A Model to Study Phase Transition and Plateaus in Relational Learning

Erick Alphonse and Aomar Osmani

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

Abstract. The feasibility of symbolic learning strongly relies on the efficiency of heuristic search in the hypothesis space. However, recent works in relational learning claimed that the phase transition phenomenon which may occur in the subsumption test during search acts as a plateau for the heuristic search, strongly hindering its efficiency. We further develop this point by proposing a learning problem generator where it is shown that top-down and bottom-up learning strategies face a plateau during search before reaching a solution. This property is ensured by the underlying CSP generator, the RB model, that we use to exhibit a phase transition of the subsumption test. In this model, the size of the current hypothesis maintained by the learner is an order parameter of the phase transition and, as it is also the control parameter of heuristic search, the learner has to face a plateau during the problem resolution. One advantage of this model is that small relational learning problems with interesting properties can be constructed and therefore can serve as a benchmark model for complete search algorithms used in learning. We use the generator to study complete informed and non-informed search algorithms for relational learning and compare their behaviour when facing a phase transition of the subsumption test. We show that this generator exhibits the pathological case where informed learners degenerate into non-informed ones.

1 Introduction

According to [Mit82], symbolic learning is defined as search: given a hypothesis space defined a priori, identified by its representation language, find a hypothesis consistent with the learning data. This paper, relating symbolic learning to search in a space state, has enabled machine learning to integrate techniques from problem solving, operational research and combinatorics. The search is NP-complete for a large variety of languages of interest (e.g. [Hau89, KV94]) and heuristic search is crucial for efficiency. Whereas heuristic search has been showed to be effective in attribute-value languages, it appeared early that learning in relational languages, also known as Inductive Logic Programming (ILP), had to face important *plateau phenomena* (see e.g. [Qui91, SP91, RM92]): the evaluation function, used to prioritise nodes in the refinement graph is constant in parts of the search space, and the search goes blind. These plateau phenomena are the pathological case of heuristic search, being complete or not [Pea85].

F. Železný and N. Lavrač (Eds.): ILP 2008, LNAI 5194, pp. 6–23, 2008.

[SGS01, Alp04, AO08] pointed out that an explanation for these plateaus is the phase transition behaviour of the NP-complete subsumption test, as shown by [GBS99, GS00]. When one studies the probability of covering a random example of a fixed size by a hypothesis given the hypothesis' size, one distinguishes three well-identified regions: a under-constrained region for small hypothesis size, named "yes", where the probability of covering an example is close to 1, an over-constrained region for large hypothesis size, named "no", where the probability is close to 0, and finally in between the phase transition or the "pt" region, where an example may or may not be covered. As the heuristic value of a hypothesis depends on the number of examples covered (positive or negative), we see that the two regions "yes" and "no" represent plateaus that need to be crossed during search without an informative heuristic value.

We think that a systematic study of the impact of plateaus on heuristic search used in learning is a necessary step for the development of scaling-up relational learners, much in the line of recent advances in combinatorics through the phase transition framework.

In this paper, we propose a consistency problem generator in relational learning, RLPG, that guarantees the existence of plateaus during search, based on model RB proposed for CSP. Using its properties, it is proved that the current hypothesis' size evaluated during learning is an order parameter of the phase transition of the subsumption test. This result asymptotically guarantees the existence of a plateau for the heuristic search. Moreover, it is shown that the size of the plateau grows sub-quadratically with the problem size. In practice, we will show that problems of very small size can be generated while still guaranteeing plateaus, which makes it suitable as a benchmark model for relational learning. This is empirically validated by running several complete search learners on problems generated by RLPG that exhibit the pathological case where informed search learners degenerate into non-informed ones.

In section 2, we present the necessary background on relational learning and Constraint Satisfaction Problems (CSP). In the next section, we discuss another model proposed to import the phase transition framework into relational learning [BGSS03]. However, they tackle the different problem of studying the link between the localisation of the target concept with respect to the phase transition and the generalisation performance on a test set. This model cannot be lifted to our problem as we will discuss it. The section 4 presents the model RLPG (Relational Learning Problem Generator). Then, this generator is empirically validated in section 5 on several complete search strategies for learning, available in the learning systems Aleph [Sri99], Progol [Mug95] and Propal [AR06]. Finally, we conclude on further developments of the model RLPG.

2 Background

2.1 Relational Learning (RL)

In this article, we study what has been termed the ILP-consistency problem for function-free Horn clauses by [GLS97]. 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 non-recursive function-free Horn clause h with no more than k literals such that h logically implies each element in E^+ and h does not implies any element in E^-.

In such hypothesis space, the logical implication is equivalent to θ-subsumption which is NP-complete and therefore decidable [Got87].

Definition 1 (θ-subsumption). *Let C, D two clauses. C θ-subsumes D, noted $C \geq_\theta D$ iff there exists a substitution θ such that $C\theta \subseteq D$*

The consistency problem is fundamental in learning as it is the core of the Statistical Learning Theory, notably studied in the PAC framework (see [KV94] for details). This *a fortiori* is true in Relational Learning where almost all noise-resistant learners are relaxation of this problem [Für97].

