Relational Learning: Hard Problems and Phase Transitions

M. Botta[(1)], A. Giordana[(2)], L. Saitta[(2)], M. Sebag[(3)]

(1) Dip. di Informatica
Università di Torino
Corso Svizzera 185
10149 Torino, Italy
botta@di.unito.it

(2) Dip. di Scienze e Tecnologie Avanzate
Università del Piemonte Orientale "Amedeo Avogadro"
Corso Borsalino 54
15100 Alessandria, Italy
{attilio, saitta}@al.unipmn.it

(3) Laboratoire de Méchanique des Solides
Ecole Polytechnique
91128 Palaiseau, France
sebag@cmapx.polytechnique.fr

Key Words: Learning Relations, Phase Transitions, Computational Complexity

Abstract

This paper focuses on a major step of machine learning, namely checking whether an example matches a candidate hypothesis. In relational learning, matching can be viewed as a Constraint Satisfaction Problem (CSP). The complexity of the task is analyzed in the Phase Transition framework, investigating the impact on the effectiveness of well two relational learners: FOIL and G-NET.

The critical factors of complexity, and their critical values, are experimentally investigated on artificial problems. This leads to distinguish several kinds of learning domains, depending on whether the target concept lies in the "mushy" region or not. Interestingly, experiments done with FOIL and G-NET show that both learners tend to induce hypotheses generating matching problems located inside the phase transition region, even if the constructed target concept lies far outside. Moreover, target concepts constructed too close to the phase transition are hard and both learners fail. The paper offers an explanation for this fact, and proposes a typology of learning domains and their hardness.

1 Introduction

In several classes of computationally difficult problems, such as K-Satisfiability, Constraint Satisfaction problems (CSP), graph K-coloring, and the decision version of the Traveling Salesm*an problem, a phenomenon termed phase transition* occurs (see

[Hogg, Huberman & Williams (Eds.), 1996] for a comprehensive presentation). A *phase transition* consists in an abrupt change in the probability of a problem being solvable, and it is coupled with a peak in the computational complexity [Cheeseman, Kanefsky & Taylor, 1991; Crawford & Auton, 1996; Freeman, 1996; Selman & Kirkpatrick, 1996; Smith & Dyer, 1996; Williams & Hogg 1994; Prosser, 1996; Hogg, 1996; Gent & Walsh, 1996; Zhang & Korf, 1996].

The class of Constraint Satisfaction Problems (CSP) exhibits a typical phase transition with respect to the number of constraints [Prosser, 1996; Giordana et al., 1999]. Proving true or false a conjunctive formula in a given universe (referred as *matching problem*, in the following) is a CSP, and has been proven to have exponential complexity in the worst case. On the other hand Giordana et Al. [1999] have shown that this task is characterized by typical phase transitions with respect to the number of predicates in the formula and the number of constants in the universe the proof occurs.

This paper investigates the influence on relational learning due to the of the phase transition in the matching problem. For any relational learner the induction process can be modeled as a cycle, where inductive hypotheses (logical formulas) are continuously generated and verified on a set of instances of the target relation, checking for consistency. In the verification step, every pair <inductive hypothesis, learning example> defines a matching problem. When a matching problem lies in the mushy region the complexity may be prohibitive, and the learner could be not able to terminate within the limits imposed by the available resources.

This point has been experimentally investigated using a suite of artificial problems. As described later on, every learning problem has been constructed by choosing a specific target formula and then generating a set of positive and negative examples for it, which defined matching problems in different positions with respect to the phase transition. Two learners (FOIL and G-NET) have been challenged to discover the original formula, or at least another one semantically equivalent.

The results we obtained are essentially two, both quite surprising: First, both learners tend to select definitions defining matching problems inside the mushy region. This occurs independently of the location of the original target relation. Second, when the target relation defines matching problems located too close to the mushy region, the learners get confused and produce definitions consisting of many small disjuncts, very little predictive on a test set. The findings are analyzed and some explanations are provided.

The paper is organized as follows: Section 2 summarizes the findings described by Giordana et al. [1999] showing that the matching problem presents a phase transition with respect to the number of predicates and the constants in the universe, used as order parameters. Section 3 describes the context and the goal of the experimentation, whose results are reported in Section 4. Section 5 presents a discussion of the results, whereas some concluding remarks are contained in Section 6.

