A

Articles d'illustration, partie I

A.1 [Nock et al., 1998]

L'article présenté ici est la version rapport de recherche de l'article, aussi disponible sur le `CiteSeer` (`http ://citeseer.nj.nec.com/358435.html`).

Generalized Graph Colorability and Compressibility of Boolean Formulae

Richard Nock[1] Pascal Jappy[1] Jean Sallantin[1]

Abstract

In this paper, we study the existence of Occam's razors for a widely studied class of Boolean Formulae : Disjunctive Normal Forms (DNF). An Occam's razor is an algorithm which compresses the knowledge of observations (examples) in small formulae. We prove that approximating the minimally consistent DNF formula, and a generalization of graph colorability, is very hard. Other results on approximating Boolean formulae (and classical graph coloring) are negligible compared to ours. Our proof technique is such that the stronger the complexity hypothesis used, the larger the inapproximability ratio obtained. Our ratio is among the first to integrate the three parameters of Occam's razors : the number of examples, the number of description attributes and the size of the target formula labelling the examples. Theoretically speaking, our result rules out the existence of efficient deterministic Occam's razor algorithms for DNF. Practically speaking, it puts a large lower bound on the formulae's sizes found by learning systems proceeding by rule-searching.

[1]LIRMM, 161 rue Ada, 34392 Montpellier Cedex 5, France
e-mail:{nock, jappy, js}@lirmm.fr

1 Introduction

The learnability of Disjunctive Normal Form formulae (disjunctions of conjunctions) is a central problem in machine learning [Pillaipakkamnatt and Raghavan, 1994]. In 1984, Valiant studies the learnability of this class, and remarks that "*the attraction of this class is that humans appear to like it for representing knowledge as is evidenced, for example, by the success of the production system paradigm and of Horn clause logics*"[Valiant, 1985]. He proves that a subclass of DNF is learnable, and leaves as an open problem whether the whole class is learnable. Since then, many theoretical studies have investigated the learnability of DNF or subclasses [Kearns *et al.*, 1987; Pillaipakkamnatt and Raghavan, 1994; Aizenstein and Pitt, 1995; Aizenstein and Pitt, 1992; Berggren, 1993; Blum *et al.*, 1994; Bshouty *et al.*, 1995; Goldman and Mathias, 1992; Khardon, 1994; Mansour, 1992; Pitt and Valiant, 1988].

Simultaneously, many programs for machine learning were designed to learn efficiently from examples. In each of these programs, the algorithms has access to a learning sample and tries to build a small function approximating as best as possible the observed examples. According to [Cohen, 1995], systems that learn sets of rules (DNF in the Boolean framework) have a number of desirable properties : they are easy to understand, they can outperform decision-tree learning algorithms on many problems, they have a natural and familiar first-order version (Prolog predicates), and techniques for learning propositional rule sets can often be extended to the first-order case [Quinlan, 1990]. Many learning algorithms either build rules, or have a stage consisting in searching for rules that are postprocessed, or are aimed at producing formulae that can be easily translated into rules, [Brunk and Pazzani, 1991; Cohen, 1995; Cohen, 1993; de Raedt, 1992; Lavrac *et al.*, 1991; Muggleton and Feng, 1994; Nock and Gascuel, 1995; Pagallo and Haussler, 1990; Quinlan, 1995; Quinlan, 1994; Quinlan, 1990; Rouveirol, 1992; Thrun *et al.*, 1991; Wnek and Michalski, 1991] and many others.

Practical and theoretical results often focus on an aspect of approximation they both share : given a set of examples (or observations), can we devise an efficient algorithm which can find a small formula consistent with all examples ? Theoretically speaking, this aspect is often related to the principle of Occam's razors [Blumer *et al.*, 1987]: an Occam's razor for a class of concept representations C is an algorithm that, given a learning sample LS whose labels are given by some unknown target concept $t \in C$, can produce in time polynomial in $|LS|$, n (the number of description variables), $|t|$ a formula $h \in C$ satisfying to the two following conditions : h is consistent with LS and has size satisfying $|h| \leq |LS|^a(n|t|)^b$. The principle of Occam's razors [Kearns and Vazirani, 1994] states that in order to learn, a system should compress the information contained in the examples. This principle is remarkable because it was orignially stated by philosoph William of Occam (1285-1349), and led to theoretical results in the PAC-learning model of Valiant [Valiant, 1984] : learning is equivalent to finding Occam's razors [Kearns and Vazirani, 1994]. The principle of Occam can be observed in various ways, in authors such as Francis Bacon, John Locke, Blaise Pascal, Chomsky, Einstein, Hume, or even on work of art, such as "the nominal three - to William of Occam" (by Dan Flavin, National Gallery of Canada). In all that follows, we are interested by deterministic Occam's razors, still sufficient to ensure

2

learnability [Kearns and Vazirani, 1994].
Practically speaking, machine learning algorithms producing rules are almost always aimed at producing small sets of rules, because they are easy to understand for the non-expert, and they appear to be sufficient on many problems [Holte, 1993].

