 1998]:

- The number Q of states in the DFA.
- The number B of output edges on each state.
- The number L of letters on each edge.
- The fraction a of accepting states, taken in [0,1].
- The size $|\Sigma|$ of the alphabet considered.
- The length ℓ of the test examples. Also the maximal length ℓ of the learning examples in S^+ (as explained below).

The study first focuses on the intrinsic properties of the search space (section 3) using a a *random sampling* strategy (all ℓ letters in the string being independently and uniformly drawn in Σ). In section 4, we examine the capacity of the studied learning algorithms to approximate a target automaton, based on *positive sampling*, where each training string is produced by following a path in the graph, randomly selecting an output edge in each step[3].

3 Phase Transitions: the FSA space and the generalization cone

This section investigates the percentage of coverage of deterministic and non-deterministic Finite-State Automata, either uniformly selected (section 3.1), or selected in the subspace actually investigated by grammatical inference algorithms, that is, the generalization cone (section 3.2).

3.1 Phase Transition in the whole FSA space

The sampling mechanism on the whole deterministic FSA space (DFA) is defined as follows. Given the order parameter values $(Q, B, L, a, |\Sigma|)$:

- for every state q, (i) B output edges (q, q') are created, where q' is uniformly selected with no replacement among the Q states; (ii) $L \times B$ distinct letters are uniformly selected in Σ; and (iii) these letters are evenly distributed among the B edges above.
- every state q is turned into an accepting state with probability a.

The sampling mechanism for NFA differs from the above in a single respect: two edges with same origin state are not required to carry distinct letters.

For each setting of the order parameters, 100 independent problem instances are constructed. For each considered FSA (the sampling mechanisms are detailed below), the coverage rate is measured as the percentage of covered examples among 1,000 examples (strings of length ℓ) uniformly sampled.

Fig. 1 shows the average coverage in the (a, B) plane, for $|\Sigma| = 2$, $L = 1$ and $\ell = 10$, where the accepting rate a varies in $[0, 1]$ and the branching factor B varies in {1,2}. Each point reports the average coverage of a sample string s by a FSA (averaged over 100 FSA drawn with accepting rate a and branching factor B, tested on $1,000$ strings s of length ℓ).

These empirical results are analytically explained from the simple equations below, giving the probability that a string of length ℓ be accepted by a FSA defined on an alphabet of size $|\Sigma|$, with a branching factor B and L letters on each edge, in the DFA and NFA cases (the number of states Q is irrelevant here).

$$P(\text{accept}) = \begin{cases} a \cdot (\frac{B \cdot L}{|\Sigma|})^\ell & \text{for a DFA} \\ a \cdot [1 - (1 - \frac{L}{|\Sigma|})^B]^\ell & \text{for a NFA} \end{cases}$$

The coverage of the FSA decreases as a and B decrease. The slope is more abrupt in the DFA case than in the NFA case; still, there is clearly no phase transition here.

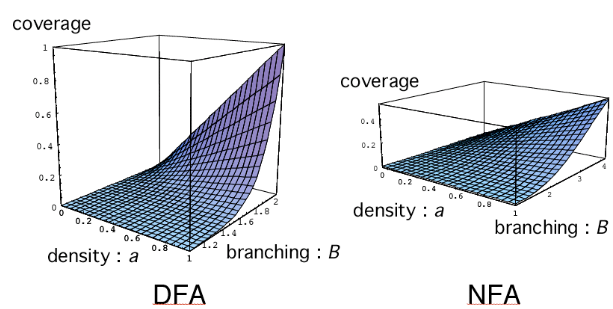

Figure 1: Coverage landscapes for Deterministic and Non-Deterministic FSA, for $|\Sigma|$=2, L=1 and ℓ=10. The density of accepting states a and the branching factor B respectively vary in $[0, 1]$ and $\{1, 2\}$.

3.2 PT in the Generalization Cone

The coverage landscape displayed in Fig. 1 might suggest that grammatical inference takes place in a well-behaved search space. However, grammatical inference algorithms do

[2] In this paper, after the standard Machine Learning terminology, a string is said to be covered by a FSA *iff* it belongs to the language thereof.

[3] The string is cut at the last accepting state met before arriving at length ℓ, if any; otherwise it is rejected.

not explore the whole FSA space. Rather, as stated in section 2.1, the search is restricted to the generalization cone, the set of generalizations of the PTA formed from the set S^+ of the positive examples. The next step is thus to consider the search space actually explored by GI algorithms.