2 Phase Transition in the Matching Problem

We first propose some order parameters to analyze the relational matching problem. These are used to design artificial random matching problems.

2.1 Order parameters

An instance of the matching problem is a pair $<\varphi,U>$, being φ a conjunctive formula in FOL, and U a universe. The instance is solvable if there exists at least one model of φ in U. The problem can be reformulated as a *Constraint Satisfaction Problem* (CSP): the goal of a CSP is to assign a value a_i to each variable x_i $(1 \le i \le n)$, such that every constraint $R_j(x_1, x_2, ..., x_n)$ $(1 \le j \le m)$ in a set **R** is satisfied. A constraint R_j is described as the set of all tuples $v_1, ..., v_n$, such that $R_j(v_1, v_2, ..., v_n)$ holds.

A formula φ to match can clearly be reformulated as a set of constraints (corresponding to the literals in φ), and the universe U contains all tuples satisfying the constraints (closed world assumption). We restrict ourselves to consider binary relations R_j in the following. The corresponding CSP is then termed *binary* [Williams & Hogg, 1994].

Phase transitions in binary CSP have been both experimentally and theoretically investigated [Williams & Hogg, 1994; Smith & Dyer, 1996; Prosser, 1996]. Two parameters are usually defined in order to account for the constrainedness degree of CSP: the *constraint density* p_1, defined, for binary constraints, as the fraction of constrained variable pairs among all possible pairs, and the *constraint tightness* p_2, defined as the average number of value pairs ruled out by any one of the constraints [Prosser, 1996]. Unlike the literature of the main stream, we will use other two parameters, which are related to p_1 and p_2, but are more useful in order to analyze the matching problem in relational learning.

For the sake of simplicity, we assume that all variables have the same domain of size L, and that the extensions of all relations have the same size N (number of atoms built on any given predicate symbol). The constrainedness degree of a matching problem is then studied with respect to two order parameters: The number m of constraints and the number L of constants in the universe (the size of any variable domain); a third parameter, namely the number N of atoms built on any predicate symbol, is kept constant in the present investigation.

2.2 Artificial Problem Generation

The experimental analysis reported in the following is based on artificial matching problems $<\varphi,U>$ generated as follows. Let $x_1, x_2, ..., x_n$ denote the variables, and $\alpha_1, \alpha_2, ..., \alpha_r$ denote the predicate symbols. Formula φ is generated by a two-step process. In order to guarantee the connectivity of φ, we first construct a connected formula φ_s as

$$\varphi_s(x_1, x_2, ..., x_n) = \alpha_{\sigma(1)}(x_1, x_2) \wedge ... \wedge \alpha_{\sigma(n-1)}(x_{n-1}, x_n), \qquad (1)$$

where $\alpha_{\sigma(i)}$ is uniformly selected in the predicate symbols. Formula φ is constructed from φ_s by adding random literals $\alpha_\kappa(x_i, x_j)$ until a total number of m literals is reached (assuming of course that $m \ge n$-1), such as all literals in φ are distinct.

Formulas φ, constructed in this way, contains exactly n variables and m literals, and the same pair of variables may appear in literals built on different predicate symbols.

Universe U is constructed by selecting, for each predicate α_ι, exactly N pairs of values (a_k, a_h), where a_j ranges over the set of all L possible values. The selection is uniform without replacement (all N pairs being distinct). In summary, the matching problems we consider are defined by a 4-tuple (n, m, L, N).

3 Observing Phase Transitions

The probability for a matching problem to be satisfiable is experimentally investigated. The phase transition region, where the probability of satisfiability abruptly drops from 1 to 0, is delineated. We then discuss the difficulty of a learning domain, depending on its position with respect to the phase transition.

3.1 Evidence of Phase Transitions

An extensive experimentation [Giordana & Saitta, 1998] considered the following settings: the cardinality of the relations N is set to 100; the number of variables n ranges in {4, 6, 10, 12, 14}; moreover, each pair (L, m) with L in [10, 50] and m in [5, 50], has been considered. For each problem, we compute if φ is satisfiable (i.e., it admits at least one model in U). We associate to each 4-tuple (n,m,L,N) the fraction P_{sol}(n,m,L,N) of matching problems that are solvable, out of 100 problems generated along these parameters.

