C4.5 Competence Map: a Phase Transition-inspired Approach

Nicolas Baskiotis NBASKIOT@LRI.FR
Michèle Sebag SEBAG@LRI.FR
TAO Group, CNRS & INRIA, Bât 490 Université Paris-Sud, F-91405 - Orsay Cedex

Abstract

How to determine *a priori* whether a learning algorithm is suited to a learning problem instance is a major scientific and technological challenge. A first step toward this goal, inspired by the Phase Transition (PT) paradigm developed in the Constraint Satisfaction domain, is presented in this paper.

Based on the PT paradigm, extensive and principled experiments allow for constructing the Competence Map associated to a learning algorithm, describing the regions where this algorithm on average fails or succeeds. The approach is illustrated on the long and widely used C4.5 algorithm. A non trivial failure region in the landscape of k-term DNF languages is observed and some interpretations are offered for the experimental results.

1. Introduction

The performance of Machine Learning (ML) algorithms has been intensively studied in a general perspective, both empirically and theoretically (see among many others (Holte, 1993; Wolpert & Macready, 1995; Lim et al., 2000)). Currently, the rapid growth of ML and Data Mining applications also asks for easy-to-use and specific guidelines, estimating *a priori* whether any given algorithm is suited to a particular problem instance.

How to select the best learning algorithm depending on the problem instance at hand has been considered a key question since the 90's. This question was formalised as a Meta-Learning problem in (Brazdil et al., 1994; Pfahringer et al., 2000; Bensusan & Kalousis, 2001). The very elegant approach of meta-learning (MetaL), like all learning applications, heavily depends upon the selection of the examples and their representation. Actually, how to represent MetaL examples, i.e. instances of learning problems, appeared to be a most difficult issue (Kalousis, 2002).

This paper presents an alternative to Meta-Learning, inspired from the complexity paradigm developed in the Constraint Satisfaction community since the 90's, where it is referred to as the Phase Transition (PT) paradigm (Hogg et al., 1996).

The PT paradigm was ported to Inductive Logic Programming by Giordana and Saitta (2000); it provided a rigorous framework for investigating the scalability of existing algorithms and the impact of the complexity barrier on the learning performances (Botta et al., 2003). Along the same lines, the PT paradigm was ported to Attribute-Value Learning and used to study the feasibility of learning in k-term DNF languages (Rückert et al., 2002).

In this paper we investigate the use of the PT paradigm for constructing principled competence maps attached to any learning algorithm, characterising the regions where this algorithm on average succeeds or fails. On one hand, such competence maps can be exploited as look-up tables, providing all needed information to select the best algorithm in a given region of the problem instance landscape, thereby achieving the Meta-Learning goal. On the other hand, the competence map attached to any particular algorithm allows for a precise identification of its failure region. Ultimately, the approach will hopefully lead to a better understanding of the practical frontiers of ML algorithms.

The proposed approach is illustrated on the particular case of C4.5 (Quinlan, 1993), one long and still widely-used ML algorithm (Witten & Frank, 1999). The C4.5 competence map built after principled and extensive experiments both demonstrates its general robustness and displays a non-trivial failure region. Specifically, the learning difficulty does not increase monotonically with the complexity of the underlying target concept.

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

These results are discussed and some tentative interpretations are proposed.

The paper is organised as follows. Section 2 briefly reviews related work and presents the PT paradigm. Section 3 describes a PT model for building C4.5 competence map. Section 4 presents this competence map and provides some interpretations for the observed regularities[1]. A refined model is proposed and experimented in Section 5. The paper ends with some perspectives for further research.

2. Related work

This section briefly reviews works concerned with *a priori* estimation of a learner performance.

2.1. Meta-learning

This estimation problem was formalised as a new learning problem in the Meta-Learning approach (MetaL) (Brazdil et al., 1994). MetaL thus faces two difficult issues: the selection and the representation of MetaL examples. A MetaL example most often involves a pair (ML problem instance, ML algorithm), labeled with the performance of the algorithm on the ML problem instance[2].