A central idea in symbolic learning is the use of a generality partial order between hypotheses, in the hypotheses space denoted $\mathcal{L}_h$, to guide the resolution of the consistency problem (see Mitchell [Mit82] for more details). Mitchell defined top-down search and bottom-up search strategies. Without loss of generality, we restrict ourselves to top-down search. The search strategies are further refined into generate-and-test (GT) and data-driven (DD) strategies. In the GT paradigm, the top-down refinement operator, noted ρ, is only based on the structure of the hypothesis space, independently of the learning data: Let $h \in \mathcal{L}_h : \rho(h) = \{h' \in \mathcal{L}_h | h \geq h'\}$. Therefore, generate-and-test algorithms have to deal with many refinements that are not relevant with respect to the discrimination task. On the contrary, the DD strategy searches the space of hypotheses that are more general than or equal to a given positive example and uses negative examples to prune irrelevant branches in the refinement graph. It is defined as a binary operator: Let $h \in \mathcal{L}_h, e^- \in E^- : \rho(h, e^-) = \{h' \in \mathcal{L}_h | h \geq h' \text{ and } h' \not\geq e^-\}$. Relying on the negative examples allows a TDD strategy to have a branching factor that is smaller than the branching factor of a generate-and-test strategy, and can therefore compensate for a poor evaluation function by using the learning data [AR06, AO08]. The so-called *near-misses* are negative examples that reduce the branching factor to one.

2.2 Constraint Satisfaction Problems and Random Problem Generators

A Constraint Satisfaction Problem (CSP) is defined by a finite set of variables $\{X_1, \ldots, X_n\}$, a set of finite domains $\{D_1, \ldots, D_n\}$, each variable X_i taking its value from its corresponding domain D_i, and a set of constraints $\{C_1, C_2, C_3, \ldots, C_m\}$. Each constraint C_i is defined over a subset of k variables called its scope and denoted by $scope(C_i)$. An extensional definition of a constraint C_i is the set of tuples of values allowed for the variables in $scope(C_i)$. Instantiating a variable is affecting to it a value from its domain. A solution of a CSP is an assignment of all variables that satisfies all constraints. When $(\forall i)$ $|scope(C_i)| = 2$, the CSP is called binary.

Studies on CSP assume a model of random instance generation [HW94, SD96]. Randomly generated CSP have widely been used experimentally and theoretically to study the phase transition between the regions of under-constrained and over-constrained CSP. Most studies use one of the known models A, B, C, D (see [Smi01] for details). In each of these models, sets of randomly generated CSP were used. Each set of problems is characterised by four parameters [SD96]: a set of n variables; the number of values d in each variable domain; p_1, the proportion between the number of generated constraints and the number of possible constraints, defining the CSP density, and p_2, the proportion between the number of incompatibles tuples and the possible ones in each constraint, defining the constraint's tightness. p_1 and p_2 are order parameters to exhibit phase transitions in CSP. For instance, by fixing one of the order parameters and varying the other one from 0 to 1, one wanders from an under-constraint region, the "yes" region, where the probability of solubility is close to 1, to an over-constrained region, the "no" region, where the probability is close to 0, with the phase transition localisation depending on the parameters' value.

We detail the standard stochastic model B [SD96] and its extension model RB [XL00, XBHL07] we are going to use in the following. Model B is defined by the tuple $B(k, n, d, p_1, p_2)$, where $k \geq 2$ denotes the arity of each constraint, $n \geq 2$ the number of variables, d the domain size for all constraints, p_1 constraint density and p_2 constraint tightness. We note that in model B, the number of constraints is $m = p_1.\binom{n}{k}$, the number of disallowed tuples of each constraint is $t = p_2.d^k$. Some limitations of model B regarding the asymptotic complexity have been pointed out by [AKK$^+$97]. They prove that random problems generated with B model suffer from trivial insoluble instances as problem size increases. Model RB, which share the same generation procedure as model B, avoids its limitations by adding constraints of the parameters' values. Model RB is denoted by $RB(k, n, \alpha, r, p)$, where k, n, p are respectively the same as k, n, p_2 in model B, α defines the domain size $d = n^\alpha$ and r defines the number of constraints $m = r.n.ln(n)$.

To generate a problem in each model, we have to build m constraints, each one formed by randomly selecting, uniformly and without replacement, a scope of k (distinct) variables and randomly selecting, uniformly and without replacement, a relation of t distinct disallowed tuples. [XL00] prove that model RB, under some conditions, avoids trivial asymptotic behaviours and provides exact phase transition thresholds for random CSP. Model RB guarantees the phase transition by varying one of the two defined parameters r and p. Note that the main difference between B and RB is that the domain size of each variable in RB grows polynomially with the number of variables. We will use the following theorem to show that the hypothesis size in relational learning is an order parameter of the phase transition.

Theorem 1 ([XL00]). *Let P_{sat} denotes a probability distribution, if k, $\alpha > \frac{1}{k}$ and $p \leq \frac{k-1}{k}$ be constants and $r_{cr} = -\alpha/ln(1-p)$ then*

$$\lim_{n\to\infty} P_{sat}[P \in RB(k, n, \alpha, r, p) \text{ is sat}] = \begin{cases} 1 \text{ if } r < r_{cr} \\ 0 \text{ if } r > r_{cr} \end{cases}$$

This theorem indicates that a phase transition is guaranteed when the domain is not too small and the constraint tightness not too large. In the case of binary constraints ($k = 2$), the domain size is required to be greater that the squared root of the number of variables.

2.3 Reduction of Extensional CSP to θ-Subsumption

As θ-subsumption is NP-complete, various models used to study phase transition phenomena from other NP-complete problems can be imported to relational learning via reduction. Models for random CSP are easy to import in relational learning because of their trivial reduction to the subsumption problem (time and space complexity linear in the problem size), that is to decide if a variabilised function-free horn clause, C, θ-subsumes a ground function-free horn clause, D.

Following the presentation of CSP in section 2.2, C encodes the scope of the m constraints with m literals built on different predicate symbols : each constraint C_i is associated with a literal l_i such that $scope(C_i) = vars(l_i)$. By definition, C cannot have more literals than $\binom{n}{k}$. The extensional definitions of the constraints is given by D : for each constraint we define as many ground literals as there are allowed tuples in it, built from its associated predicate symbol. The size of D is then $\Sigma_i^m |C_i|$, with $|C_i|$ the cardinality of the set of allowed tuples by C_i.