Figure 1 plots P_{sol}(10,m,L,100) as a function of m and L, with n and N respectively frozen to 10 and 100. When m and L are both low, all problems are solved (P_{sol} is 1). Then, P_{sol} dramatically drops to almost zero along a very regular hyperbolic curve. The dashed region in the (L, m) plane shows all problems for which P_{sol} belongs to the interval [.15, .85]: this region is quite narrow, witnessing how steep the transition is from solvable to non-solvable problems. This is the phase transition, or *mushy,* region.

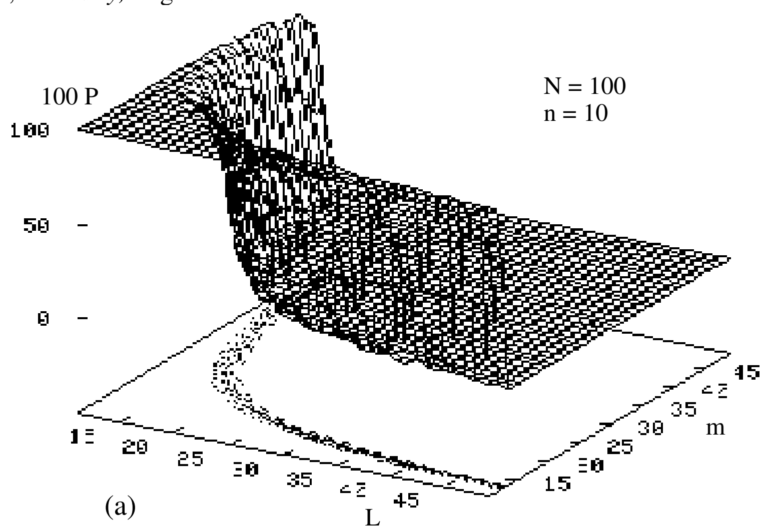

Figure 1 – 3-Dimensional plot of the probability of solution P_{sol} for n = 10, when

N = 100. In the (L, m) plane some contour level curves, corresponding to P_{sol}'s values in the interval [0.15, 0.85] have been drawn.

We also analyze the complexity of solving a problem (φ,U), measured as the number of explored nodes using a depth-first search. The region of highest complexity corresponds to the mushy region (Figure 3(a)), but it is more irregular and much broader, like a mountain chain. And the variance is high (Figure 3(b)): not all problems in the mushy region are equally difficult to solve.

