The phase transition of the bounded ILP consistency problem

Erick Alphonse

LIPN-CNRS UMR 7030, Université Paris 13, France
erick.alphonse@lipn.univ-paris13.fr

Abstract. A number of recent works have been focusing on analysing the phase transition of the NP-complete ILP covering test, which have been fruitful in linking this phenomenon to plateaus during heuristic search. However, it is only a facet of the ILP complexity as it is very dependent of the search strategy. Its inherent difficulty has to be studied as a whole to design efficient learners. ILP is arguably harder than attribute-value learning, which has been formalised by Gottlob et al. who showed that the simple bounded ILP consistency problem is $\Sigma_2-complete$. Some authors have predicted that a phase transition could be exhibited further up the polynomial hierarchy and we show this is the case in this problem space, where the number of positive and negative examples are order parameters. Those order parameters are the same as for the k-term DNF consistency problem studied in the context of attribute-value learning. We show that the learning cost exhibits the easy-hard-easy pattern with a lgg-based learner.

1 Introduction

The phase transition framework, which has been strongly developped in many combinatorics domains, like in SAT or CSP domains, since [1], has changed the way search algorithms are empirically evaluated. This lead to new designs of search algorithms, from incomplete to complete solvers and from deterministic to randomised solvers [2].

Symbolic learning, which has been cast more than 20 years ago as search into a state space [3] has known few developments of the phase transition (PT) framework. As far as we know, the only work that studied the PT of learning has been done by [4] who showed that the number of positive and negative examples where order parameters of the k-term DNF consistency problem, which is NP-complete. Indeed, if one keeps one parameter constant and varies the other, one wanders from an under-constraint region, named the "yes" region, associated with a small value, where there is almost surely a solution, to an over-constraint region, named the "no" region, as the parameter value increases, where almost surely no generalisation of the positive examples is correct. A related work studied the PT of the subsumption test which is a key NP-complete sub-problem of learning. Although, this study has been fruitful in linking this phenomenon to plateaus during heuristic search [5], it is only a facet of the ILP complexity as it is very dependent of the search strategy and does not study the complexity of learning in the PT framework.

ILP is arguably harder than attribute-value learning, like k-term DNF learning, which has been formalised by Gottlob et al. [6] who showed that the simple bounded ILP consistency problem is Σ_2-complete. This is one class higher in the polynomial hierarchy than NP-complete (or Σ_1-complete) problems. Some authors, e.g. [7], have predicted that a phase transition could be exhibited further up the polynomial hierarchy and therefore that this framework could be useful to other PSPACE-complete problems.

We show this holds for the bounded ILP consistency problem, where the number of positive and negative examples are order parameters of the phase transition. We show that the median learning cost exhibits the easy-hard-easy pattern with a simple lgg-based learner.

We present, in the next section, the necessary background on the bounded ILP consistency problem and the model RLPG which is a generator proposed to study this problem, first described in [5]. The section 3 will present the complete learner used to answer the ILP consistency problem. Section 4 will exhibit the phase transition, beyond NP, of the ILP consistency problem with respect to the two order parameters which are the number of positive and negative examples. We show that the solver used allow to exhibit the easy-hard-easy pattern of median search cost. Finally, we will conclude and draw some perspective and benefits of these results for relational learning.

2 Background

In this article, we study what has been termed the bounded ILP consistency problem for function-free Horn clauses by [6]. Given a set of positive examples E^+ and a set of negative examples E^- of function-free ground Horn clauses and an integer k polynomial in $|E^+ \cup E^-|$, does there exist a function-free Horn clause h with no more than k literals such that h θ-subsumes each element in E^+ and h does not θ-subsume any element in E^-.