How to represent an ML problem instance, i.e. a set of ML examples, was tackled using diverse descriptors, e.g. number of examples, number of attributes, percentage of missing values, landmarkers (Pfahringer et al., 2000). The difficulty is due to the fact that these descriptors should account for the example distribution in the ML problem instance; and characterising the example distribution is not easier than learning. A second difficulty concerns the selection of the ML problem instances, most often derived through principled perturbations of problems in the Irvine repository (Blake et al., 1998), e.g. increasing the rate of missing values or incorporating irrelevant attributes. Two critical issues, the representativity of these problems and the choice of the perturbations considered, impose strong biases on the MetaL classifier (Kalousis, 2002).

2.2. Phase transition

Relatedly, the Phase Transition paradigm was initially developed to better understand the performances of Constraint Satisfaction (CS) algorithms, and *where the really hard problems are* (Cheeseman et al., 1991).

The notion of stochastic complexity was introduced for this purpose, opening new avenues of research (Hogg et al., 1996). Given *order parameters* on CS problems, i.e. constraint density and tightness, and a distribution probability on the CS problem instances, stochastic complexity is viewed as a random variable conditioned by the order parameters. For given density and tightness values, stochastic complexity manifests as the actual complexities observed over all CS problem instances with this density and tightness. Following this paradigm, a regular complexity landscape can be observed: the actual complexity is negligible in two wide regions, the YES and NO region, where the probability of satisfiability is respectively close to 1 and close to 0. These regions are separated by a narrow one, the so-called phase transition, where the probability of satisfiability abruptly falls from almost 1 to almost 0, and where the hardest problems on average concentrate.

The transportation of such a paradigm to Inductive Logic Programming (ILP), pioneered by Giordana and Saitta (Giordana & Saitta, 2000), was meant to study the complexity barrier in ILP; it enabled observing and understanding its far-fetched effects on learning performances (Botta et al., 2003).

2.3. Feasibility of k-term DNF learning

The PT paradigm was also exploited by Rückert et al. to study the feasibility of learning formulas in Disjunctive Normal Form, involving at most k disjuncts (k-term DNF) (Rückert et al., 2002). Considering as order parameters the number m of attributes (also referred to as variables), the number p (respectively n) of positive (resp. negative) examples in the training set, and the number k of disjuncts, extensive experiments were conducted to estimate the probability of finding a target concept i) covering all p positive examples and rejecting all n negative examples; ii) expressed as a k-term DNF concept. Formally, a k-term DNF over variables $\{x_1, \ldots, x_m\}$ is the disjunction of k terms, where a term is a conjunction of literals, and a literal is either a variable x_i or its negation.

In this approach, a (pessimistic) estimate of the learning feasibility is provided using uniformly selected positive and negative examples. However, such a model is not appropriate to the MetaL goal, as real-world training examples are not usually selected and labelled from a uniform distribution.

[1] All dataset and results are available at `http://www.lri.fr/∼nbaskiot/c45data`

[2] An alternative representation involves triplets (ML problem instance, algorithm1, algorithm2), labelled as positive iff algorithm1 outperforms algorithm2 on this problem instance.

3. Overview

This paper focuses on estimating *a priori* the performance of learning algorithms. The presented approach is illustrated on the C4.5 algorithm (Quinlan, 1993), assuming the reader's familiarity with this well known ML algorithm.

3.1. Order parameters

As opposed to the learning feasibility with respect to a given hypothesis space (Rückert et al., 2002), the learning performance is evaluated with respect to the underlying target concept.

In the rest of the paper, the target concept space is set to formulas in Disjunctive Normal Form. Accordingly, we consider the Rule mode of C4.5; in this mode, a set of decision trees is constructed, pruned and compiled into rules; the rules are then filtered and ordered on the training set, and the ruleset is used as a decision list on the test examples (Quinlan, 1993).

In this way, the considered hypothesis search space coincides with the target concept space; the lack of syntactic language biases is meant to simplify the interpretation of the learning performance.

Four order parameters are considered at this stage:

- m is the number of boolean variables or attributes $x_1, \ldots, x_m$ representing the problem domain, $m \in \mathbb{N}$.
- k is the number of (distinct) terms C_i in the target concept, $k \in \mathbb{N}$; each term is the conjunction of a set of (distinct) literals y_i, being either a variable (x_i) or its negation ($\bar{x}_i$).
- l is the number of literals in a term, $l \in \mathbb{N}$. Assuming that all terms are of the same length, the target concept space is actually a restriction of the DNF language, termed (k, l)-term DNF. This restriction will be relaxed in section 5.1.
- r is the imbalance ratio (fraction of positive examples) in the training set.

