Étude empirique de la transition de phase en apprentissage relationnel

Erick Alphonse

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

Résumé : L'Apprentissage Relationnel (AR) a suscité un grand intérêt en Fouille de Données récemment, étant vu comme un moyen de combler le fossé entre des algorithmes d'apprentissage attribut-valeur efficaces, travaillant sur une seule table ou relation, et un nombre croissant d'applications, intrinsèquement relationnelles, dont les données sont stockées dans les bases de donnéees relationnelles. Cependant, les systèmes d'AR actuels utilisent des solveurs génériques qui ne leurs permettent pas de passer à l'échelle. Cette approche est à opposer aux progès réalisés depuis une dizaine d'années dans les communautés de la combinatoire, telles que SAT ou CSP, où l'étude de problèmes aléatoires, dans le cadre de la transition de phase, a permis d'évaluer et de développer des algorihmes spécialisés très performants. Dans cet article, nous argumentons que l'import d'un tel outil est nécessaire au développement de l'AR. Un certain nombre de travaux récents se sont penchés sur l'analyse de la transition de phase d'un sous-problème de l'AR, le test de subsomption NP-complet, qui a été utile pour faire le lien entre ce phénomène de transition de phase et les plateaux de la recherche heuristique. Cependant, ce n'est qu'une facette de la complexité de l'AR car celle-ci est très dépendante de la stratégie de recherche. Sa complexité intrinsèque doit être globalement analysée pour développer des algorithmes efficaces. L'AR est plus difficile que l'apprentissage attribut-valeur, ce qui a été formellement montré par Gottlob et al. qui ont démontré que le problème dit problème de cohérence borné simple est $\Sigma_2 - complet$. Certains auteurs ont prédit que la transition de phase pouvait être exhibée au-delà de NP, dans toute la hiérachie polynomiale, et nous montrons que c'est le cas en AR. Nous proposons un générateur d'instances de problèmes aléatoire pour l'AR, où le nombre d'exemples positifs et négatifs sont les paramètres d'ordre de la transition de phase. Nous montrons que le coût d'apprentissage exhibe bien le profile "easy-hard-easy" avec un algorithm d'apprentissage de type lgg.

Mots-clés : Transition de phase, Complexité, Apprentissage relationnel

1 Introduction

Even though the expressiveness of (supervised) Relational Learning (RL), also known as Inductive Logic Programming (ILP), is attractive for many modern applications[1], such as life sciences, environmental sciences, engineering, natural language processing or arts (see also (Domingos, 2003)), RL has to face the well-known trade-off between expressivity and efficiency.

From the efficiency perspective, one of the major obstacles is search efficiency, and several authors have acknowledged that a step forward would be in the design of novel search techniques (e.g. (Page & Srinivasan, 2003; Domingos, 2003)). RL, as a sub-domain of symbolic learning, has been cast more than 25 years ago as search into a state space (Mitchell, 1982) : given a hypothesis space defined a priori, identified by its representation language, find a hypothesis consistent with the learning data. This seminal paper, relating symbolic learning to search in a space state, has enabled machine learning to integrate techniques from problem solving, operational research and combinatorics : greedy search in FOIL, beam search in ICL, breadth-first search in Aleph, 'A' search in PROGOL, IDA (Iterative-Deepening A) search in MIO to name a few systems[2]. Besides very few exceptions, all systems are rooted in the generate-and-test paradigm (Newell & Simon, 1972) and therefore rely on general-purpose search strategies.

This approach has to be contrasted with the important progress, made during the past few years, in the performance of boolean satisfiability (SAT) solvers, which can deal with problems' sizes that are orders of magnitude larger than this research community would expect to solve only a decade ago. This progress has been driven by studies of randomly generated problem instances, in the *phase transition framework*, where hard to solve instances can be reliably generated, independently from the solver used. This framework, strongly developed in many combinatorics domains, as in SAT or CSP domains, since (Huberman & Hogg, 1987; Cheeseman *et al.*, 1991), has changed the way search algorithms are empirically evaluated. This has lead to new designs of search algorithms, from incomplete to complete solvers and from deterministic to randomised solvers (see e.g. (Carla Gomes & Selman, 2007)).