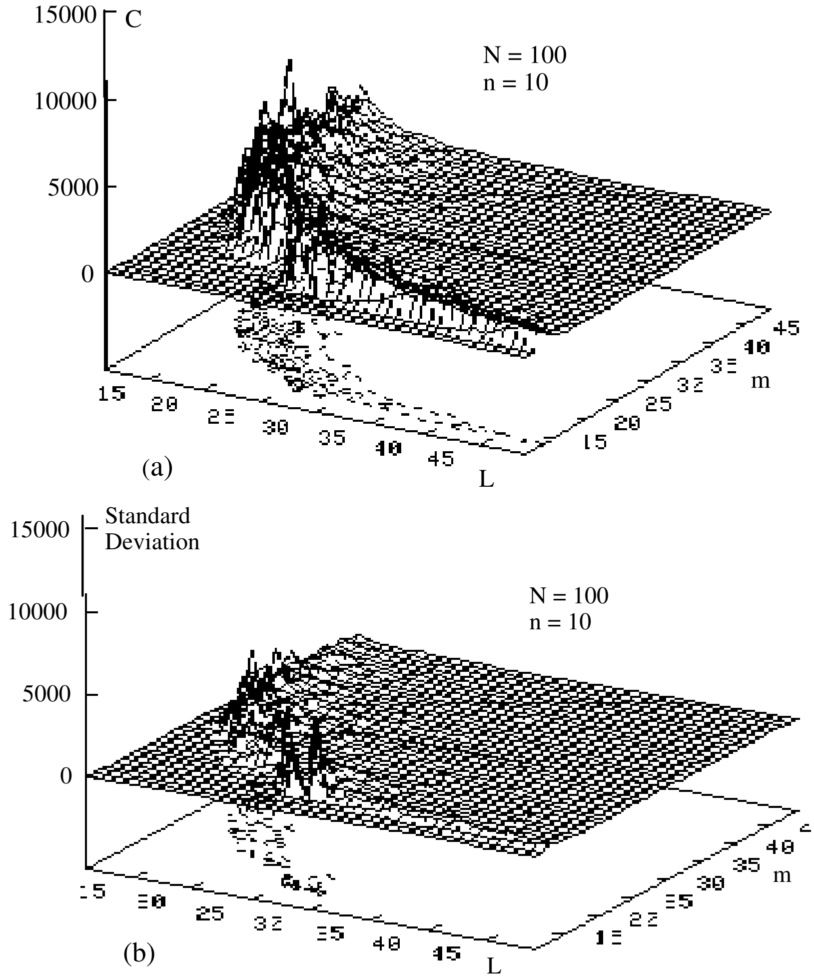

Figure 3 – (a) Plot of the complexity C (for n = 10), measured as the number of expanded nodes by a depth-first searcher, averaged over 100 problem instances in each point. (b) Plot of the standard deviation of the complexity.

3.2 Concept Learning in the (L, m) Plane

In the following we adopt the learning framework defined by Giordana et al., [1998], where an example *e*, of a relational concept corresponds to a universe U_e. With respect to a target relation φ, the positive and negative examples of the concept thus correspond to different universes: a learner solves many matching problems $<\varphi, U_e>$, for *e* ranging over the positive and negative examples. Assuming that all variable domains still have same size L, all examples belong to a line parallel to axis L in the (L,m) plane. This remark allows us to distinguish several cases.

(a) The easiest case is when positive examples, satisfying φ, fall to the left of the phase transition region, whereas negative examples fall to the right. Here, static estimates based on the value of m and L, would suffice to separate positive from negative examples (the transition phase itself being the separating surface, as in Figure 2, using formula (3)).

(b) A second case is when positive and negative examples fall within the phase transition region: the matching complexity is high in any case. This implies that positive examples would contain very few models of the target concept, and that negative examples almost match the target concept (near-misses).

(c) A third case is when all positive and negative instances fall to the right of the phase transition. This entails that positive examples are very far from being random: by construction, random universes would hardly have any chance to match φ. But indeed, examples illustrating an interesting target concept should contain regularities: one should expect these regularities to be naturally or artificially emerging from the random noise.

(d) The last case, complementary to the previous one, is when all positive and negative instances fall to the left of the phase transition. This symmetrically implies that negative examples are very far from being random. However, though it is always possible to construct formulas and artificial examples satisfying these formulas whatever their size (see next section), it might be impossible to construct a negative instance to the left of the phase transition when the value of L is too small.

4 Hardness of the Learning Problems

The conjectures made in the previous section are confronted to the actual performance of FOIL using artificial problems

4.1 Experimental Setting

The analysis considers the phase transition region, and the region to the right of the phase transition. In order to keep the problems to a reasonable complexity, we restricted ourselves to consider concepts with only four variables. In this comparatively simple case, the phase transition still occurs (see Figure 4), but the complexity peak is small enough to guarantee that the matching problem can always be solved in a reasonable time.

Artificial learning problems are constructed as follows, with relation size N=100 in all cases. In case (b), a formula φ laying in the phase transition region, and 500 examples (universes) are randomly generated as detailed in Section 2.2. The class of each example is positive if it contains at least one model of φ, negative otherwise. Examples are divided into a 200-training and a 300-test set. By construction, since φ lies in the phase transition region, positive and negative examples of φ are quasi balanced (Table 1).