The choice of these parameters will be discussed in section 3.3.

3.2. Constructing a Competence Map

For each (m, k, l, r) setting, 100 learning problem instances noted $\mathcal{L}_i(m, k, l, r), i = 1 \ldots 100$ are generated; indices m, k, l, r will be omitted when clear from the context. A learning problem instance $\mathcal{L}$ is composed of a target concept, a training set and a test set.

The target concept noted tc involves k distinct terms C_i; each C_i is the conjunction of l literals, set to a variable or its negation with equal probability, such that C_i involves l distinct variables uniformly selected in $\{x_1, \ldots, x_m\}$.

For each problem instance, a 400-example training set is generated: examples are uniformly generated in $\{0, 1\}^m$, labelled according to tc and filtered or uniformly repaired until the desired fraction r of positive examples is obtained ($r = 1/2, 1/3, 1/5, 1/9$). The test set is made of 400 examples evenly distributed among positive and negatives ones.

For each learning problem instance $\mathcal{L}$, C4.5[3] learns from the training set a k'-term DNF concept noted $\hat{tc}$; the error $Err(\mathcal{L})$ is the probability of $\hat{tc} \neq tc$, estimated on the test set.

The C4.5 error noted $Err(m, k, l, r)$ averages $Err(\mathcal{L}_i(m, k, l, r))$ for $i = 1 \ldots 100$. This hypersurface in the (m, k, l, r) landscape, viewed as a probabilistic error surface, defines the competence map of C4.5.

3.3. Discussion

The main limitation of the above order parameters is that they do not induce a "canonic" representation of the target concept space with respect to the learning error. On one hand, a (k, l)-term DNF might admit several logically equivalent but syntactically distinct expressions, corresponding to distinct order parameters values. Formally, this implies that the C4.5 error reported for a (k, l) setting can also reflect the error made with other settings. In the worst case, this might blur the competence map, smoothing the error and possibly hiding abrupt transitions in the performance landscape.

On the other hand, C4.5 will have the same performance when the target concept is replaced by its negation (flipping the class of every example), although the order parameters attached to a (k, l)-term DNF target concept and to its negation (a (l^k, k)-term DNF in the worst case), differ. Similarly, this equivalence could lead to blurring the competence map.

Such drawbacks of the order parameters can be detected by checking the variance of the error, that is, the precision of the competence map. Ultimately, the choice of the order parameters can mainly be justified with respect to the quality of the competence map, as defined below.

[3] C4.5 is launched with command line `C4.5 -i10`, using all other default options.

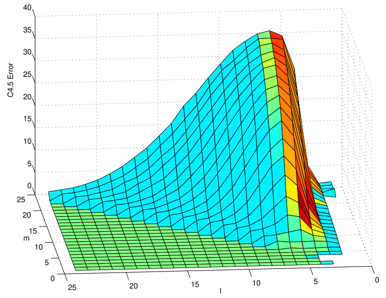

Figure 1. Competence Map of C4.5 for (k,l)-term DNF, represented in plane (m,l) with $k = 15$, balanced is $r = 1/2$

4. Competence Map and Phase Transition

This section presents and discusses the competence map constructed after the above model.

4.1. Experimental setting and goal

Extensive experiments have been conducted for $m \in [1, 30]$, $k \in [1, 20]$, $l \in [1, m]$, $r = 1/2, 1/3, 1/4, 1/5, 1/9$, running C4.5 on more than 1,000,000 learning problem instances for an overall computational cost of 12 days on a PC Pentium-IV.

To each (m, k, l, r) setting is associated the error $Err(m, k, l, r)$ of C4.5, measured as detailed in section 3.2.

The goal of the experiments is both to check the relevance of the order parameters, and, equivalently, to produce a good quality competence map. The quality will be evaluated from both the precision (low variance) and the intelligibility of the competence map. Ideally, the competence map should display a sufficiently regular behaviour, allowing for the detection of non trivial regularities. Ultimately, interpretations for these regularities should lead to a better understanding of the algorithm strengths and weaknesses.