We think that RL, as a combinatorics field, must follow the same path in order to re-new and improve its algorithmic approach in order to answer the new challenges of modern applications. More generally, symbolic learning has seldom known any developments of the Phase Transition (PT) framework. As far as we know, the only work that studied the PT of learning has been done in attribute-value by (Rückert *et al.*, 2002) who showed that the number of positive and negative examples where order parameters of the k-term DNF consistency problem, a well-known NP-complete problem (Kearns & Vazirani, 1994). Indeed, if one keeps the number of positive examples constant and varies the number of negative examples, one wanders from an under-constraint region, named the "yes" region, with few negative examples, where there is almost surely a consistent hypothesis, to an over-constraint region, named the "no" region, as the number of negative examples increases, where almost surely no generalisation of the posi-

[1]http ://www-ai.ijs.si/ ilpnet2/apps/index.html

[2]Relevant information on these systems can be found at http ://www-ai.ijs.si/ ilpnet2/systems

tive examples is correct. The same holds if we keep the number of negative examples constant and vary the number of positive examples. A related work studied the PT of the subsumption test, which is a key NP-complete sub-problem of relational learning (Giordana & Saitta, 2000; Botta *et al.*, 2003). Although, this study has been fruitful in linking this phenomenon to plateaus during heuristic search (Alphonse & Osmani, 2008), 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 as a whole in the PT framework. Also, this framework inspired a number of approaches in learning where threshold phenomena of the generalisation error are shown depending on learning parameters (e.g. Ahr *et al.* (1999); Baskiotis & Sebag (2004)).

RL is arguably harder than attribute-value learning, like k-term DNF learning, which has been formalised by Gottlob *et al.* (1997) who showed that the simple bounded ILP consistency problem, which will be discussed in later, is Σ_2-complete. This is one class higher in the polynomial hierarchy than NP-complete (or Σ_1-complete) problems. Some authors, (Bylander, 1996; Gent & Walsh, 1999), have conjectured that a phase transition could be exhibited further up the polynomial hierarchy and therefore that this framework could be useful to other PSPACE problems. This was supported by results on planning and QBF-2 (Quantified Boolean Formulas with two alternating quantifiers, see also (Chen & Interian, 2005)).

In this paper, we show that this also holds true for the bounded ILP consistency problem : we propose a random problem instance generator 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. As such, it is the first work that studies the phase transition of RL.

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 (Alphonse & Osmani, 2008). Section 3 will present the complete learner used to answer the ILP consistency problem. Section 4 will then go on to 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 allows to exhibit the easy-hard-easy pattern of median search cost. Finally, we will conclude with a discussion on some benefits and perspectives of these results for RL.

2 Background

In this article, we study what has been termed the bounded ILP consistency problem for function-free Horn clauses by (Gottlob *et al.*, 1997). Given a set of positive examples E^+ and a set of negative examples E^- of function-free ground Horn clauses and an integer l polynomial in $|E^+ \cup E^-|$, does there exist a non-recursive function-free Horn clause h with no more than l literals such that h is consistent with E^+ and E^- :

– h logically implies each element in E^+ (completeness)
– h does not logically implies each element in E^- (correctness)

In such hypothesis space, the logical implication is equivalent to θ-subsomption[3] which is NP-complete and therefore decidable (Gottlob, 1987).

The consistency problem is fundamental in learning as it is at the core of the Statistical Learning Theory, notably studied in the PAC framework (see (Haussler, 1989; Kearns & Vazirani, 1994) for details). This *a fortiori* is true in RL where almost all noise-resistant learners are relaxation of this problem (Fürnkranz, 1997), therefore studying this problem will benefit search strategies for learning.

(Gottlob *et al.*, 1997) proved that this problem is Σ_2^P-complete (or NP^{NP}) : the search is NP-complete and it is guided by the subsumption test which is NP-complete. (Alphonse & Osmani, 2008) proposed a random generator for this problem, named model RLPG (Relational Learning Problem Generator), although they did not study the PT of learning. A learning problem instance 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.This requirement is imposed as the model RLPG was proposed to study the impact of the phase transition of the NP-complete subsumption test on heuristic search. We recall their meaning and focus on the last two parameters, which were not studied before and which will be shown to be order parameters of the phase transition of the ILP consistency problem.