We illustrate the reduction on an example. Let be a CSP defined over 3 variables with $D_1 = D_2 = \{a, b\}$ and $D_3 = \{a, b, c\}$, and 2 constraints $C_1(X_1, X_2) = \{(a, b)\}$ and $C_2(X_1, X_2, X_3) = \{(a, b, c), (b, a, a)\}$. We define C as:
$c \leftarrow c1(X_1, X_2), c2(X_1, X_2, X_3)$ and D as $c \leftarrow c1(a, b), c2(a, b, c), c2(b, a, a)$. Note that the positive literal is only relevant for the learning task as it doesn't encode anything CSP specific. It is easy to see that a substitution θ, solution to the subsumption problem, is the solution tuple of the CSP and conversely. In the example $\theta = \{X_1/a, X_2/b, X_3/c\}$.

3 Related Work

A model to study the phase transition of the subsumption test has been proposed in [GS00] and the study of its possible impact on relational learning efficiency has been proposed in [GBS99, BGSS03]. We are going to discuss them and show their limitations to study the impact of the phase transition of the subsumption test on plateaus during search.

3.1 Model to Study the Phase Transition of the Subsumption Test

In [GS00], they propose a model, inspired by model B, to study the phase transition of the subsumption test. Hypotheses are function-free horn clauses from the hypothesis space $\mathcal{L}_h^m$ built as follow:

$$\mathcal{L}_h^m = \{c \leftarrow \bigwedge_{k=1}^{n-1} p_{l_k}(X_k, X_{k+1}) \wedge \bigwedge_{k=n}^{m} p_{l_k}(X_{i_k}, X_{j_k})\}$$

where c is the clause head without variables, $i_k < j_k \in \{1, \ldots, n\}, l_k \in \{1, \ldots, m\}$ and such that all literals in the clause body are built on distinct binary predicate symbols. The first $n-1$ literals ensure that all variables are linked, and therefore, that the set of variables cannot be decomposed into sets of independent variables that can break the subsumption test into easier sub-problems [GS00]. That is to say, p_1, the constraint density, can not be fewer than $p_{1min} = 2/n$. Examples are represented as ground clauses as described earlier to encode the constraints' domains. They showed that m, the hypothesis size, and L, the domain size, were order parameters of the phase transition in their settings. However, there is no guarantee that a phase transition will occur as their model differs from random CSP models. p_1 and p_2, the order parameters in most CSP models, are random variables here, depending on m and L. Indeed, the variables pairs are randomly drawn with replacement and several literals can be built on the same pair of variables. The constraint on the variables is no longer a set of tuples built on a unique predicate symbol, but the intersection of all sets of tuples related to the literals. For instance, let $n = 4$, then the maximal number of constraints in a binary CSP is $m_{max} = 6$. Let $m = m_{max}$ and the two following hypotheses:

$$h_1 : c \leftarrow p_1(X, Y), p_2(Y, Z), p_3(Z, T), p_4(X, Y), p_5(X, Y), p_6(X, Y)$$

$$h_2 : c \leftarrow p_1(X, Y), p_2(Y, Z), p_3(Z, T), p_4(X, Z), p_5(X, T), p_6(Y, T)$$

We remark that in h_1, the order parameter $p_1 = p_{1min}$ and $p2$ is undefined, and in h_2, the order parameter $p_1 = 1$ and $p_2 = 1 - N/L^2$.
To exhibit the phase transition of subsumption, we propose to use model RB, where phase transition is proved to occur asymptotically and can be precisely located.

3.2 Phase Transition of Subsumption and Relational Learning

Bringing the phase transition framework to the realm of Relational Learning has been first done by Giordana et al. [GBS99, BGSS03] where they proposed to study the link between the localisation of the target concept with respect to the phase transition of the subsumption test and the generalisation performance on a test set. They tackle the learning of a function-free horn clause from the hypothesis space $\mathcal{L}_h^m$ described above. A learning problem is parametrised with the pair (m, L), the number of variables n being fixed to 4 and the number of allowed tuples N is fixed to 100 in their experiments. In a (m, L) problem, m is the size of the target concept drawn from $\mathcal{L}_h^m$ and L the number of constants in the examples. For each problem, a learning set and a test set are built to evaluate generalisation performance of learning algorithms. Both are balanced sets of examples with 100 positive examples and 100 negative examples. It has to be noted that if (m, L) lies in the "yes" (resp. "no") region, by construction the concept description will almost surely cover (resp. reject) any randomly constructed example. For those problems, the example generator is modified and relies on a repair mechanism to ensure a balanced distribution of positive and negative examples [BGSS03].

It was shown in [AO08], however, that this localisation of the concept was not a reliable indication of the learning problem difficulty, and that plateaus generated by the phase transition behaviour of the subsumption test prevented FOIL from solving any problem.

Their model cannot be lifted to our study of the link between plateaus and heuristic search efficiency. Notably, we can note that the hypothesis space is very large in their settings and prohibits the study of average heuristic search behaviour of complete learners and even incomplete learners. There is also no guarantee that plateaus, through the occurrence of a phase transition, will occur with smaller parameter values, as their model differs from random CSP models as stated above, but also as a balanced distribution of the examples is ensured by a repair mechanism and is no longer random.
Also, the proposed problem generator does not translate into lattice-like hypothesis space (the target concept being one of the most specific elements) and the link between variables is a hidden structure. This point is important to be able to easily run standard learning approaches on generated problems. For instance, this prevents the use of popular approaches based on the existence of a bottom-most element in the search space, like top-down seed-based approaches (Aleph, Progol, Propal) and bottom-up approaches, like lgg-based approaches [Plo70].