The principal result on the inapproximability of DNF comes from [Kearns *et al.*, 1987]. They show that DNF is as hard to approximate as Graph Colorability. [Feige and Kilian, 1996] prove that Graph Colorability is hard to approximate to within n^δ ($\forall 0 < \delta < 1$). Using the result of [Kearns *et al.*, 1987], this proves that DNF is not approximable to within n^δ ($\forall 0 < \delta < 1$), where n is the number of description variables of the examples, a measure of the problem's complexity. In this paper, we first prove that the upperbound of δ can be removed : the result holds in fact $\forall \delta > 0$. We go further into negative results, and prove that size-$|t|$ DNF cannot be approximated by DNF having size not greater than $|LS|^a n^b |t|^c$. a, b, c are any constants satisfying $\frac{1}{19} > a \geq 0$, $b \geq 0$, and $1 + \frac{1}{145} > c \geq 0$. $|t|$ is the number of monomials (conjunctions) of the target concept. This proof is stated under the hypothesis $NP \not\subset ZPP$, where ZPPdenotes the class of langages decidable by a random expected polynomial-time algorithm that makes no errors [Balcazar *et al.*, 1988].

In order to achieve our result, we firstly prove an equivalence of approximating DNF with a generalization of graph colorability. We then prove our result on the inapproximability of DNF by proving an inapproximability result on the generalization of Graph Colorability.

2 Equivalence between approximating DNF and a generalization of graph colorability

In this paper, we are interested in approximating optimization problems. An optimization problem contains an instance, the definition of a feasible solution, and a cost function defined for any feasible solution. The aim of any approximation algorithm is to find feasible solutions whose cost (*e.g.* number of colors for coloring a graph) is as close as possible from the problem's optimum. We also use the cost notion for instances : the cost of an instance is the optimal cost among all feasible solutions for this instance.

Let $\mathcal{F}$ be a class of Boolean formulae; any of its elements, f, is a function $f : \{0,1\}^n \rightarrow \{0,1\}$. An element $x \in \{0,1\}^n$ is an *example*. It is composed of n binary *variables* $\{x_1, ..., x_n\}$ assigned in $\{0,1\}$ (the negative and positive *literals*). The value $f(x)$ is the *class* $\in \{0,1\}$ that f assigns to x (the *negative* and *positive* class). The size of any formula f is denoted $|f|$. We investigate the class of DNF, set of formulae described as a disjunction of monomials. A monomial is a conjunction ($\wedge$) of literals (a literal is a Boolean descriptor, taking value either True or False). We are interested by the possibility, for some efficient (Ptime) algorithm, to approximate the following minimization problem:

- **Name** : *Opt*(DNF)

- **Instance** : A learning sample LS
- **Feasible Solutions** : Formulae from DNF consistent with LS
- **Cost Function** : Size of the formula (number of monomials)

It is well-known [Kearns *et al.*, 1987] that this problem is as hard as the problem *Opt* (Independant-set cover) (this is the same as the graph-coloring problem [Garey and Johnson, 1979]; however, this name is convenient for our proofs). We show in this paper that it is in fact as hard as a generalization of the *Opt*(Independant-set cover) problem (for any integer $k > 0$, $[k]$ denotes the set $\{1, 2, ..., k\}$):

Definition 1 ***Opt*(Multi independent-set cover)**

- **Name** *: Opt(Multi independent-set cover)*
- **Instance** *: $G^{\oplus} = (X^{\oplus}, E^{\oplus})$, a graph presenting the following form : for some positive integer d, $X^{\oplus}$ is partitionned into $X_1, ..., X_d$ such that if $d > 1$, $\forall 1 \leq i < j \leq d$, none of the edges between X_i and X_j are in $E^{\oplus}$.*
- **Feasible Solutions** *: a cover of $X^{\oplus}$ in subsets $s_1, ..., s_k$ such that :*
  1. *$\forall 1 \leq i \leq k, \forall 1 \leq j \leq d$, $s_i \cap X_j \neq \emptyset$ and induces an independent set in X_j.*
  2. *$\forall (x_1, x_2, ..., x_d) \in \prod_{i=1}^{d} X_i, \exists j \in [k] : \forall l \in [d], x_l \in s_j$*
- **Cost Function** *: k, that is, the number of subsets used to cover $X^{\oplus}$.*

Note that every graph can be represented according to the preceeding definition, but the uniqueness of the representation is not ensured for a given graph. The cost of a graph G instance of *Opt*(Independant-set cover) is usually written $\chi(G)$. We note $\chi_g(G^{\oplus})$ as the cost of a graph $G^{\oplus}$ instance of *Opt*(Multi independent-set cover). The following proposition states the equivalence between these two problems.