$k \geq 2$ denotes the arity of each predicate present in the learning language, $n \geq 2$ the number of variables in the hypothesis space, α the domain size for all variables as being equal to n^α, 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 $\binom{n}{k}$, and 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. Each example defines N literals for each predicate symbol and then its size is $N.\binom{n}{k}$. 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.

As an illustration, table 1 shows a random $RLPG(2, 3, \alpha, 1, 1, 1)$ problem, with α such that $n^\alpha = 5$. The first line shows the bottom-most element of the hypothesis space, which encodes all binary constraints defined from the set of 3 variables. The next two lines show the positive and the negative example, respectively, allowing only one matching of a given predicate symbol (as $N = 1$). The search space is of size 2^3 and consists of all hypothesis built with the same head as the bottom clause, and with a subset of its body as body. In such a space, it is easy to see that there is no solution, given that no hypothesis subsumes the positive example without subsuming the negative example.

Whereas, the problem illustrated in table 2 accepts the following clause as solution : $p0(A) \leftarrow p2(A, B, D), p3(A, C, D)$

[3]The clause C θ-subsumes the clause D iff there exists a substitution θ such that $C\theta \subseteq D$

$\perp$	$p0(A) \leftarrow p1(A,B,C), p2(A,B,D), p3(A,C,D)$
$+$	$p0(e1) \leftarrow p1(e1,b,c), p2(e1,c,d), p3(e1,e,f)$
$-$	$p0(e2) \leftarrow p1(e2,c,f), p2(e2,d,e), p3(e2,d,c)$

TAB. 1 – Example of a random learning problem generated with RLPG, with no solution.

$\perp$	$p0(A) \leftarrow p1(A,B,C), p2(A,B,D), p3(A,C,D)$
$+$	$p0(e1) \leftarrow p1(e1,b,c), p2(e1,d,e), p3(e1,e,e)$
$-$	$p0(e2) \leftarrow p1(e2,b,b), p2(e2,e,e), p3(e2,e,c)$

TAB. 2 – Example of a random learning problem generated with RLPG, with a solution.

Note that an extra argument has been artificially added to each predicate in order to link all body literals to the head.

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

In the phase transition framework, it is conjectured that the hardest problem instances occur in the phase transition (see e.g. (Cheeseman *et al.*, 1991; Davenport, 1995; Gent & Walsh, 1999)). The under-constraint problems from the "yes" region appear to be easily solvable, as there are many 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 (Hogg & Williams, 1994; Davenport, 1995).

In other words, exhibiting the easy-hard-easy pattern requires a "good" algorithm. We propose to use a depth-first algorithm to solve the ILP consistency problem, DF-BDD (Depth-First Bottom-up Data-Driven), which has been proposed by (Plotkin, 1970) for such a problem, that is when a single clause is sought as a solution (see also (Haussler, 1989)). This is similar to the approach of (Rückert *et al.*, 2002) for the k-term DNF consistency problem. Its Prolog code is given below :

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

Starting from the bottom element, the algorithm generalises the current hypothesis to subsume each positive example in turn, until it outputs a consistent hypothesis, or until it proves that no correct hypothesis subsumes all positive examples. The generalisation step uses Plotkin's binary operator, namely the least-general generalisation (lgg) operator, which computes the least-general element that subsumes both input elements. Note that if the hypothesis space is not a lattice, which is the case here under θ-subsumption as the hypothesis space is finite, the lgg operator outputs all possible generalisations on subsequent backtracks. The computation of lggs (line 5) is done with depth-first search into possible subsets of the hypothesis (see (Kietz, 1993) for implementation details). It outputs the largest subsets that subsume the example. Once a lgg has been computed, we test, in a depth-first way, if it is correct with respect to all negative examples (line 7).

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 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 already been studied in (Alphonse & Osmani, 2008) 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 (Gent & Walsh, 1999), but also various experiments, not shown here, indicated that they 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.