We introduce in the next section a model to analyse phase transition and plateau phenomena in relational learning. It guarantees plateaus during search in problems of very small size, suitable to evaluate the average performance of learners, and defines the hypothesis space as a boolean lattice to ease the implementation of various learning strategies.

4 Model for Exhibiting Plateaus in Random RL Problems

A learning problem instance in our model is denoted $RLPG(k, n, \alpha, N, Pos, Neg)$. The parameters k, n, α are the same parameters as in RB. N is defined as in [GBS99] as the number of allowed tuples by each constraint and we have $N = (1-p).d^k$, with p the constraint tightness. As they argued, this is more meaningful for learning as it is the number of literals built on the predicate symbol associated to a given constraint. Pos and Neg are the number of positive and negative examples in the learning dataset, respectively.

Given k and n, the maximum number of constraints is $\binom{n}{k}$. All these constraints are encoded in a clause which is set as the bottom clause of the hypothesis space $\mathcal{L}_h$. $\mathcal{L}_h$ is then defined as the power set of the bottom clause, which is isomorphic to a boolean lattice. As said previously, this property is interesting for learning because it eases the implementation of various learning strategies like bottom-up generate-and-test and data-driven (i.e. lgg based [Plo70]) strategies. Also, this restriction is often used in top-down learning systems like Aleph, Progol or Propal to define complete and efficient refinement operators. In that space, it is guaranteed that each hypothesis evaluated by the learning algorithms encodes a valid constraint network of the underlying model RB. The refinement operator is to add a literal from the bottom clause that is not in the hypothesis,

hence the number of literals in the hypothesis is exactly m, the number of constraints in the underlying CSP.

Learning examples are randomly drawn, independently and identically distributed, given n, α and N, as explained in section 2.2. Their size is $N.\binom{n}{k}$. Each example defines the set of allowed tuples of size N for possible constraint networks ranging from 0 to $\binom{n}{k}$ constraints.

In the next section, we detail how this model exhibits a phase transition of the subsumption test when varying hypothesis sizes, and in section 4.2 how this phase transition translates into a plateau for the heuristic search, during the resolution of the consistency problem. From now on, we restrict ourselves to binary CSP, that is all logical predicates are binary. As in [BGSS03], this restriction is for the sake of simplicity while still being representative of typical relational learning problems.

4.1 The Phase Transition of the Subsumption Problem

Given a randomly drawn example according to model RLPG, a hypothesis of size m defines m constraints over n variables, each constraint being extensionally defined in the example. As the hypothesis size m varies during search from 1 to $n(n-1)/2$ ($k = 2$ here), $r = m/(nln(n))$, the order parameter of the phase transition, varies (or equivalently in model B, $p1$ varies from 0 to 1). In model RB, there exists an exact localisation of the phase transition for $r = r_{cr}$. As we use the same model, varying m, the hypothesis size of the current hypothesis asymptotically exhibits a phase transition.

[XL00] give an asymptotic value of the cross-over point of the phase transition as $r_{cr} = -\alpha/ln(1-p)$ (theorem 1). This critical value is the point where the expected number of solutions of the problem $E(N) = 1$. In practice, this is often

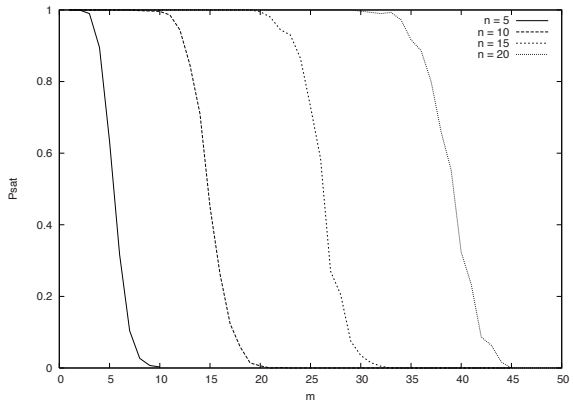

Fig. 1. P_{sat} drawn for different values of n ($d = n^{\alpha}$ with $\alpha = 1$, $p = 0.75$) as m varies, averaged over 300 drawn (hypothesis,example) pairs

used to localise the critical value of the order parameter where $P_{sat} = 0.5$ (see e.g. [SD96]). In our case, the critical hypothesis size is $m_{cr} = -\frac{\alpha n ln(n)}{ln(1-p)}$. We can see that as n increases, the value of m_{cr} grows in $nln(n)$. Interestingly, even when n is small (starting from 5), experiments corroborate the theoretical results as shown in figure 1. Note that for each plot, the maximum number of possible literals is $n(n-1)/2$.

Empirical validation of the localisation of the crossover point regarding to the order parameter m is summarised in table 1. We observe that for all values of n, greater than 5, the empirical value of the crossover point m_{cr} is close to the theoretical asymptotic value. Even when the conditions of theorem 1 are not respected ($p = 0.75$ in Figure 1), we observe that $P_{sat} = 1$ almost surely for $m < m_{cr}$. It was noted already in [Smi01] that the conditions attached to α and p are strong and it should be possible to relax them by using other methods to compute P_{sat} lower bounds. This robustness allows us to define small problems where the phase transition is exhibited, which translates into small hypothesis spaces as we will see in the next section.

Table 1. Comparison between empirical and theoretical values of m for $\alpha = 1, p = .75$

n	m_{TP}	empirical value of m
5	5,79 ± 1	between 5 and 6
10	16,61 ± 1	between 14 and 15
15	29,30 ± 1	between 26 and 27
20	43,21 ± 1	between 39 and 40

We show the phase transition along the second order parameter p of model RB in figure 2, with $\alpha = 1.4$ where the contour plots correspond to $P_{sat} = 0.99, 0.5$ and 0.01, and in figure 3, where the contour plot corresponds to $P_{sat} = 0.5$, with different values of α. Each point is averaged over 1000 subsumption tests. p is controlled with N in RLPG and we observe that as N increases, p decreases which gives smaller values for m_{cr}.

p can also be controlled by varying α, keeping N constant, as in [GS00] who used the domain size as order parameter, which is pictured in figure 4. We are going to use this control parameter in the next section to change the localisation of the "pt" region and therefore to change the plateau length of a problem in the next section. Although this is not strictly model RB, the advantage of $d = n^{\alpha}$ here to control p instead of N is that the examples' size does not change and it keeps learning problem size constant. However, p changes faster, quadratically in d instead of linearly in N, as it can be seen in the figure.