4.2. Experimental results

The error is in most regions very low, confirming the known robustness of C4.5 (Fig. 1). However, a failure region (error equal or greater than 20%) is observed as the term length l takes on medium values ($l \in [5, 10]$), whenever the number m of variables is non negligible ($m > 15$ in Fig. 1). It is no surprise that the learning difficulty increases with the total number m of variables, since the representativity of the training set (fixed to 200 positive and 200 negative examples in Fig. 1) decreases. The relationship between the error and the term length l appears less obvious: $Err(m, k, l, r)$ first increases then decreases as l increases, for fixed m, r and k; and the error is almost insensitive to the imbalanced ratio r.

The fact that error increases as l first increases ($l \in [1, 6]$, Fig. 1) is naturally blamed on the myopic search of C4.5, greedily optimising the gain ratio criterion. Indeed, as the term length increases, each one of its literals becomes less discriminant; further, it is often the case that both a variable and its negation contribute to (appear in some terms of) the target concept. Like in the standard XOR problem, the gain ratio criterion might thus miss the variables that contribute to the target concept. Therefore, a significant amount of look-ahead would be necessary to prevent the greedy search from becoming trapped in local optima due to erroneous early choices. In other words, the (univariate) gain ratio becomes a noisy selection criterion as the target concept involves more specific terms; hence the probability of making no errors along l selections based on this criterion gets exponentially low with l.

When the term length l increases again ($l > 10$ in Fig 1), the error decreases. This empirical finding was unexpected since the learning difficulty is usually seen as proportional to the target concept complexity, and l is considered a factor of complexity. The fact that the failure region does not much depend on the imbalance ratio r is unexpected too, since imbalance example distributions are widely acknowledged as a factor of complexity. Still, Fig 2 shows that the error peak is observed for $l = 6$ or $l = 7$ ($m = 25$, $k = 15$ or $k = 25$, $r = 1/2, 1/3, 1/5, 1/9$) and the peak smoothly increases as r decreases. Similar results are observed for other values of m and k.

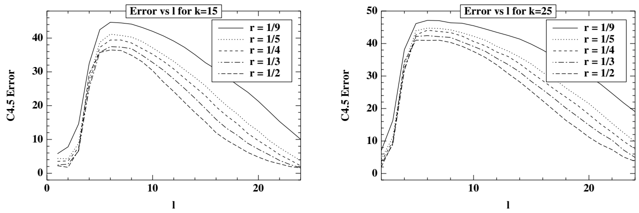

Figure 2. C4.5 Error vs l for imbalance ratio $r = 1/2, 1/3, 1/5, 1/9$, $m = 25$, $k = 15$ (left) and 25 (right)

The tentative interpretation offered for this finding is based on the phase transition effects and the learning bias toward generality (Botta et al., 2003). Specifically, rules produced by C4.5 are not arbitrarily long as they must cover a significant number of training examples; on average their size is limited by the (log of the) size of the training set. In the experimental range, this maximal size, (noted l_c) is almost constant (#positive examples in $[45, 200]$ out of 400 examples).

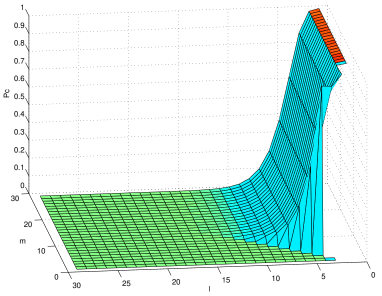

Figure 3. Coverage P_c of (k,l)-term DNF, represented in plane $(m, k = 15, l)$

Therefore, the probability $\varepsilon(m,l)$ for a leaf in a C4.5 tree to be irrelevant (differ by at least one irrelevant literal from a generalisation of true conjunct) when learning a (k,l)-term DNF concept is bounded by the probability of selecting at least one irrelevant literal out of l_c choices. On the other hand, the probability of selecting an irrelevant feature decreases as l increases.

The rise of the error as l increases up to l_c is thus explained as the number of choices (hence the probability of error) increases; the fall of the error for $l > l_c$ is explained as the error is the product of l_c factors which all decrease as l increases[4].

More intensive experiments are required to test the above interpretation: in order to significantly modify the critical size l_c, the size of the training set must be increased by one or several orders of magnitude.