[5] proposed a random generator for this problem, named model RLPG (Relational Learning Problem Generator). A learning problem instance in this model is denoted $RLPG(k, n, \alpha, N, Pos, Neg)$. The parameters k, n, α, N are related to the definition of the hypothesis and example spaces. Pos and Neg are the number of positive and negative examples respectively. The first four parameters are defined in order to ensure that a subsumption test between a hypothesis and an example during search encode a valid CSP problem, following models for random CSP. This requirement is imposed as the model RLPG was proposed to study the impact of the phase transition of the subsumption test on heuristic search. We briefly recall their meaning and focus on the last two parameters.

$k \geq 2$ denotes the arity of each predicate present in the learning language, $n \geq 2$ the number of variables in the hypothesis space, n^α the domain size for all variables, and finally N the number of literals in the examples built on a given predicate symbol. Given k and n, the size of the bottom clause of the hypothesis space $\mathcal{L}_h$ is $\binom{n}{k}$. It encodes the largest constraint network of the underlying CSP model. Each constraint between variables is encoded by a literal built on a unique predicate symbol. $\mathcal{L}_h$ is then defined as the power set of the bottom clause, which is isomorphic to a boolean lattice. Its size is $2^{\binom{n}{k}}$.

Learning examples are randomly drawn, independently and identically distributed, given k, n, α and N. Their size is $N\binom{n}{k}$. Each example defines N literals for each predicate symbol. The N tuples of constants used to define those literals are drawn uniformly and without replacement from the possible set of $\binom{n^\alpha}{k}$ tuples.

3 Exhibiting the easy-hard-easy pattern with a complete solver

Besides the phase transition behaviour of decision problems, a strong motivation of its study is that it is conjectured that the hardest problem instances occur in the phase transition (see e.g. [1, 8, 7]). The under-constraint problems from the "yes" region appear to be easily solvable, as there are a lot of solutions. This is the same for over-constraint problems from the "no" region as it is easy to prove that they are insoluble. These findings have been corroborated on several problems, with different types of algorithms, and it is considered that the problem instances appearing in the phase transition are inherently hard, independently of the algorithms used. In the "yes" and "no" regions, the easy ones, the complexity appears to be very dependent of the algorithm. There are, in these regions, some problems exceptionally hard, whose complexity dominates the complexity of instance problems in the phase transition region for certain types of algorithm [8].

In other words, exhibiting the easy-hard-easy pattern require a "good" algorithm. We propose to use a depth-first lgg-based algorithm to solve the ILP consistency problem, DF-BDD (Depth-First Bottom-up Data-Driven), which is similar to the approach of [4] for the k-term DNF consistency problem. Its Prolog code is given below:

```
1 df_bdd(Sol,[],_,Sol).
2 df_bdd(Hypo,[Pos|L_Pos],L_Neg,Sol) :-
3         lgg(Hypo,Pos,LGG), % non-deterministic computation of a LGG
4         % consistency check
5         correctness(LGG,L_Neg),
6         df_bdd(LGG,L_Pos,L_Neg,Sol).
```

The computation of lggs (line 3) is done with depth-first search into possible subsets of the hypothesis. It outputs the largest subsets that subsume the example. The implementation is rather naive, which may blur the easy-hard-easy pattern, as we will see. Once a lgg has been computed, we test, in a depth-first way, if it is correct with respect to all negative examples (line 5).

4 Numbers of positive and negative examples as order parameters

In this section, we study the effect of the number of positive and negative examples on the solubility probability and the solving cost of the ILP bounded consistency problem. If we refer to section 2, *RLPG* is parametrised with 6 parameters but we only study the last two, *Pos* and *Neg*, as the effect of the other parameters have been already studied in [5] for constant number of positive and negative examples. Here, we focus on few settings for these parameters, with $k = 2$, $n = 5$ and $n = 6$, to study different problem sizes, $\alpha = 1.4$ and $N = 10$. The choice of these parameters ensures that we do not generate trivially insoluble problems [7], but also various experiments, not shown here, indicated that there were representative of the phase transition behaviour of the ILP consistency problem. In all experiments below, statistics were computed from a sample of 500 learning problems.