4.2 The Plateaus of Heuristic Search

A relational learning problem is defined by drawing, independently and identically distributed, Pos positive examples and Neg negative examples. By construction, a hypothesis, solution of the consistency problem, can only be found

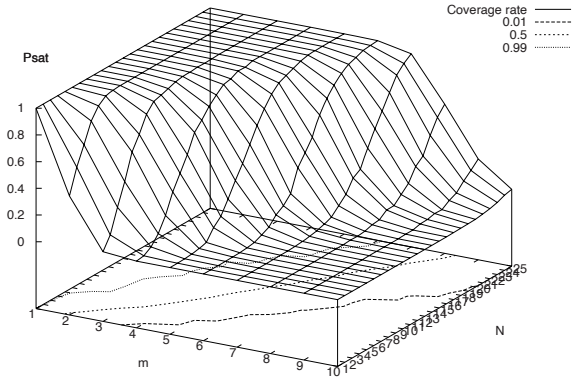

Fig. 2. Probability of solubility of the subsumption test in the (m, N) plane for $n = 5$ and $\alpha = 1.4$

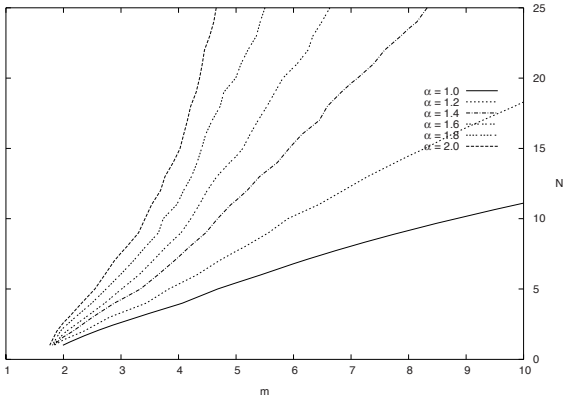

Fig. 3. Contour plot corresponding to $P_{sat} = 0.5$ in the (m, N) plane for $n = 5$ and various values of α

in the phase transition region, as hypotheses in the "yes" (resp. the "no") will almost surely subsume (resp. not subsume) all learning examples. The top-down search, which starts from the top-most hypothesis in the hypothesis space lattice, will have to specialise hypotheses by adding a literal at a time to cross the "yes" region before reaching the phase transition region where a solution can be expected. Dually, a bottom-up search will have to cross the "no" region before reaching the "pt" region.

The "yes" and "no" regions implies a plateau to cross during a heuristic search in these regions. Indeed, if we study the state of the art on evaluation functions used in learning (see for instance [FF03]), it shows that all of them are based, without loss of generality, on three parameters that are the coverage rate of positive examples, the coverage rate of negative examples and possibly

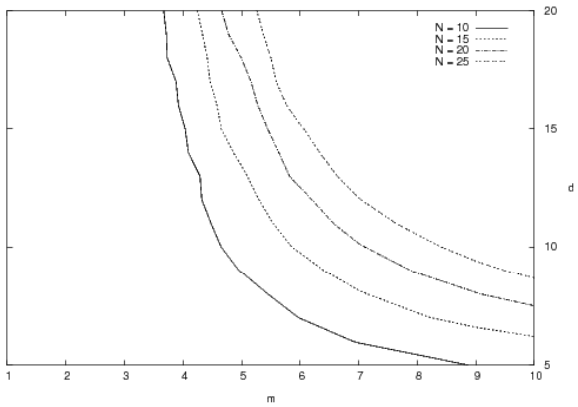

Fig. 4. Contour plot corresponding to $P_{sat} = 0.5$ in the (m, d) plane for $n = 5$ and various values of N

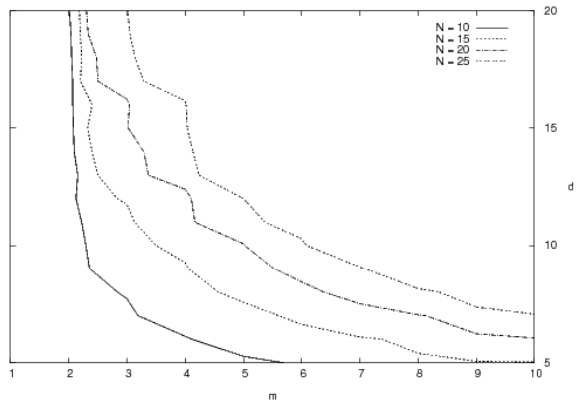

Fig. 5. Contour plot corresponding to plateaus size: $P_{sat} \geq 0.99$ in the (m, d) plane for n=5 and various values of N

a complexity measure of the hypothesis under consideration. In the "yes" or "no" regions, the coverage rate of positive and negative examples is constant and implies that all evaluation functions are constant, defining a plateau. It has to be noted that, as the first two parameters are inherited from the learning task definition, it is unlikely that a solution for solving the plateau problem consists in designing new evaluations functions.