4.3. Phase transition

Following the CS inspiration, we also considered the satisfiability or coverage of (k,l)-term DNFs, estimated as their probability of covering a uniformly selected example. For each learning problem instance $\mathcal{L}$, let $P_c(\mathcal{L})$ denote the fraction of examples labelled positive out of 1000 uniformly extracted examples, and define $P_c(m,k,l)$ as the average of $P_c(\mathcal{L}_i(m,k,l)), i = 1\ldots,100$. Fig. 3 shows as expected a sharp (exponentially fast) decrease of the concept coverage as the term length l increases. Interestingly, the region where the satisfiability abruptly drops broadly coincides with the failure region of C4.5, where the error is above 20% (Fig. 1).

4.4. Discussion

The goal set in section 4.1 is only partially achieved. Although the above competence map displays interesting regularities, it does not allow for a precise estimation of the error when learning a (k,l)-term DNF. Specifically, the error variance is high in the failure region.

Also, the restriction to (k,l)-term DNF languages is a severe one, as real-world concepts usually involve disjuncts of diverse generality. But several attempts made to relax this restriction and consider richer DNF languages, only result in increasing the error variance, and the imprecision of the competence map.

These remarks lead us to consider another PT model.

5. Operational Phase Transition

In this section, the model presented in section 3.2 is refined to produce a more precise and general competence map of C4.5.

5.1. Observed vs Controlled Order Parameters

The competence and coverage maps (Figs. 1, 3) suggest that C4.5 error might be related to the coverage P_c of the underlying target concept.

However, whereas parameters (m,k,l) allowed for directly generating the learning problem instances, coverage P_c is hardly a generative, controllable parameter. Finding a k-term DNF target concept, uniformly selected among the k-term DNF concepts with coverage P_c, is a difficult combinatorial problem.

Therefore, an extended PT model is defined over k-term DNF formulas. This model takes as generative order parameters the number m of variables and the number k of terms. Target concepts are uniformly generated as in section 3.2, except for the fact that the term lengths are uniformly and independently selected in $[1,m]$. A further requirement is that no term is subsumed by another term in a same target concept.

For each learning problem instance $\mathcal{L}$, the coverage $P_c(\mathcal{L})$ is measured (see below) and P_c will be used as another order parameter, observed as opposed to generative of the model.

The effects of noise in the data will be investigated in section 5.4. Due to space limitations, only balanced training sets ($r = 1/2$) will be considered in the rest

[4]Further research will be devoted to an analytical study of this error peak. Let us denote $\eta(m,l)$ the probability of top ranking an irrelevant feature among the total m features, of which l are relevant. The probability for a j-length monom constructed to be irrelevant (generalize no true monom), denoted $Pr(j,l,m,)$ is given as:

$$Pr(j+1,l,m) = Pr(j,l,m) + \eta(m-j,l-j) \times (1 - Pr(j,l,m))$$

and the probability $\varepsilon(m,l)$ to construct an irrelevant monom is then $Pr(l_c,m,l)$.

of the paper.

5.2. Operational Competence Map

We experimented the above model on 450,000 learning problem instances $\mathcal{L}$, likewise composed of a target concept, a training set and a test set, generated as follows:

For each $m \in \{30, 40, 50, 60, 70, 80\}$ and $k \in \{5, 10, 15, 20, 25\}$, 15,000 k-term DNF target concepts are generated. For each concept, the length l_i of the i-th term is uniformly selected in $[1, m]$, where each term is constructed as in section 3.2, additionnally requiring that no term is subsumed by another one in a same target concept. The training and test sets attached to a given target concept are generated as in section 3.2 (balanced datasets).

For each learning problem instance $\mathcal{L}$, its coverage $P_c(\mathcal{L})$ is measured as the fraction of examples out of 1000 uniformly selected examples that are covered by the target concept; the error $Err(\mathcal{L})$ is measured as detailed in 3.2.