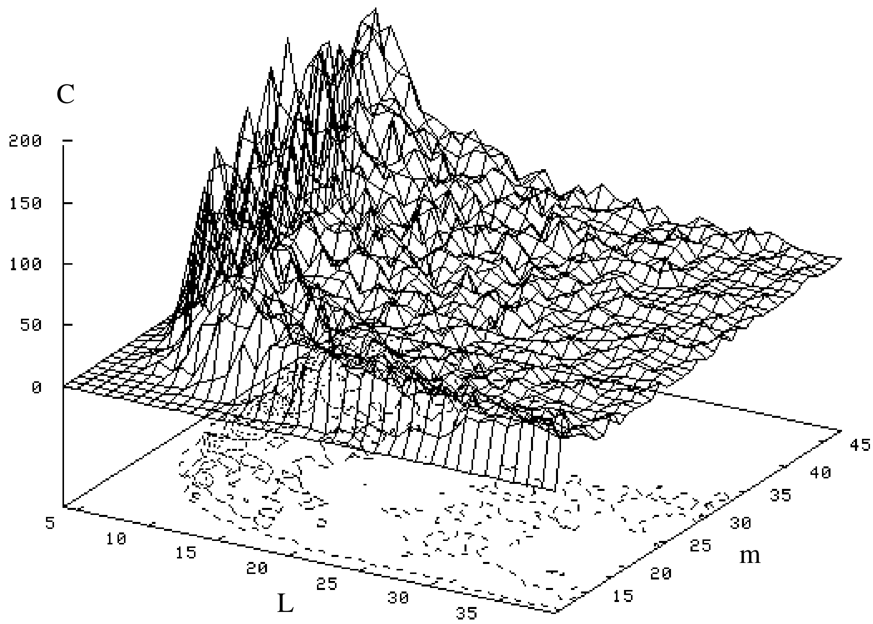

Figure 4 – Complexity peak at the phase transition for a formulas of four variables.

In case (c), a formula φ lying to the right of the phase transition has been constructed, and 500 examples are again randomly generated as detailed in Section 2.2. But none of theses examples will satisfy φ. We then select one half of these examples (100 in the training set and 150 in the test set), and transform each one of them into a positive example by adding to it a model of φ.

Five problems in the region (b) and six problems in the region (c) are generated, as summarized in Table I.

Table I

Artificial problems used for the experimental analysis. The symbol (*) denotes the problems located at the phase transition where the probability of solution is about 1/2.

Problem	m	L	LS	TS
LP_1*	9	19	118^+, 72^-	168^+, 132^-
LP_2*	8	22	95^+, 105^-	147^+, 153^-
LP_3*	7	25	116^+, 84^-	189^+, 111^-
LP_4*	6	30	135^+, 65^-	215^+, 85^-
LP_5*	6	35	71^+, 129^-	101^+, 199^-
LP_6	12	25	100^+, 100^-	150^+, 150^-
LP_7	12	30	100^+, 100^-	150^+, 150^-
LP_8	18	20	100^+, 100^-	150^+, 150^-
LP_9	18	23	100^+, 100^-	150^+, 150^-

LP_{10}	18	30	100^+, 100^-	150^+, 150^-
LP_{11}	18	50	100^+, 100^-	150^+, 150^-

4.2 Results

The hardness of the artificial learning problems generated as above is estimated from the results obtained by two relational learners having different strategies: FOIL [Quinlan, 1990], and G-Net [Anglano et al., 1997, 1998. Both learners explore candidate hypotheses ψ lying on a line $L = L_U$, being L_U the number constants in the example universes. FOIL performs a general to specific search, and moves from left to right on the line $L = L_U$. The number m of literals in the candidate hypothesis increases along the search. The hypothesis to be considered and specialized is selected on the basis of its information gain.

G-Net is based on an evolutionary search: It considers a population of hypotheses lying also on the line $L = L_U$, and moves back and forth as evolution determines whether these hypotheses should be specialized or generalized. Further, the selection of the hypotheses to be considered and refined is guided by the MDL principle.

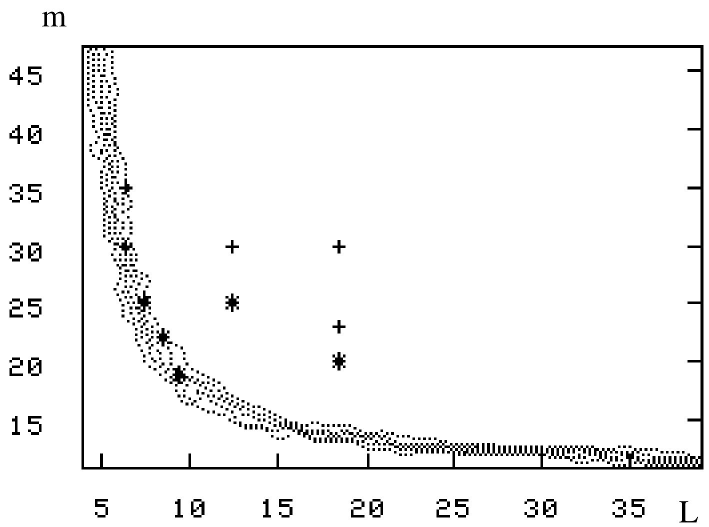

Figure 5 – Contour plots of the probability of solution. The symbol "+" denotes the problems where Foil succeeded, whereas symbol "*" denotes the problem where FOIL failed.