We show the plateaus for a top-down approach in figure 5. It shows the contour plot corresponding to $P_{sat} = 0.99$[1] for problems defined with $n = 5$ variables and different values of N. P_{sat} is evaluated on average at each point in the (m, d) plan

[1] Any value arbitrarily closer to 1.0 gives the same result.

by running 1000 subsumption tests. These contour plots indicate the beginning of the "pt" region to the right, the left part, for low m, being the plateau in the "yes" region. We see that varying d the number of constants in the example, we change the tightness of the constraints which varies the plateau length.

Various experiments have been conducted with different parameter values which show all similar plateau profiles. It is interesting to note that the smallest value for n where plateaus were exhibited is $n = 5$. In this case, the bottom-most hypothesis in the lattice has a size of $n(n-1)/2 = 10$, and therefore the plateaus are exhibited for a hypothesis space of 2^{10} hypotheses only. This is a very small problem size which makes it very useful for studying average performances of complete search learners in reasonable time. We are going to use it in the next section to compare the behaviours of different complete informed and non-informed search learners.

5 Experimental Results

Complete search learners, available in the learning systems Aleph, Progol and Propal are run on a collection of problems denoted by $RLPG(2, 5, d, 15, Pos, Neg)$, with d varying from 5 to 20, and $Pos = Neg$ varying from 1 to 5. We evaluate the impact of the plateaus on their heuristic search cost by recording the number of evaluated hypotheses to answer the consistency problem. Every plot is averaged over 1000 randomly drawn learning problems. As said previously, we limit ourselves to top-down approaches. For a detailed description of the various strategies we discuss below, we refer to [Pea85, Sri99, Mug95, AR06] because of the space requirements of the paper.

As non-informed searches, we use the breadth-first TGT search (BF-TGT) and the depth-first TGT search (DF-TGT). As informed searches, we use the A TGT search (A-TGT) and the best-first TGT search (BESTF-TGT). Informed

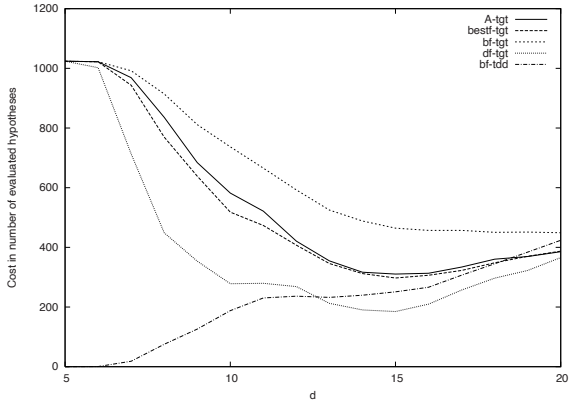

Fig. 6. Median number of evaluated hypotheses during search for learning problems with Pos = Neg = 3 and d varying from 5 to 20

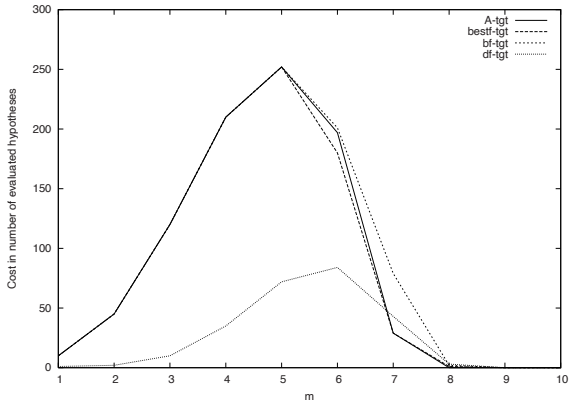

Fig. 7. Median number of evaluated hypotheses during search according to their size, with $d = 8$

search makes use of an evaluation function to minimise, whose general form is $f = g + h$. g is defined as the cost from the start to the current hypothesis and h as an estimation of the distance from the current hypothesis to the goal. We define A-TGT according to the Progol system: g is defined as the length of the current hypothesis and h has the difference between the number of negative examples and the number of positive examples. In our context, as all positive examples must be subsumed, it simplifies to the number of negative examples. BESTF-TGT is not biased towards shorter hypotheses and defines $g = 0$.

In all these complete strategies, the implemented refinement operator structures the hypothesis space as a tree in order to avoid redundancy during search. This is done classically with boolean lattices by fixing a total order on the possible refinements.

The last learning strategy we study is the one used in the TDD learner Propal. This is an incomplete learner as it performs a beam search guided by the Laplace function. So we set Propal with a beam of unlimited size, which basically turns down to a non-informed breadth-first search (BF-TDD). The only difference is that when the solution is reached at a level of the search, it will be the first picked up at the next level. Note also that, as an incomplete learner, it does not have an optimal refinement operator, like the other learners, and may evaluate the same hypothesis several times.

We only show results obtained when $Pos = Neg = 1, 3$ and 5, as extensive experiments varying the number of examples exhibited the same patterns. Figure 6 shows the median cost of the different learning strategies for $Pos = Neg = 3$ with various plateau lengths defined by d (as illustrated in figure 5). We can notice several patterns for the learners depending on the plateau lengths.

We first compare BF-TGT and BF-TDD. BF-TDD outperforms its generate-and-test counter-part, as it has been shown in [AR06] that the TDD approach necessarily has a branching factor smaller or equal to that of a TGT strategy.

When $d = 5$, the plateau is the largest and no consistent hypothesis exists. BF-TGT has to evaluate all the hypotheses (1024) before answering the problem. As d increases, the plateau length decreases and BF-TGT will develop all hypotheses up to the phase transition region where a hypothesis can be found. As for BF-TDD, the computation of a near-miss with the bottom of the search space in the case of $d = 5$ yields an inconsistency as the near-miss is equal to the bottom-most hypothesis and the search halts with no evaluated hypotheses. As d increases, near-misses will be farther from the bottom of the search space and the branching factor will increase. This is pictured by an increasing number of evaluated hypotheses, which converge towards the cost of BF-TGT. The other non-informed learner, DF-TGT, shows the same behaviour as BF-TGT for low values of d as it cannot detect trivial inconsistency. When d is above 13, it performs best as there are several solutions in the "pt" region. DF-TGT directly crosses the plateau and ends up doing few backtracks before finding a consistent hypothesis.