A new sampling mechanism is defined to explore the DFA generalization cone:

1. $|S^+|$ ($= 200$ in the experiments) examples of length ℓ are uniformly and independently sampled within the space of all strings of length $< \ell$, and the corresponding PTA is constructed;
2. N ($= 50$ in the experiments) PTAs are constructed in that way.
3. K ($= 20$ in the experiments) generalization paths, leading from each PTA to the most general FSA or Universal Acceptor (UA), are constructed;
   In each generalization path ($A_0 = PTA, A_1, \ldots, A_t = UA$), the i-th FSA A_i is constructed from A_{i-1} by merging two uniformly selected states in A_{i-1}, and subsequently applying the determinisation operator.
4. The generalization cone sample is made of all the FSAs in all generalization paths (circa 270,000 FSAs in the experiments).

The sampling mechanism on the non-deterministic generalisation cone differs from the above in a single respect: the determinisation operator is never applied.

Fig. 2 shows the behaviour of the coverage in the DFA generalisation cone for $|\Sigma| = 4$ and $\ell = 8$. Each DFA A is depicted as a point with coordinates (Q, c), where Q is the number of states of A and c is its coverage (measured as in section 3). The coverage rate for each FSA in the sample is evaluated from the coverage rate on 1000 test strings of length ℓ.

Fig. 3 similarly shows the behaviour of the coverage in the NFA generalisation cone, with $|\Sigma| = 4$ and $\ell = 16$.

Fig 2, typical of all experimental results in the range of observation ($|\Sigma| = 2, 4, 8, 16$, and $\ell = 2, 4, 6, 8, 16, 17$), shows a clear-cut phase transition. Specifically, here, the coverage abruptly jumps from circa 13% to 54%; and this jump coincides with a gap in the number of states of the DFAs in the generalization cone: no DFA with a number of states in $[180, 420]$ was found. The gap is even more dramatic as the length of the training and test sequences ℓ is increased.

Interestingly, a much smoother picture appears in the non-deterministic case (fig. 3); although the coverage rapidly increases when the number of states decreases from 300 to 200, no gap can be seen, neither in the number of states nor in the coverage rate itself[4].

In the following, we focus on the induction of DFAs.

4 Phase transition and search trajectories

The coverage landscape, for the DFAs, shows a hole in the generalization cone, with a density of hypotheses of coverage in between a large interval (typically between less than 20% to approximately 60%) falling abruptly. Therefore, a random exploration of the generalization cone would face severe difficulties in finding a hypothesis in this region and would likely return hypotheses of poor performance if the target concept had a coverage rate in this "no man's land" interval.

It is consequently of utmost importance to examine the search heuristics that are used in the classical grammatical inference systems. First, are they able to thwart the *a priori* very low density of hypotheses in the gap? Second, are they able to guide the search toward hypotheses of appropriate coverage rate, specially if this coverage falls in the gap?

The study will thus focus on two standard algorithms in grammatical inference, namely the RPNI and the RED-BLUE algorithms [Oncina and Garcia, 1992; Lang *et al.*, 1998].

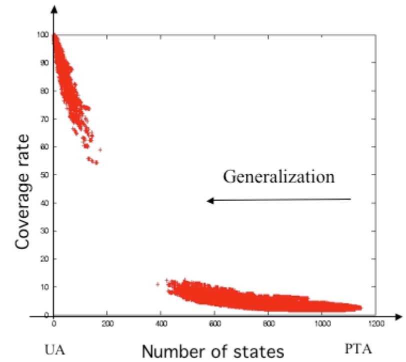

Figure 2: Coverage landscape in the DFA generalization cone ($|\Sigma| = 4$, $\ell = 8$). At the far right stand the 50 PTA sampled, with circa 1150 states each. The generalization cone of each PTA includes 1,000 generalization paths, leading from the PTA to the Universal Acceptor. Each point reports the coverage of a DFA, evaluated over a sample of 1,000 strings. This graph shows the existence of a large gap regarding both the number of states and the coverage of the DFAs that can be reached by generalization.

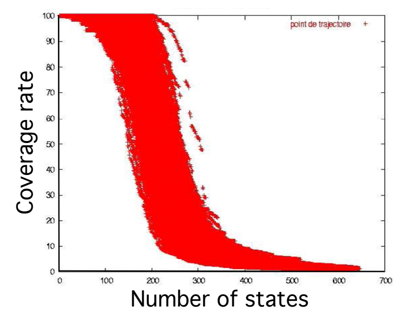

Figure 3: Coverage landscape in the NFA generalization cone, with same order parameters as in fig. 2.

[4]The difference with the DFA case is due to the determinisation process that forces further states merging when needed. A diffusion like analytical model was devised, that predicts the observed start of the gap with 15% precision.