5.3. Results

Fig. 4 plots all 15,000 learning problem instances $\mathcal{L}$ considered for $m = 50$ and $k = 15$, with coordinates $(P_c(\mathcal{L}), Err(\mathcal{L}))$. This figure shows that the error significantly rises when the coverage is below 50%. For all problem instances with coverage lower than 30%, the C4.5 error is above 20%. This trend is confirmed for other values of m and k. It appears that the coverage

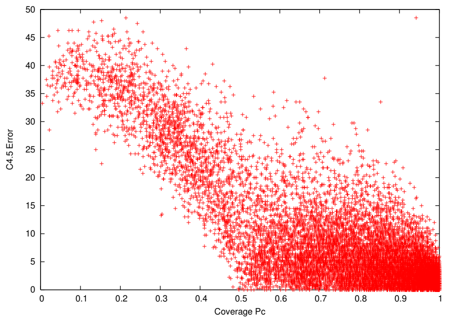

Figure 4. C4.5 Error vs Concept Coverage in k-term DNF languages, for $m = 50$, $k = 15$

is a weak predictor of the error, especially when the coverage is close to 50%. Another quantity was considered, the average term coverage P_{ac} defined for each problem instance with target concept $tc = C_1 \vee \ldots \vee C_k$, as the average coverage of terms C_i. Fig 5 plots all learning problem instances $\mathcal{L}$ considered for $m = 50$ and $k = 15$, represented as $(P_{ac}(\mathcal{L}), Err(\mathcal{L}))$.

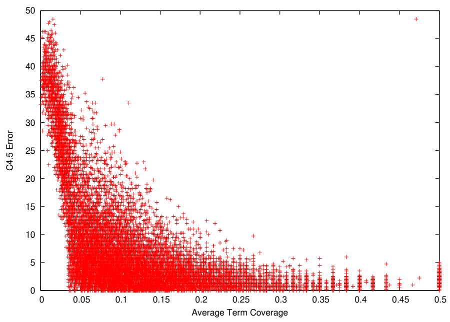

Figure 5. C4.5 Learning Error vs Average Term Coverage in k-term DNF languages, for $m = 50$, $k = 15$

The legibility of Figs. 4 and 5 is hindered as the distribution of problem instance is far from uniform with respect to P_c, with a bias toward high coverage target concepts. Following (Chapelle et al., 2000), these distributional effects are filtered using the convolution of the error with a Gaussian kernel of parameter K:

$$Err(P_c) = \frac{\sum_{\mathcal{L}} Err(\mathcal{L}) \times exp^{-K \times (P_c(\mathcal{L}) - P_c)^2}}{\sum_{\mathcal{L}} exp^{-K (P_c(\mathcal{L}) - P_c)^2}} \quad (1)$$

Fig. 6 (left) displays the error behaviour versus the concept coverage P_c for $m = 50$ and k in $\{10, 15, 20, 25\}$, for the Gaussian parameter $K = 100$. Similar behaviours are observed for other values of m. A competence region, where the error is lower than 10%, is observed for high coverage concepts $P_c > 50\%$, while an error peak is observed around $P_c = 10\%$. An even clearer picture is obtained in Fig. 6 (right), showing the error behaviour versus the average term coverage P_{ac}, for $m = 50$ and k in $\{10, 15, 20, 25\}$, with $K = 100$. In all settings, the error is lower than 5% for an average term coverage $P_{ac} > 5\%$ (competence region), while the error rises abruptly when $P_{ac} < 4\%$ (failure region).

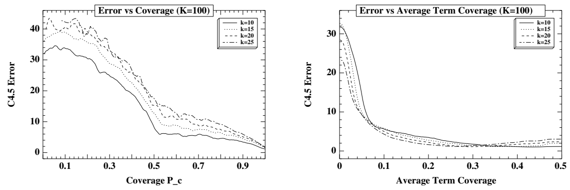

Figure 6. C4.5 Error vs Coverage and Average Term Coverage in k-term DNF languages, for $m = 50$, $k = 10, 15, 20, 25$, with Gaussian parameter $K = 100$.

The competence maps obtained for other settings were similar, showing a broad competence region, where the error is less than 5%, and a smaller failure region, where the error is higher than 25%.