We discuss now the informed strategies. We can see that they both have a similar behaviour, and mimic BF-TGT, although they evaluate fewer hypotheses than it for smaller plateaus. However, they systematically evaluate more hypotheses than DF-TGT, and than BF-TDD, except for the largest values of d. This behaviour is the pathological case of an informed search. As an illustration, we plot the number of hypotheses evaluated according to their size, for $d = 8$ in figure 7, where the plateau is large, and for $d = 15$ in figure 8, where the plateau is smaller. We can see that they all develop all hypotheses up to $m = 5$ for $d = 8$, given a plateau size of 4 in figure 5, and up to $m = 3$ for $d = 15$, given a plateau size of 2. It is only in the "pt" region that the heuristic becomes useful to guide the search and differentiates the different approaches. The fact that each time m is one literal bigger than the plateau size is not clear. We can note that BESTF-TGT systematically outperforms A-TGT, even slightly, as A-TGT will

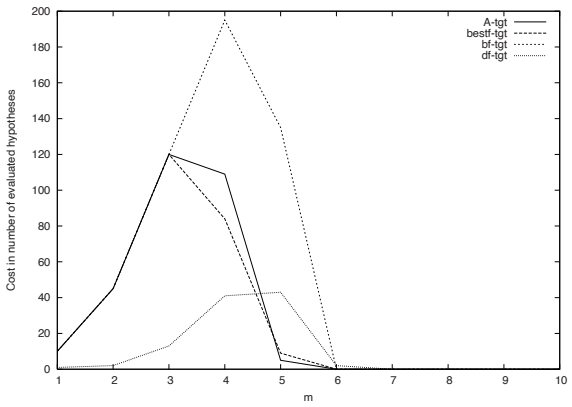

Fig. 8. Median number of evaluated hypotheses during search according to their size, with $d = 15$

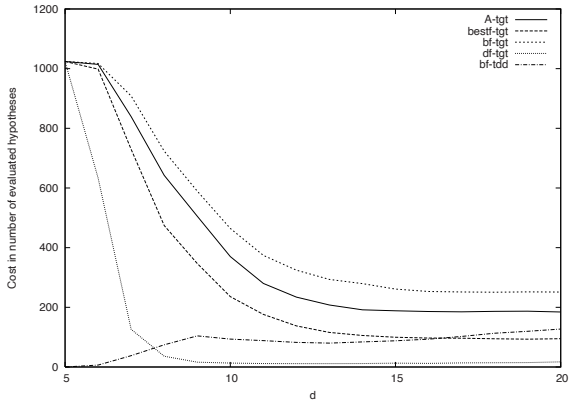

Fig. 9. Median number of evaluated hypotheses during search for learning problems with Pos = Neg = 1 and d varying from 5 to 20

give preference to a hypothesis that subsumes 2 negative examples compared to a hypothesis one literal longer that subsumes only 1 negative example.

We can note that the cost of resolution, after decreasing up to $d = 15$ for the GT learners, starts increasing after this point. The problems seem harder for BF-TDD too. This point corresponds to problems that have a probability of solubility close to 0.5 and would correspond to the phase transition region of the consistency problem. As noted in [AO08], learners solving the ILP-consistency problem potentially have to face two phase transitions: the phase transition of the NP-complete subsumption test and the phase transition of NP-complete search. It is a very interesting follow-up to study how the parameters of RLPG can exhibit this second phase transition as the probability of solubility of the

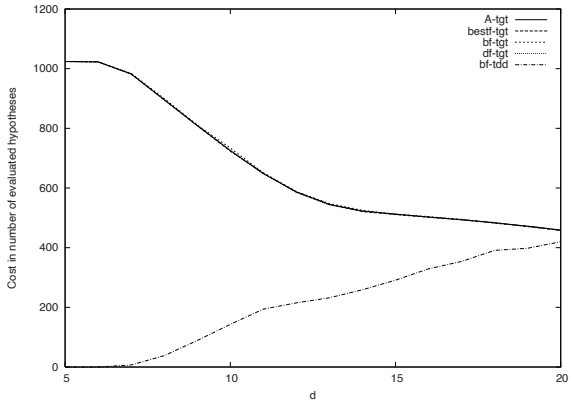

Fig. 10. Median number of evaluated hypotheses during search for learning problems with Pos = Neg = 5 and d varying from 5 to 20

consistency problem may impact the behaviour of learning strategies. Figure 9 shows problems generated with $Pos = Neg = 1$, where the number of solutions is high. In this case, we observe the same behaviour of algorithms as in figure 6, with a notable difference for DF-TGT which quickly reaches the "pt" region and finds a solution with almost no backtrack.

Figure 10 shows the median cost of the different learning strategies in problems with $Pos = Neg = 5$ where almost no solution exists and all learners have to search the entire space. We observe that all informed and non-informed TGT approaches have the same high median cost as they cannot efficiently detect inconsistency as opposed to the TDD approach for all values of d.

6 Conclusion

Recent works in relational learning pointed out that the phase transition phenomenon, which may occur in the subsumption test during search, acts as a plateau for the heuristic search, strongly hindering its efficiency [AO08]. We proposed to systematically investigate this issue by designing a relational learning problem generator where it is shown that top-down and bottom-up learning strategies face a plateau during search before reaching a solution. This property is ensured by using the generative model RB defined for CSP to exhibit the phase transition of the subsumption test, with the hypothesis size as an order parameter. The size of the plateau grows sub-quadratically with the problem size and it is guaranteed asymptotically. Intensive experiments show that even for small problems the asymptotic model of RLPG still holds. This feature allows to study a wide range of algorithms in reasonable time and is therefore suitable as a benchmark model. At the end of the paper, we have presented preliminary results with various complete learners and interesting behaviours have been pointed out. Notably, it has been shown, as a validation, that RLPG exhibited the pathological case where informed search degenerates into non-informed one when facing plateaus.