4.1 Experimental setting

Previous experiments considered training sets made of positive randomly drawn strings sequences only. However, in order to assess the performance of learning algorithms, the hypothesis learned must now be compared to the target automaton. Therefore, another experimental setting is used in this section, with the sampling of target automata, and the construction of training and test sets. These data sets include positive and negative examples as most GI algorithms (and specifically RPNI and RED-BLUE) use negative examples in order to stop the generalization process.

In our first experiments, we tested whether heuristically guided inference algorithms can find good approximations of the target automata considering target automata with approximately (i) 50% coverage rate (as considered in the influential *Abbadingo* challenge, and in the middle of the "gap"), and (ii) 5% coverage rate.

For each target coverage rate, we used the experimental setting described in [Lang *et al.*, 1998] in order to retain a certain number of target automata with a mean size of Q states ($Q = 50$, in our experiments). For each automaton then, we generated N (=20) training sets of size $|S|$ (= 100) labeled according to the target automaton, with an equal number of positive and negative instances ($|S^+| = |S^-| = 50$) of length $\ell = 14$. The coverage rate was computed as before on 1000 uniformly drawn strings (with no intersection with the training set).

In a second set of experiments, we analyzed the learning performances of the algorithms with respect to test errors, both false positive and false negative.

In these experiments, we chose the type of target automata by setting the number of states Q and some predetermined structural properties[5].

4.2 The heuristically guided search space

Due to space limitation, only the graph obtained for the RPNI algorithm is reported (see figure 4), with three typical learning trajectories. Similar results were obtained with the RED-BLUE algorithm.

One immediate result is that both the RPNI and the EDSM heuristics manage to densely probe the "gap". This can explain why the gap phenomenon was not discovered before, and why the RED-BLUE algorithm for instance could solve some cases of the *Abbadingo* challenge where the target concepts have a coverage rate of approximately 50%. However, where RPNI tends to overspecialize the target automaton, RED-BLUE tends to overgeneralize it by 5% to 10%.

In order to test the capacity of the algorithms to return automata with a coverage rate close to the target coverage, we repeated these experiments with target automata of coverage rate of approximately 3%. The results (figure 5) shows that, in this case, RPNI ends up with automata of coverage 4 to 6 times greater than the target coverage. The effect is even more pronounced with RED-BLUE which returns automata of average coverage rate around 30%!

[5]The datasets and more detail are available at `http://www.lri.fr/~antoine/www/pt-gi/`

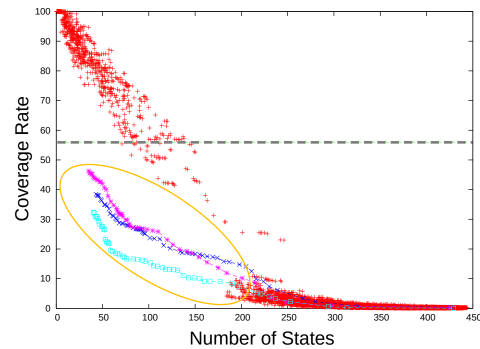

Figure 4: Three RPNI learning trajectories for a target concept of coverage=56%. Their extremity is outlined in the oval on the left. The doted horizontal line corresponds to the coverage of the target concept. The cloud of points corresponds to random trajectories.

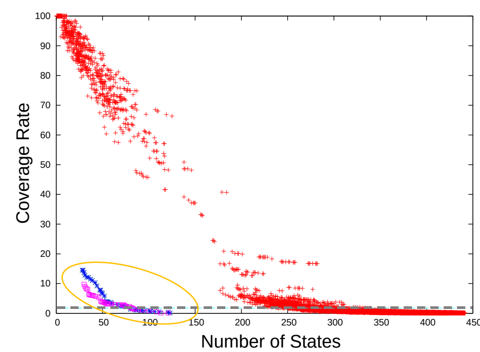

Figure 5: Same as in figure 4, except for the coverage of the target concept, here 3%.

4.3 Generalization error

Table 1, obtained for different sizes of the target automata and for training sets of structural completeness above 40%, confirms that both RPNI and RED-BLUE return overgeneralized hypotheses. On one hand, their average coverage is vastly greater than the coverage of the target automata, on the other hand, they tend to cover only part of the positive test instances, while they cover a large proportion of the negative test instances. This shows that the heuristics used in both RPNI and RED-BLUE may be inadequate for target concepts of low coverage.

5 Conclusion

This research has extended the Phase Transition-based methodology [Botta *et al.*, 2003] to the Grammatical Inference framework. Ample empirical evidence shows that the search landscape presents significant differences depending on the search operators that are considered.