Average	Coverage P_c								
Term Cov	$< 10\%$	10 - 20%	20 - 30 %	30 - 40 %	40 - 50 %	$> 50\%$	cumulated	# Fail	# OK
$< 1\%$	33.7 ± 4.5	32.4 ± 4.4					33.7 ± 4.5	417	1
1 - 2 %		32.0 ± 5.1	29.1 ± 1.6				32.1 ± 5.0	599	4
2 - 3 %		31.1 ± 5.2	27.5 ± 5.7	22.9 ± 5.1			27.9 ± 5.8	575	52
3 - 4 %			26.1 ± 5.7	23.3 ± 5.1			24.2 ± 5.5	548	159
4 - 5 %				20.7 ± 5.4	18.7 ± 5.4		20.0 ± 5.5	272	325
5 - 6 %				15.3 ± 2.7	13.6 ± 7.1	3.6 ± 2.6	9.5 ± 7.5	92	756
$> 6\%$					12.0 ± 6.4	4.0 ± 4.2	4.1 ± 4.3	116	10940
cumulated	33.7 ± 4.5	31.9 ± 5.1	27.1 ± 5.7	22.1 ± 5.4	14.5 ± 7.0	4.0 ± 4.2	9.0 ± 10.5		
# Fail	430	658	699	553	177	102		2619	
# OK	1	4	77	345	641	11169			12237

Table 1. Average and standard deviation of C4.5 Error from Coverage and Average Term Coverage, for $m = 50$, $k = 10$. #OK (#Fail) is the number of learning problem instances with error less than (greater than) 20%.

These competence regions and failure regions, broadly observed for $P_{ac} > 5\%$ and $P_{ac} < 4\%$, are separated by a narrow region where the error rises abruptly. Further studies will investigate this transition. Finally, it appears that the coverage and the average term coverage together provide a good quality estimation of the C4.5 error, bounding *a priori* the generalisation error according to the coverage and average term coverage (Table 1). Specifically, for all problems with coverage less than 50%, an average term coverage less than 4% implies an error greater than 20% $((P_{ac} < 4\%) \Rightarrow ((Err > 20\%)$, with support 60% and confidence 90%), and reciprocally (with confidence 85%).

5.4. Sensitivity to noise

The sensitivity of C4.5 to noise was investigated, focusing on label noise[5]. The training sets used in the above experiments were corrupted by flipping the example labels with probability $\epsilon = 0, 1, ..20\%$. The test sets are unchanged. The above competence map (Fig. 7) demonstrates the known C4.5 robustness with respect to label noise. The predictive error smoothly increases with the data noise, almost linearly in the competence region (from 3% to 20% as ϵ goes from 0% to 20%) while it increases comparatively less in the failure region (from 35% to 41%).

6. Discussion and Perspectives

Indeed, several fundamental frameworks have been proposed for analysing the generalisation error from a theoretical perspective, ranging from PAC learning (Valiant, 1984; Kearns & Vazirani, 1994) to statistical and non-parametric learning (Vapnik, 1998; Devroye et al., 1996); these frameworks have been foundational for the development of new and powerful algorithms (Schapire, 1990; Schölkopf et al., 1998). Independently, many empirical studies have been undertaken to evaluate ML algorithms on artificial and real-world problems (Lim et al., 2000).

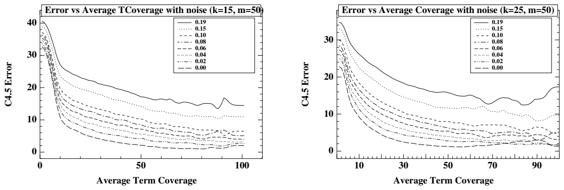

Figure 7. C4.5 Error vs Average Term Coverage with noise probability $\epsilon = 0, 1\%, \ldots, 10\%, 15\%, 19\%$, $k = 15$ (left) and 25 (right), $m = 50$, Gaussian parameter $K = 100$.

The work reported in this paper takes a different form, which is more familiar in empirical sciences, where principled experiments allow for gathering facts and organising them in such a way that non trivial regularities can be observed, and thereafter interpreted into a model of the phenomenon under study. Along these lines, we proposed a methodology for modeling a learning algorithm, accounting for the complex interactions between efficient learning heuristics, and specificities of the example distribution [6].

The model obtained through the competence map extensionally describes the algorithm behaviour, i.e. through look up tables. By exploiting (at the moment manually) these tables, some regularities are found.

A first result is that C4.5 does better on more general concepts in the experiment range, which appears *a posteriori* natural due to the greedy search bias effects. As an added value however, the competence map specifically localises the competence region to problems with coverage above 30%. A second finding regards the phase transition observed, and the steep rise of the er-

[5]The effects of the attribute noise are uneasily interpreted due to the complexity of the target concept.