We start by varying both Pos and Neg. Figure 1 shows the solubility probability of the ILP consistency problem when $Pos = Neg$ are varied from 1 to 15, 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 $n = 5$. 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

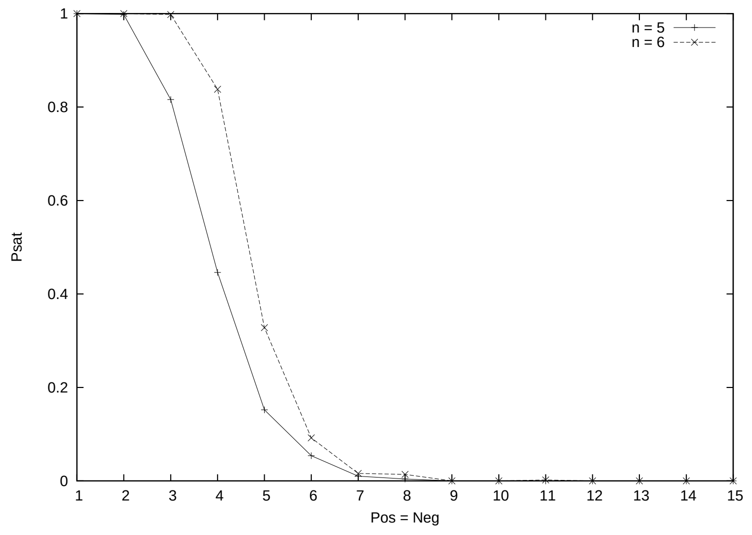

FIG. 1 – Probability of satisfiability according to the number of learning examples (=Pos = Neg), with $n = 5$ and $n = 6$

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 choice of algorithm. Figure 4 and 5 show the associated cost for problem instances with $n = 6$. The increase in cost is even sharper in the region where the hardest problems are located, for this set of parameters.

We study now the phase transition along the number of positive examples, for constant values of Neg, but omit cost plots. Figures 6 and 7 show the phase transition when Pos varies from 1 to 25, for $n = 5$ and $n = 6$ respectively. With no positive examples, the bottom element of the search space is solution, but as Pos increases, complete hypotheses get more general and eventually subsume a negative example. The transition becomes sharper as Neg increases, which is not surprising as the subset of correct hypotheses shrinks as Neg increases. The second order parameter is the number of negative examples Neg. The results almost have the same profile when Pos is constant and Neg varies, and are not shown here because of space constraints.

5 Conclusion

Although Relational Learning has been cast, more than 25 years ago, as search, it has known very few developments from the search strategy point of view and most learners rely on general-purpose solvers. This is a strong limitation to its applicability on

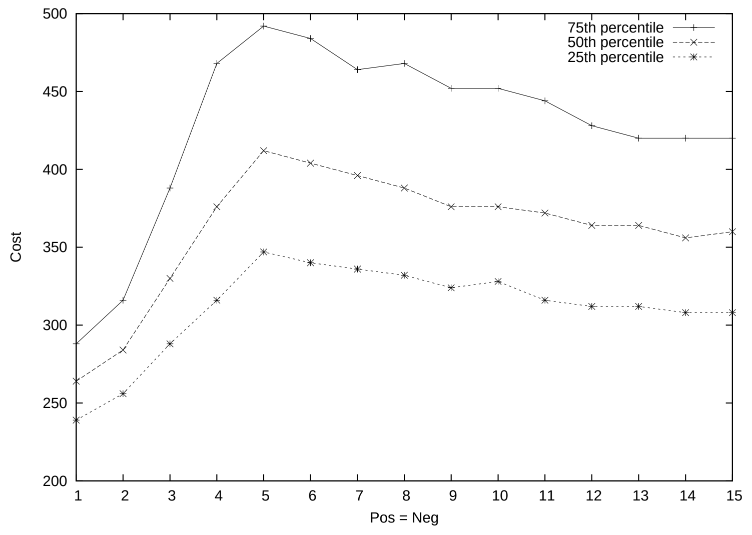

FIG. 2 – Cost in resolution time (ms.) according to the number of learning examples (= Pos = Neg), for $n = 5$