Algo.	Q_c	$ucov_c$	Q_f	$ucov_f$	$pcov_f$	$ncov_f$
RB	15	5.97	10.38	33.81	60.93	34.69
RB	25	4.88	12.77	40.35	62.68	37.87
RB	50	4.2	14.23	45.38	66.14	42.23
RB	100	3.39	13.13	30.35	42.81	28.69
RPNI	15	5.95	5.14	22.9	57.51	26.99
RPNI	25	4.7	7.56	23.07	56.38	25.98
RPNI	50	3.87	14.08	23.45	51.89	24.42
RPNI	100	3.12	26.41	23.151	50.12	24.40

Table 1: Performances of RED-BLUE (RB) and RPNI for target DFA of sizes Q = 15, 25, 50 and 100 states. Q_f, $ucov_f$, $pcov_f$ and $ncov_f$ respectively denote the average size of the learned automata, their average coverage, the true positive and the false positive rates.

A first result is that random search appears to be more difficult in the DFA generalization cone than in the whole search space: a large gap was found, in terms of hypothesis coverage and size. This remark explains why sophisticated search biases are needed for grammatical inference algorithms in the problem range corresponding with the hole in the generalization cone.

A second finding regards the limitations of the search operators in RPNI and RED-BLUE, especially outside the region of the *Abbadingo* target concepts. Experiments with artificial learning problems built from target concepts with coverage less than 10% reveal that RPNI and RED-BLUE alike tend to learn overly general hypotheses; with respect to both the size (estimated by the number of states) and the coverage of the hypotheses, often larger by an order of magnitude than that of the target concept. What is even more worrying, is that this overgeneralization *does not imply* that the found hypotheses are complete: quite the contrary, the coverage of the positive examples remains below 65%, in all but one setting.

The presented study opens several perspectives for further research. First, it suggests that the learning search, and especially the stopping criterion, could be controlled using a hyper-parameter: the coverage rate of the target concept (possibly supplied by the expert, or estimated e.g. by cross-validation). In other words, the stopping criterion of the algorithms might be reconsidered. Secondly, more conservative generalisation operators will be investigated. Preliminary experiments done with e.g. reverted generalisation (same operator as in RPNI, applied on the reverted example strings) show that such operators can delay the determinisation cascade, and offer a finer control of the final coverage rate of the hypotheses.

Finally, the main claim of the paper is that the phase transition framework can be used to deliver indications regarding when and where specific search biases might fail – hopefully leading to understand and ultimately alleviate their limitations.

Acknowledgments

The last two authors are partially supported by the PASCAL Network of Excellence IST-2002-506 778.

References

[Botta *et al.*, 2003] Marco Botta, Attilio Giordana, Lorenza Saitta, and Michèle Sebag. Relational learning as search in a critical region. *Journal of Machine Learning Research*, 4:431–463, 2003.

[Cheeseman *et al.*, 1991] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the Really Hard Problems Are. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, IJCAI-91, Sidney, Australia*, pages 331–337, 1991.

[Coste, 1999] Francois Coste. State merging inference of finite state classifiers. Technical report, Irisa, May 1999.

[Dupont *et al.*, 1994] Pierre Dupont, Laurent Miclet, and Enrique Vidal. What Is the Search Space of the Regular Inference? In *Proceedings of the Second International Colloquium on Grammatical Inference and Applications, ICGI-94*, pages 25–37, 1994.

[Giordana and Saitta, 2000] Attilio Giordana and Lorenza Saitta. Phase transitions in relational learning. *Machine Learning*, 41:217–251, 2000.

[Gold, 1967] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[Hogg *et al.*, 1996] Thomas Hogg, Bernard Hubberman, and C. Williams. Phase transitions and the search problem. *Artificial Intelligence*, 81:1–15, 1996.

[Lang *et al.*, 1998] Kevin Lang, Barak Pearlmutter, and Rodney Price. Results of the abbadingo one dfa learning competition and a new evidence driven state merging algorithm. In *Fourth International Colloquium on Grammatical Inference (ICGI-98)*, volume LNCS-1433, pages 1–12. Springer Verlag, 1998.

[Muggleton and Raedt, 1994] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.

[Oncina and Garcia, 1992] J. Oncina and P. Garcia. Inferring regular languages in polynomial update time. *Pattern Recognition and Image Analysis*, pages 49–61, 1992.

[Pitt, 1989] Leonard Pitt. Inductive inference, dfas, and computational complexity. In *Proceedings of the Workshop on Analogical and Inductive Inference (AII-89)*, volume LNCS-397, pages 18–44. Springer Verlag, 1989.

[Sakakibara, 1997] Yasubumi Sakakibara. Recent advances of grammatical inferences. *Theoretical Computer Science*, 185:15–45, 1997.

[Vapnik, 1995] Vladimir Vapnik. *The Nature of Statistical Learning.* Springer, 1995.