[6]In the same spirit, a methodology based on ROC curves analysis is proposed in (Furnkranz & Flach, 2003) to compare the behaviour of diverse search criteria.

ror as the average term coverage decreases below 5%. Again, though this transition might be explained in the well-known framework of Small-Disjunct problems (Holte et al., 1989), the competence map brings in an added value as it shows precisely where the trouble begins.

This work opens up several perspectives. The proposed methodology must be confronted to other algorithms (e.g. CN2) and target concept spaces. In parallel, the wealth of data gathered about C4.5 behaviour will be better exploited, e.g. providing analytical models of the error and ideally identifying the deep causes for the failure cases. The abrupt transition of the error will be investigated with respect to additional order parameters of k-term DNF learning (e.g. probability for examples in distinct terms of admitting a correct lgg, not covering any negative training example) and with respect to the order statistics of the gain ratio criterion, inspired from Stoppiglia et al. (2003).

Modestly, the presented approach aims at a better understanding of the frontiers of ML algorithms, using an empirical approach to see *where the really hard problems are.*

Acknowledgments

The authors thank Antoine Cornuéjols, Céline Rouveirol, Erick Alphonse, Jacques Ales-Bianchetti and Mary Felkin for many discussions. Thanks are also due to the anonymous reviewers for their insights and suggestions.

References

Bensusan, H., & Kalousis, A. (2001). Estimating the predictive accuracy of a classifier. *Proc. ECML 2001* (pp. 25–36). Springer Verlag.

Blake, C., Keogh, E., & Merz, C. (1998). *UCI repository of machine learning databases.* http://www.ics.uci.edu/∼mlearn/MLRepository.html.

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.

Brazdil, P., Gama, J., & Henery, B. (1994). Characterizing the applicability of classification algorithms using meta-level learning. *Proc. ECML 1994* (pp. 83–102). Springer.

Chapelle, O., Weston, J., Bottou, L., & Vapnik, V. (2000). Vicinal risk minimization. *NIPS* (pp. 416–422).

Cheeseman, P., Kanefsky, B., & Taylor, W. (1991). Where the really hard problems are. *Proc. IJCAI 1991* (pp. 331–337).

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition.* New York: Springer.

Furnkranz, J., & Flach, P. (2003). An analysis of rule evaluation metrics. *Proc. ICML 2003* (pp. 202–209). Morgan Kaufmann.

Giordana, A., & Saitta, L. (2000). Phase transitions in relational learning. *Machine Learning*, *41*, 217–251.

Hogg, T., Huberman, B., & (Eds), C. W. (1996). *Artificial intelligence: Special issue on frontiers in problem solving: Phase transitions and complexity*, vol. 81(1-2). Elsevier.

Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*, 63–90.

Holte, R., Acker, L., & Porter, B. (1989). Concept learning and the problem of small disjuncts. *Proc. IJCAI 1989* (pp. 813–818). Morgan Kaufmann.

Kalousis, A. (2002). *Algorithm selection via meta-learning.* Doctoral dissertation, Université de Genève.

Kearns, M., & Vazirani, U. (1994). *An introduction to computational learning theory.* MIT Press.

Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, *40*, 203–228.

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Meta-learning by landmarking various learning algorithms. *Proc. ICML 2000* (pp. 743–750). Morgan Kaufmann.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Rückert, U., Kramer, S., & De Raedt, L. (2002). Phase transitions and stochastic local search in k-term dnf learning. *Proc. ECML 2002* (pp. 405–417). Springer Verlag.

Schapire, R. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197.

Schölkopf, B., Burges, C., & Smola, A. (1998). *Advances in kernel methods: Support vector machines.* MIT Press, Cambridge, MA.

Stoppiglia, H., Dreyfus, G., Dubois, R., & Oussar, Y. (2003). Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research*, *3*, 1399–1414.

Valiant, L. (1984). A theory of the learnable. *Communication of the ACM*, *27*, 1134–1142.

Vapnik, V. N. (1998). *Statistical learning theory.* Wiley.

Witten, I. H., & Frank, E. (1999). *Data mining: Practical machine learning tools and techniques with java implementations.* Morgan Kaufmann.

Wolpert, D., & Macready, W. (1995). *No free lunch theorems for search* (Technical Report). Santa Fe Institute.