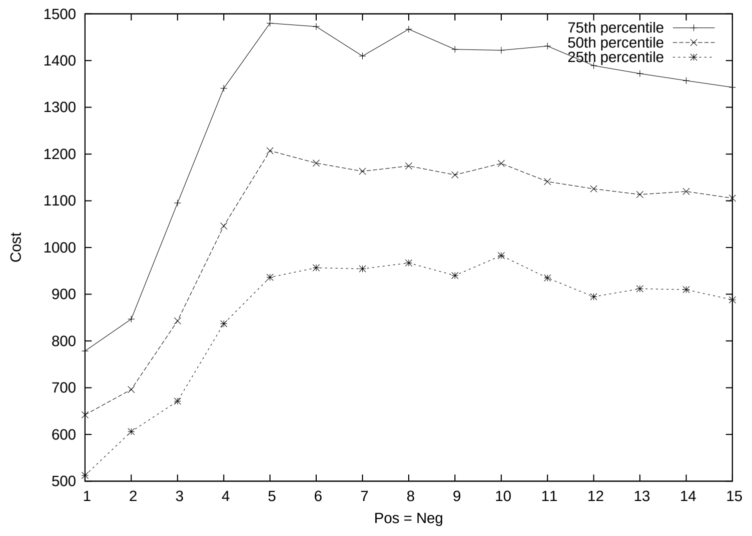

FIG. 3 – Cost in number of backtracks of the subsumption test according to the number of learning examples (= Pos = Neg), for $n = 5$

many modern applications, as it prevents RL to scale-up well. On the other hand, important progress has been made in other combinatorics communities, such as SAT and CSP, in the development of efficient specialised solvers, through the study of random

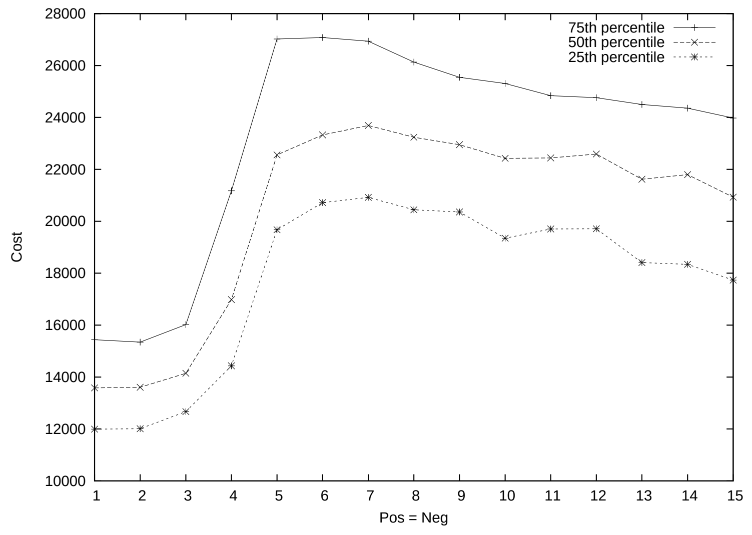

FIG. 4 – Cost in resolution time (ms.) according to the number of learning examples (= Pos = Neg), for $n = 6$

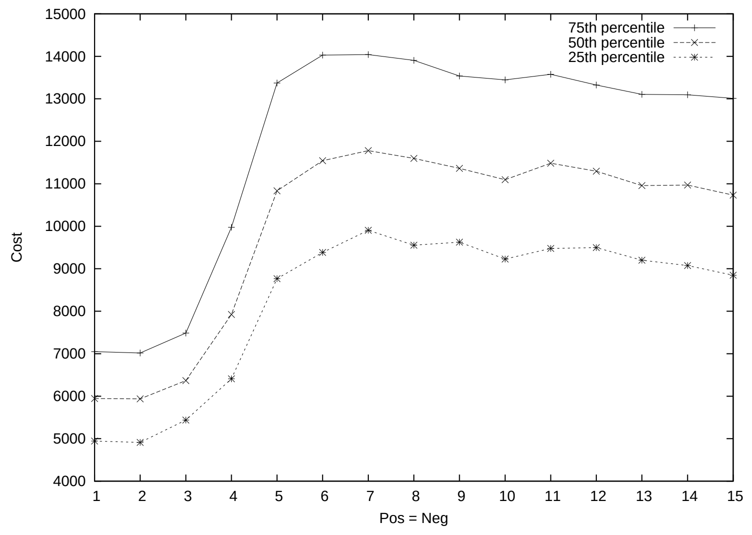

FIG. 5 – Cost in number of backtracks of the subsumption test according to the number of learning examples (= Pos = Neg), for $n = 6$