Proposition 1 Equivalence of approximating *Opt*(Multi independent-set cover) and *Opt*(DNF) : *For any graph $G^{\oplus} = (X^{\oplus}, E^{\oplus})$ instance of Opt(Multi independent-set cover), we can create in time polynomial in $|X^{\oplus}|$ a set of examples LS such that if there exists a feasible solution to Opt(Multi independent-set cover) whose cost is k, then we can create in Ptime a DNF having no more than k monomials and consistent with LS. Reciprocally, if there exists a DNF of size k consistent with LS, then (i) we can suppose without loss of generality that it is monotonous (no negative literals), and (ii) we can generate in Ptime a feasible solution to Opt(Multi independent-set cover) whose cost does not exceed k.*

(Proof left in appendix).

3 *Opt*(Multi independent-set cover) is hard to approximate

This section is devoted to the proof of the following theorem.

Theorem 1 *Unless NP⊆ZPP, Opt(Multi independent-set cover) is not approximable to within*

$$\rho^{\oplus} = \left(\sup_{1\leq i\leq n}\{|E_i|+|V_i|\}\right)^{da}\left(d\sup_{1\leq i\leq n}\{|V_i|\}\right)^{b}(\chi_g(G^{\oplus}))^{c-1}$$

where $G_i = (X_i, E_i)$ is the subgraph of $G^{\oplus}$ induced by X_i. The result holds

- $\forall b \geq 0$,
- $\forall \frac{1}{19} > a \geq 0$,
- $\forall 1 + \frac{1}{145} > c \geq 0$.

This theorem means that no polynomial-time (Ptime) algorithm can guarantee to find, from an instance $G^{\oplus}$ of *Opt*(Multi independent-set cover), a solution whose cost does not exceed $\chi_g(G^{\oplus}) \times \rho^{\oplus}$. The proof technique basically relies on multiplying d instances (or stairs) of *Opt*(Independant-set cover) to form an instance of *Opt*(Multi independent-set cover), without linking each stair to the others. Ideally, we would like to obtain a relationship such as

$$\chi_g(G^{\oplus}) = \chi(G)^d \tag{1}$$

which would ease a lot the proof since it would also blow up any inapproximability ratio ρ for *Opt*(Independant-set cover) to $\rho^{\oplus} = \rho^d$ for *Opt*(Multi independent-set cover). However, this relationship is *not* true for any graph instance of *Opt*(Independant-set cover), that is why we need to build particular, very hard to solve instances of *Opt*(Independant-set cover). Figure 1 presents the building of a hard instance G of *Opt*(Independant-set cover), from two reductions τ_1 and τ_2. Reduction τ_3^d is the stacking-up of d instances of *Opt*(Independant-set cover) to form an instance of *Opt*(Multi independent-set cover). The main theorem we need in this part is the following:

Theorem 2 *(From [Håstad, 1996]) Unless NP⊆ZPP, $\forall 0 \leq \delta < 1$, Opt(Independent set) is not approximable to within $\rho'' = n(G'')^{\delta}$.*

We suppose for the sake of simplicity, as it is pointed out in [Feige and Kilian, 1996; Håstad, 1996], that theorem 2 is proven from the decision problem "SAT". It means that there exists a reduction from "SAT" to *Opt*(Independent set) such that

- Any graph G'' obtained from a satisfiable instance of "SAT" satisfies $\alpha(G'') = g$.
- Any graph G'' obtained from an unsatisfiable instance of "SAT" satisfies $\alpha(G'') < g/n(G'')^{\delta}$, $\forall 0 < \delta \leq 1$.

We now describe the two first reductions, τ_1 and τ_2. Define as K_r the complete graph over r vertices.