Being FOIL faster than G-Net, the experiments have been primarily dove using FOIL. G-Net has been run for comparison on the only learning problems that seemed most significant to us. In all cases, the two systems have been surprisingly in agreement, reporting minimal differences in the classification rate. The results are summarized in Table II, and in Figure 5 a graphical representation of the generated learning problems is reported, together with an indication of success/failure.

Table II

Summary of the experimental results obtained for the set of learning problems described in Table I

Probl.	#Nof Clauses	Complexity	Class.Rate (LS) [%]	Class. Rate (TS) [%]	CPU Time [sec]	Avg. #N of Models
LP_1*	10	<7-13> 8.9	88	50	398.2	1.7
LP_2*	11	<6-11> 8.6	92	53	624.7	2.0
LP_3*	15	<7-11> 8.9	98.5	52	513.9	4.9
LP_4*	1	6	100	100	43.3	2.1
LP_5*	1	6	99.9	100	132.6	1.25
LP_6	12	<1-12> 6	81	58	825.4	10.7
LP_7	1	6	100	96	73.4	34.6
LP_8	6	<1-11> 5	98.5	75.3	723.8	1.4
LP_9	1	9	100	99.6	620.1	1.0
LP_{10}	1	6	100	99.6	36.9	4.2
LP_{11}	1	6	100	99.6	72.2	9.1

For problems laying in the mushy region, both systems fail when the value of L is low ($L \leq 25$, problems LP_1 to LP_3). They overfit the training set (φ is approximated as a dozen of concept ψ) and the accuracy on the test set is close to the default accuracy. When the value of L is high ($L > 25$, LP_4 and LP_5) both systems find a quasi-perfect solution.

For problems lying to the right of the mushy region (LP_6 to LP_{11}), both systems similarly fail when the value of L is low (LP_6 and LP_8), though failures are observed for lower values of L than in the case (b) (around $L = 22$ for $m = 18$). For higher values of L both systems succeed from the point of view of predictive accuracy. Interestingly, they do not find the exact solution φ; rather they find a single concept ψ lying in the mushy region (the number of literals is 6 or 7).

This can be explained as follows. FOIL is bound to stop the search as soon as it finds a concept complete and correct with respect to the training set. But all ψ generalizing the target concept φ are complete: they cover the positive examples by construction. One thus only searches for generalizations ψ that are sufficiently specific to rule out random examples. Such ψ would then be consistent with respect to the negative training examples since those where generated using random universes.

Considering generalizations of φ on the right side of the mushy region, these will rule out most negative instances: and given the relatively small datasets we used, they will likely be consistent with respect to the training and test sets.

FOIL will end up exploring the generalization of φ laying in the mushy region; provided there exists a sufficient number of such generalizations, it will succeed in finding a complete and correct solution ψ. Though G-Net might explore more specific concepts than FOIL, it is also biased toward generality (using MDL principle instead of information gain): it will thus end up discovering more or less the same complete and correct generalization of φ as FOIL.

5 Discussion

As shown in the previous section, both FOIL and G-Net find hard, and actually fail to solve, learning problems including few constants (low values of L). Why is it so?

Consider first the case where the target concept φ lies in the mushy region (case (b)). In this case, negative examples fail to match φ by only a few traits, i.e. they are "near-misses" of φ. In the left side of the mushy region, all concepts ψ, including the generalizations of φ, match on average any example. Let *M*(m, L) denote the average number of models of a m-literals formula in universes including L constants. From [Giordana & Saitta, 98; Prosser 96], the number *M*(m, L) increases as L decreases (the extension of any predicate being of constant size N).

Further, the variance of *M*(m, L) is high: this can be inferred from the high variance of the matching complexity (Figure 3 (b)). Indeed, the complexity depends on the probability of finding a model, and hence, on the total number of models of the current formula in the current universe: the complexity thus exponentially decreases with the number of models in the universe.

Figure 3(b) shows that the variance reaches its maximum around the corner of the mushy region. FOIL starts the search in the left side of the mushy region, where any example matches on average any formula. The information gain criterion then fails to guide the search; further, FOIL tends to be misled by the fluctuations of the number of models of the candidate hypotheses. When L decreases, the left side of the mushy region is larger, and the complexity landscape is more rugged: both facts explain the increasing difficulties met by FOIL search strategy.