NP-complete problem generators in the phase transition framework. RL has a higher complexity, being at least Σ_2-hard in the general case. However, we argue that this framework will benefit RL, based on the conjecture that the phase transition can be

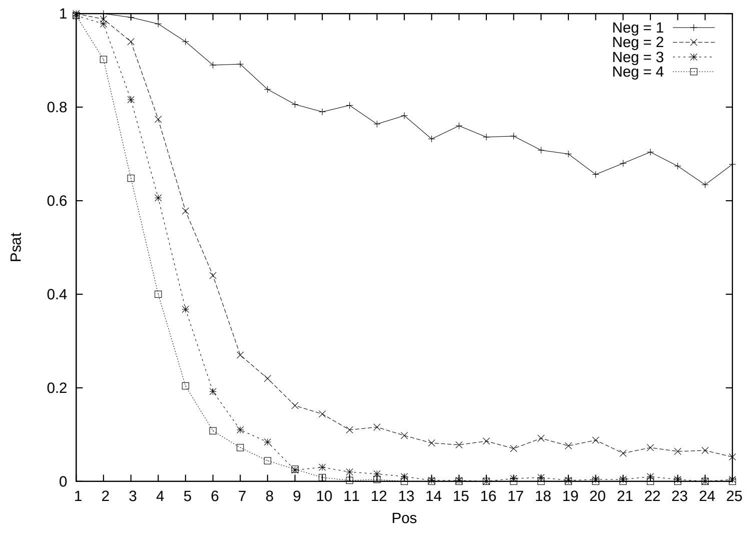

FIG. 6 – Probability of satisfiability according to the number of positive examples with $n = 5$, for $Neg = 1, 2, 3, 4$

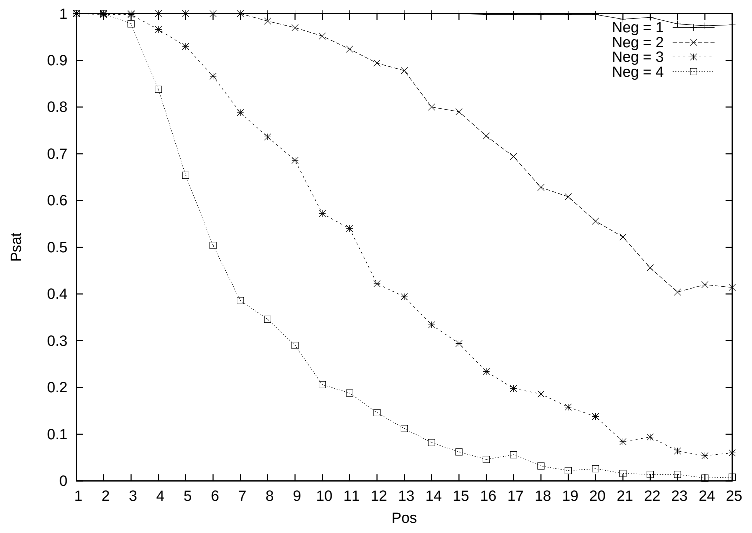

FIG. 7 – Probability of satisfiability according to the number of positive examples with $n = 6$, for $Neg = 1, 2, 3, 4$

exhibited further up the polynomial hierarchy. We show that this conjecture holds true with the bounded ILP consistency problem, a Σ_2-complete problem, representative of RL problems. We propose a first simple random generator that exhibits a phase transition in the problem's 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. As a follow-up, we plan to study the impact of the other RLPG's parameters on the generation of hard instances and to study the behaviour of the different solvers proposed in RL on those instances.

Acknowledgment

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

Références

AHR M., BIEHL M. & SCHLOESSER E. (1999). Weight decay induced phase transitions in multilayer neural networks. *J. Phys. A : Math. Gen.*, p. 5003–5008.