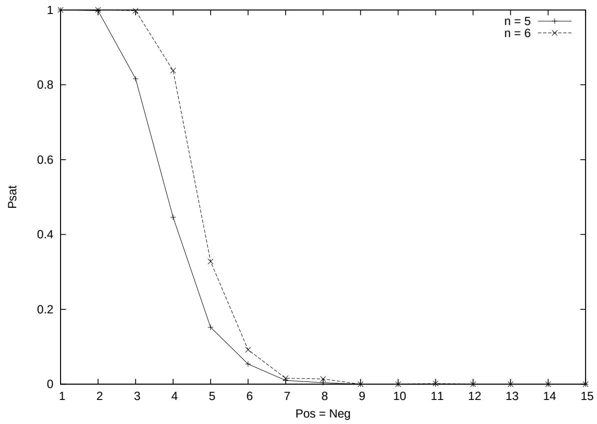

Fig. 1. Probability of statisfiability according to the number of learning examples (Pos = Neg), with $n = 5$ and $n = 6$

We start by varying both *Pos* and *Neg*. Figure 1 shows the solubility probability of the ILP consystency problem when $Pos = Neg$ are varied from 1 to 25, for $n = 5$ and $n = 6$. As we can see, when the number of examples is small, there is almost surely a consistent hypothesis, and when the number is large it is almost surely impossible to find a consistent hypothesis. The cross-over point, where the probability of solubility is about 0.5, is around 4 for $n = 5$ and 5 with $n = 6$. It is not surprising that it increases with bigger problems. For $n = 5$, the hypothesis space size is 2^{10} and 2^{15} for $n = 6$. We could not conduct experiments for larger values of n as the hypothesis space grows too fast in *RLPG*. For instance, $n = 7$ sets a hypothesis space of size 2^{21}, which cannot be handled by our complete solver. In the future, it would be interesting to modify *RLPG* to specify the size of the bottom clause and then draw the number of variables accordingly.

Figure 2 and 3 show the associated cost (the median cost along with the 25th and 75th percentiles) to solve the problem instances, with $n = 6$. We measured the cost by recording the time in milliseconds, as well as the number of backtracks of the subsumption procedure, needed to solve a learning problem. The latter seems relevant, as the subsumption test is used to compute the lggs.
We can see that a complexity peak is associated with instances in the phase transition region, and that the search cost follows the easy-hard-easy pattern. The complexity in the "no" region slowly decreases as the number of examples increases, where we could have expected a sharper decrease, but it may be related to our implementation.

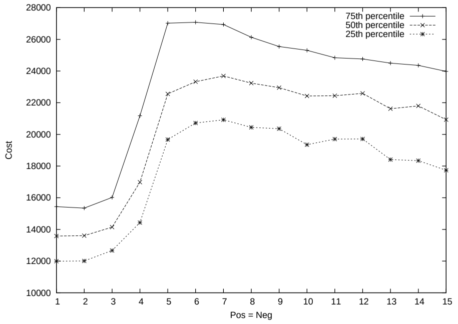

Fig. 2. Cost in resolution time (ms.) according to the number of learning examples (Pos = Neg), for $n = 6$

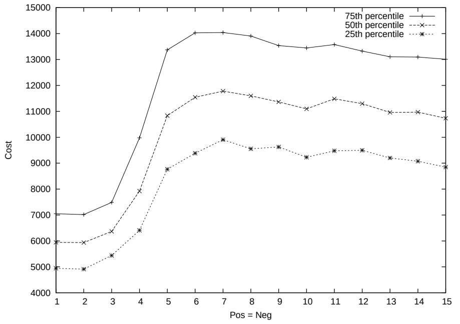

Fig. 3. Cost in number of bactracks of the subsumption test according to the number of learning examples (Pos = Neg), for $n = 6$

We study now the phase transition along the number of positive examples, for constant values of Neg, but omit cost plots. The results are almost symmetric when Pos is constant and Neg varies, and is not shown here. Figures 4 and 5 show the phase transition when Pos varies from 1 to 25, for $n = 5$ and $n = 6$ respectively. The transition becomes sharper as Pos increases, which is not surprising as the subset of complete hypotheses shrinks with Pos.