Practically, experiments give an order of idea of *M*(m,L): for problem LP_1 (m=4, L=35), *M* is 66; for problem LP_2 (m=4, L=19), *M* is 667. But, when increasing the number of literals, *M* rapidly decreases. In 10 other experiments done with (m=6, L=19), *M* is 62 on average.

When FOIL explores too general concepts, it fails to find consistent hypotheses; it then specializes the candidate hypotheses, until it ends up exploring the concepts with the right level of generality, in the mushy region. But then, the high number of models hinders the search. The system then specializes again the candidate hypotheses, until the number of models falls down to tractable values.

But these hypotheses are then too specific compared to the target concept: many of them must then be retained in order to cover all positive examples. Ultimately, these concepts show a low predictive accuracy on the test set. Incidentally, such a behavior closely account for the small disjunct phenomenon [Holte et al. 89].

G-Net meets the same difficulties, which is more surprising given the fact that it goes back and forth, specializing and generalizing the candidate hypotheses. However, it soon dismisses all hypotheses lying on the right side of the mushy region, since these hardly cover any positive example. G-Net is then bound to explore the same regions, and meet the same difficulties as FOIL.

Let us now consider the case where the target concept φ lies on the right side of the mushy region (case (c)). These problems were expected to be very hard, since the target concept is hardly satisfiable. But surprisingly, both FOIL and G-Net succeed on such learning domains, provided again that L is low enough.

The explanation proposed for this fact is the following. The distribution of the positive examples was altered from the random generation, to ensure that they include at least one model of φ. If we consider the generalizations of φ, their number of models is thus artificially increased, compared to the random (negative) examples: the information gain criterion hence favors the selection of these generalizations. (Experimentally, 4-literal generalizations of φ in LP_9 admit 244 models on average for positive examples, and 225 models on average for negative examples. On one hand, the search is biased toward exploring generalizations of φ, and it stops when it meets a generalization of φ in the mushy region.

On the other hand, the number of k-literal generalizations of φ varies with $\binom{n}{k}$, (being m is the number of literals in φ): the more specific the target concept, the more likely its generalizations are explored. The complexity analysis of G-net offers evidence supporting this fact. The system explores 28,000 hypotheses in LP_7 (m=12) against 15,000 in LP_{10} (m=18).

6 Conclusions

This paper focuses on the average case analysis of the matching problem, considered as a constraint satisfaction problem. As it could have been expected, relational learning is actually needed when the target concept to discover is non trivial (lies in the mushy region of the CSP) or almost unsatisfiable (lies on the right side of the mushy region). Problems in the mushy region are expected to be hard for two reasons. On the one hand positive and negative instances are very similar and so intrinsically difficult to discriminate. On the other hand, verifying hypotheses in the mushy region may be very complex.

Unexpectedly, experiments on artificial problems suggest that learning an almost unsatisfiable target concept is not always a hard learning problem: many generalizations of the target concept appear consistent with respect to the training set provided that they are sufficiently specific. Generality-biased learners would then end up with a complete and correct solution belonging to the mushy region. Some care must however be exercised in interpreting these results, as we actually considered random negative examples.

Further research will examine what happens when negative examples are no longer random, e.g. when the learning problem is to separate examples of two different target concepts. Other complexity parameters should then be defined to account for the degree of difference of these concepts.

Another perspective of research is concerned with learning a target concept in the mushy region. In this case, there is no other possibility than learning exactly the target concept: any generalization (resp. specialization) would unlikely to be consistent (resp. correct). One possibility is to take advantage of the fact that negative examples fail to match the target concept by only a few traits, i.e. are near-misses.

Ultimately, the major obstacle to relational learning remains the complexity of dealing with hypotheses in the mushy region: as was shown, exploring this region is unavoidable in truly relational learning domains. When target concepts include a low number of variables(4 or 5), the search complexity remains affordable. Otherwise, the learning search fully faces with the exponential complexity of matching.

This difficulty could be alleviated by relaxing the matching task, e.g. replacing the exhaustive exploration of the universe by a stochastic exploration: as discussed in [Sebag & Rouveirol, 97; Giordana & Saitta, 98], stochastic matching gives correct and probabilistically complete answers within bounded resources (any-time matching). A perspective for further research is to study how the phase transition would be affected by using a stochastic resolution process.

References

Anglano C., Giordana A., Lo Bello G., and Saitta L. (1997). "A Network Genetic Algorithm for Concept Learning". In *Proc. 7th Int. Conf. on Genetic Algorithms* (East Lansing, MI), pp. 434-441.