ALPHONSE E. & OSMANI A. (2008). A model to study phase transition and plateaus in relational learning. In *ILP '08 : Proceedings of the 18th international conference on Inductive Logic Programming*, p. 6–23, Berlin, Heidelberg : Springer-Verlag.

BASKIOTIS N. & SEBAG M. (2004). C4.5 competence map : a phase transition-inspired approach. In *Proc. of International Conference on Machine Learning*.

BOTTA M., GIORDANA A., SAITTA L. & SEBAG M. (2003). Relational learning as search in a critical region. *Journal of Machine Learning Research*, **4**, 431–463.

BYLANDER T. (1996). A probabilistic analysis of propositional strips planning. *Artificial Intelligence*, **81**(1-2), 241–271.

CARLA GOMES, HENY KAUTZ A. S. & SELMAN B. (2007). Satisfiability solvers. In *Handbook of Knowledge Representation*. Elsevier.

CHEESEMAN P., KANEFSKY B. & TAYLOR W. (1991). Where the really hard problems are. In R. MYOPOULOS, JOHN ; REITER, Ed., *Proc. of the 12th International Joint Conference on Artificial Intelligence*, p. 331–340, Sydney, Australia : Morgan Kaufmann.

CHEN H. & INTERIAN Y. (2005). A model for generating random quantified boolean formulas. In L. P. KAELBLING & A. SAFFIOTTI, Eds., *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, p. 66–71 : Professional Book Center.

DAVENPORT A. (1995). A comparison of complete and incomplete algorithms in the easy and hard regions. In *Workshop on Studying and Solving Really Hard Problems, CP-95*, p. 43–51.

DOMINGOS P. (2003). Prospects and challenges for multi-relational data mining. *SIGKDD Explorations*, **5**(1), 80–83.

FÜRNKRANZ J. (1997). Pruning algorithms for rule learning. *Mach. Learn.*, **27**(2), 139–172.

GENT I. P. & WALSH T. (1999). Beyond np : the qsat phase transition. In *AAAI '99/IAAI '99 : Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference* *innovative applications of artificial intelligence*, p. 648–653, Menlo Park, CA, USA : American Association for Artificial Intelligence.

GIORDANA A. & SAITTA L. (2000). Phase transitions in learning relations. *Machine Learning*, **41**, 217–25.

GOTTLOB G. (1987). Subsumption and implication. *Information Processing Letters*, **24**(2), 109–111.

GOTTLOB G., LEONE N. & SCARCELLO F. (1997). On the complexity of some inductive logic programming problems. In N. LAVRAČ & S. DŽEROSKI, Eds., *Proc. of the 7th International Workshop on Inductive Logic Programming*, volume 1297 of *LNAI*, p. 17–32, Berlin : Springer.

HAUSSLER D. (1989). Learning conjunctive concepts in structural domains. *Machine Learning*, **4**(1), 7–40.

HOGG T. & WILLIAMS C. (1994). The hardest constraint problems : A double phase transition. *Artificial Intelligence*, **69**(1–2), 359–377.

HUBERMAN B. A. & HOGG T. (1987). Phase transitions in artificial intelligence systems. *Artif. Intell.*, **33**(2), 155–171.

KEARNS M. J. & VAZIRANI U. V. (1994). *An Introduction to Computational Learning Theory*. Cambridge, Massachusetts : The MIT Press.

KIETZ J.-U. (1993). A comparative study of structural most specific generalisations used in machine learning. In *Proc. Third Workshop on ILP*, p. 149–164.

MITCHELL T. M. (1982). Generalization as search. *Artificial Intelligence*, **18**, 203–226.

NEWELL A. & SIMON H. A. (1972). *Human Problem Solving*. Englewood Cliffs, New Jersey : Prentice-Hall.

PAGE D. & SRINIVASAN A. (2003). Ilp : a short look back and a longer look forward. *J. Mach. Learn. Res.*, **4**, 415–430.

PLOTKIN G. (1970). A note on inductive generalization. In *Machine Intelligence*, p. 153–163. Edinburgh University Press.

RÜCKERT U., KRAMER S. & RAEDT L. D. (2002). Phase transitions and stochastic local search in k-term dnf learning. In *Proceedings of the 13th European Conference on Machine Learning*, p. 405–417, London, UK : Springer-Verlag.