Finally, this model points out interesting follow-ups. We plan to further study the properties of generated problems depending on RLPG's parameters; implement and compare other learning strategies, notably bottom-up, to exhibit characteristic behaviours to help design better heuristic approaches for relation learning.

References

[AKK+97] Achlioptas, D., Kirousis, L.M., Kranakis, E., Krizanc, D., Molloy, M.S.O., Stamatiou, Y.C.: Random constraint satisfaction: A more accurate picture. In: Smolka, G. (ed.) CP 1997. LNCS, vol. 1330, pp. 107–120. Springer, Heidelberg (1997)

[Alp04] Alphonse, E.: Macro-operators revisited in inductive logic programming. In: Camacho, R., King, R., Srinivasan, A. (eds.) ILP 2004. LNCS (LNAI), vol. 3194, pp. 8–25. Springer, Heidelberg (2004)

[AO08] Alphonse, E., Osmani, A.: On the connection between the phase transition of the covering test and the learning success rate. Machine Learning Journal 70(2-3), 135–150 (2008)

[AR06] Alphonse, E., Rouveirol, C.: Extension of the top-down data-driven strategy to ILP. In: Muggleton, S., Otero, R., Tamaddoni-Nezhad, A. (eds.) ILP 2006. LNCS (LNAI), vol. 4455, pp. 49–63. Springer, Heidelberg (2007)

[BGSS03] Botta, M., Giordana, A., Saitta, L., Sebag, M.: Relational learning as search in a critical region. Journal of Machine Learning Research 4, 431–463 (2003)

[FF03] Fürnkranz, J., Flach, P.: An analysis of rule evaluation metrics. In: Proc. 20th International Conference on Machine Learning, pp. 202–209. AAAI Press, Menlo Park (2003)

[Für97] Fürnkranz, J.: Pruning algorithms for rule learning. Mach. Learn. 27(2), 139–172 (1997)

[GBS99] Giordana, A., Botta, M., Saitta, L.: An experimental study of phase transitions in matching. In: Proc. of the 16th International Joint Conference on Artificial Intelligence, pp. 1198–1203

[GLS97] Gottlob, G., Leone, N., Scarcello, F.: On the complexity of some inductive logic programming problems. In: Džeroski, S., Lavrač, N. (eds.) ILP 1997. LNCS, vol. 1297, pp. 17–32. Springer, Heidelberg (1997)

[Got87] Gottlob, G.: Subsumption and implication. Information Processing Letters 24(2), 109–111 (1987)

[GS00] Giordana, A., Saitta, L.: Phase transitions in learning relations. Machine Learning 41, 217–225 (2000)

[Hau89] Haussler, D.: Learning conjunctive concepts in structural domains. Machine Learning 4(1), 7–40 (1989)

[HW94] Hogg, T., Williams, C.P.: The hardest constraint problems: A double phase transition. Artificial Intelligence 69(1–2), 359–377 (1994)

[KV94] Kearns, M.J., Vazirani, U.V.: An Introduction to Computational Learning Theory. The MIT Press, Cambridge (1994)

[Mit82] Mitchell, T.M.: Generalization as search. Artificial Intelligence 18, 203–226 (1982)

[Mug95] Muggleton, S.: Inverse entailment and PROGOL. New Generation Computing 13, 245–286 (1995)

[Pea85] Pearl, J.: Heuristics. Addison-Wesley, Reading (1985)

[Plo70] Plotkin, G.: A note on inductive generalization. In: Machine Intelligence, pp. 153–163. Edinburgh University Press (1970)

[Qui91] Quinlan, J.R.: Determining literals in inductive logic programming. In: Proc. of the 12th International Joint Conference on Artificial Intelligence, Sydney, New South Wales, Australia, pp. 746–750. Springer, Heidelberg (1991)

[RM92] Richards, B.L., Mooney, R.J.: Learning relations by pathfinding. In: Proc. of the Tenth National Conference on Artificial Intelligence, San Jose, California, pp. 723–738. AAAI Press/MIT Press (1992)

[SD96] Smith, B.M., Dyer, M.E.: Locating the phase transition in binary constraint satisfaction problems. Artificial Intelligence 81(1-2), 155–181 (1996)

[SGS01] Serra, A., Giordana, A., Saitta, L.: Learning on the phase transition edge. In: Nebel, B. (ed.) Proc. of the 7th Int. Conf. on Artificial Intelligence, pp. 921–926

[Smi01] Smith, B.M.: Constructing an asymptotic phase transition in random binary constraint satisfaction problems. Theoretical Computer Science 265(1–2), 265–283 (2001)

[SP91] Silverstein, G., Pazzani, M.J.: Relational cliches: Constraining constructive induction during relational learning. In: Proc. of the 8th Int. Workshop on Machine Learning, pp. 203–207. Morgan Kaufmann, San Francisco (1991)

[Sri99] Srinivasan, A.: A learning engine for proposing hypotheses (Aleph) (1999), `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph`

[XBHL07] Xu, K., Boussemart, F., Hemery, F., Lecoutre, C.: Random constraint satisfaction: Easy generation of hard (satisfiable) instances. Artif. Intell. 171(8-9), 514–534 (2007)

[XL00] Xu, K., Li, W.: Exact phase transitions in random constraint satisfaction problems. J. Artif. Intell. Res (JAIR) 12, 93–103 (2000)