5 Conclusion

It is conjectured that the phase transition of decision problem can be exhibited further up the polynomial hierarchy and therefore that this framework could be useful to other PSPACE-complete problems. We have shown that this holds with the bounded ILP cosistency problem, a Σ_2-complete problem, which exhibits a phase transition in its solubility, with the number of positive and negative examples as order parameters. The search cost as given by a depth-first lgg-based solver exhibits the easy-hard-easy pattern. This is the first work that

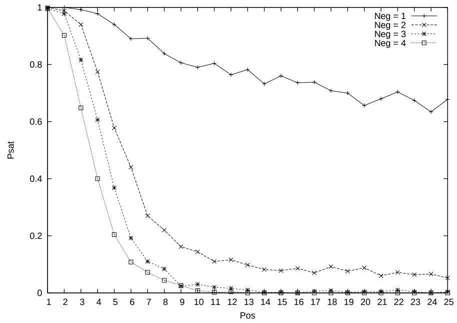

Fig. 4. Probability of statisfiability according to the number of positive examples with $n = 5$, for $Neg = 1, 2, 3, 4$

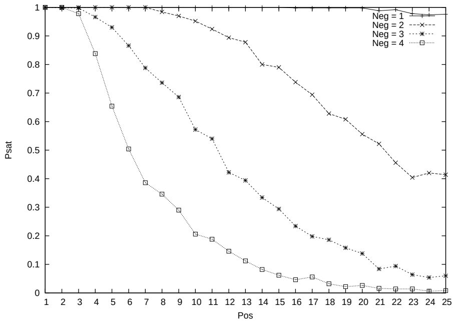

Fig. 5. Probability of statisfiability according to the number of positive examples with $n = 6$, for $Neg = 1, 2, 3, 4$

study the phase transition of learning in ILP and we hope that it will stimulate algorithmic developments, in the line of what has been done in combinatorics. It points out interesting follow-ups: the model RLPG has been used to generate random problems and we plan to study the impact of its other parameters on the generation of hard instances; we plan to generate hard problems to study the different solvers proposed in ILP.

Acknowledgment

We thank Aomar Osmani for his support and the numerous discussions about this work, as well as Henry Soldano and Dominique Bouthinon.

References

1. Cheeseman, P., Kanefsky, B., Taylor, W.: Where the really hard problems are. In Proc. of the 12th Int. Joint Conf. on Artificial Intelligence. (1991) 331–340
2. Carla Gomes, Heny Kautz, A.S., Selman, B.: Satisfiability solvers. In: Handbook of Knowledge Representation. (2007)
3. Mitchell, T.M.: Generalization as search. Artificial Intelligence **18** (1982) 203–226
4. Rückert, U., Kramer, S., Raedt, L.D.: Phase transitions and stochastic local search in k-term DNF learning. Lecture Notes in Computer Science **2430** (2002) 405–417
5. Alphonse, E., Osmani, A.: A model to study phase transition and plateaus in relational learning. In: Proc. of Inductive Logic Programming. (2008) to be published.
6. Gottlob, G., Leone, N., Scarcello, F.: On the complexity of some inductive logic programming problems. In Proc. of the 7th Int. Workshop on Inductive Logic Programming. Volume 1297 of LNAI. (1997) 17–32
7. Gent, I.P., Walsh, T.: Beyond np: the qsat phase transition. In Proc. of the 16th Nat. Conf. on Artificial intelligence. (1999) 648–653
8. Davenport, A.: A comparison of complete and incomplete algorithms in the easy and hard regions. In: Work. on Studying and Solving Really Hard Problems, CP-95. (1995) 43–51