Anglano C., Giordana A., Lo Bello G., and Saitta L. (1998). "An Experimental Evaluation of Coevolutive Concept Learning". In *Proc. 15th Int. Conf. on Machine Learning* (Madison, WI), pp. 19-27.

Bailleux O. (1998). "Local Search for Statistical Counting". In *Proc. 15th Nat. Conf. on Artificial Intelligence* (Madison, WI), pp. 386-391.

Brassard G., and Bratley P. (1988). *Algorithmics: Theory and Practice.* Prentice Hall, Englewood Cliffs, NJ. [pp. 262-267].

Cheeseman P., Kanefsky B., and Taylor W.M. (1991). "Where the *Really* Hard Problems Are". In *Proc. 12th Int. Joint Conf on Artificial Intelligence* (Sidney, Australia), pp. 331-337.

Crawford J.M., and Auton L.D. (1996). "Experimental Results on the Crossover Point in Random 3-SAT". *Artificial Intelligence, 81,* 31-58.

de Raedt (1996). "Induction in Logic". In *Proceedings of the Third International Workshop on Multistrategy Learning* (Harpers Ferry, Wv), pp. 29-38.

Freeman J.W. (1996). "Hard Random 3-SAT Problems and the Davis-Putnam Procedure". *Artificial Intelligence, 81,* 183-198.

Gent I.P., and Walsh T. (1994). "Easy Problems are Sometimes Hard". *Artificial Intelligence, 70,* 335-345.

Gent I.P., and Walsh T. (1996). "The TSP Phase Transition". *Artificial Intelligence, 88,* 349-358.

Giordana A., Neri F., Saitta L., and Botta M. (1998). "Integrating Multiple Learning Strategies in First Order Logics". *Machine Learning, 27,* 209-240.

Giordana A., and Saitta L. (1998). "Phase Transition in Learning with FOL Languages", TR 97-25, Dip. di Informatica, Università di Torino.

Giordana A., Saitta L. and Botta M. (1999). "An Experimental Study of Phase Transitions in Matching", In *Proc. 16th Int. Joint Conf. on Artificial Intelligence*, Stockolm, Sweden, Aug. 1999, to appear.

R.C. Holte and L.E. Acker and B.W. Porter (1989). "Concept learning and the problem of small disjuncts". In Proceedings of IJCAI-89 (Detroit, MI), pp. 813-818.

Hogg T. (1996). "Refining the Phase Transition in Combinatorial Search". *Artificial Intelligence, 81,* 127-154.

Hogg T., Huberman B.A., and Williams C.P. (1996). *Artificial Intelligence, 81,* 1-15.

Hogg T., Huberman B.A., and Williams C.P. (Eds.) (1996). *Artificial Intelligence,* *Special Issue on Frontiers in Problem Solving: Phase Transitions and Complexity, 81 (1-2).*

Prosser P. (1996). "An Empirical Study of Phase Transitions in Binary Constraint Satisfaction Problems". *Artificial Intelligence, 81,* 81-110.

Sebag M., and Rouveirol C. (1997). "Tractable Induction and Classification in FOL". In *Proc. 15th Int. Joint Conf. on Artificial Intelligence* (Nagoya, Japan), pp. 888-892.

Selman B., and Kirkpatrick S. (1996). "Critical Behavior in the Computational Cost of Satisfiability Testing". *Artificial Intelligence, 81,* 273-296.

Smith B.M., and Dyer M.E. (1996). "Locating the Phase Transition in Binary Constraint Satisfaction Problems". *Artificial Intelligence, 81,* 155-181.

Vlasie D. (1996). "The Very Particular Structure of the Very Hard Instances". In *Proc. 8th Innovative Applications of Artificial Intelligence Conference* (Portland, OR), pp. 266-270

Walsh T. (1998). "The Constrainedness Knife-Edge". In *Proc. 15th National Conf. on Artificial Intelligence* (Madison, Wisconsin, USA), pp. 406-411.

Williams C.P., Hogg T. (1994). "Exploiting the Deep Structure of Constraint Problems". *Artificial Intelligence, 70,* 73-117.

Zhang W., and Korf R.E. (1996). "A Study of Complexity Transition on the Asymmetric Travelling Salesman Problem". *Artificial Intelligence, 81,* 223-239.