5

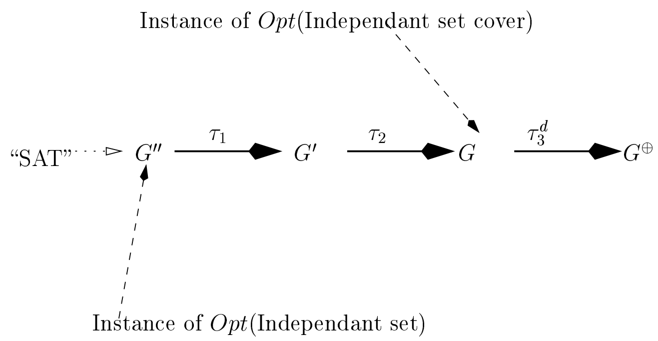

Figure 1: Scheme of the reduction $\tau_1 \circ \tau_2 \circ \tau_3^d$.

Definition 2 τ_1 **[Lund and Yannakakis, 1993]** *From G'', we create the graph product $G' = K_r \times G''$ with $r \geq \alpha(G'')$ as follows. Each vertex of G' is a pair $< i, v >$ where $i = 1, ..., r$ is a vertex from K_r and v is a vertex from G''. Sets of vertices with the same first component induce a clique in G'. Two vertices $< i, v >, < j, w >$ from different cliques are adjacent iff their second components v and w are equal or adjacent vertices in G''.*

(In that follows, we fix $r = n(G'')$). As pointed out in [Lund and Yannakakis, 1993], Section 2.1, we have $\alpha(G'') = \alpha(G')$. Let Z_p be the field of integers modulo p.

Definition 3 τ_2 **[Lund and Yannakakis, 1993]** *Let G' be a graph whose vertices are partitionned into cliques $C_1, ..., C_r$. Let p be a prime at least as large as*

$$\sup\{\sup_i |C_i|, r/\alpha(G'), \sqrt{r}\}$$

The image of G' by τ_2, G, has vertices described as quadruplets $< i, k, y, w >$, where $i = 1, ..., r$, $k \in Z_p$, $y \in Z_p^2$, and $w \in Z_p^2$. Let $u_A =< i_A, k_A, y_A, w_A >$ and $u_B =< i_B, k_B, y_B, w_B >$ be two vertices of G. The two vertices are **not** *adjacents iff $(\exists s \in Z_p)(\exists z \in Z_p^2)$ such that:*

- *$< i_A, k_A - s > \in G'$, $< i_B, k_B - s > \in G'$ and they are not adjacent.*
- *$w_A = (k_A - s)y_A + z$ and $w_B = (k_B - s)y_B + z$.*

Like [Lund and Yannakakis, 1993], we apply transformations τ_1 and τ_2 to any graph G'' instance of *Opt*(Independent set). We have

Proposition 2 *G satisfies:*

1. *$n(G) = rp^5$ ([Lund and Yannakakis, 1993], part 2.1)*
2. *$\alpha(G) = p^2\alpha(G')$ ([Lund and Yannakakis, 1993], corollary 2.2)*

6

3. *If G is built from a satisfiable instance of "SAT", then $\chi(G) = n(G)/\alpha(G)$ ([Lund and Yannakakis, 1993], theorem 2.3)*
4. $p^2\chi(G) \leq n(G)$

(proof of [4] is left in the appendix). The third reduction, τ_3^d, with $d > 0$ an integer, simply consists in stacking-up d times G without linking each "stair" to the others. Let $G^\oplus = \tau_3^d(G)$. Since there are $n(G)^d$ d-tuples containing one vertex from each $X_1, ..., X_d$, since any set from any solution to Opt(Multi independent-set cover) contains at most $\alpha(G)^d$ of these d-tuples, and since any of these d-tuples are covered, it comes $\chi_g(G^\oplus) \geq \frac{n(G)^d}{\alpha(G)^d}$. Furthermore, making the d-times cross-product of the independent sets of a solution to Opt(Graph colorability) lead to a feasible solution to Opt(Multi independent-set cover) whose cost satisfies $\chi_g(G^\oplus) \leq \chi(G)^d$. Consequently,

$$\frac{n(G)^d}{\alpha(G)^d} \leq \chi_g(G^\oplus) \leq (\chi(G))^d \tag{2}$$

We refine these inequations. Any graph $G^\oplus$ corresponding to a satisfiable instance of "SAT" leads by proposition 2 to $\chi(G) = n(G)/\alpha(G)$. Therefore, $\chi_g(G^\oplus) = \chi(G)^d$. Any graph $G^\oplus$ corresponding to an unsatisfiable instance "SAT" leads by theorem 2 and proposition 2 to $\forall 0 \leq \delta < 1, \chi_g(G^\oplus) > \left(p^3 n(G'')^\delta\right)^d$. From proposition 2, we also get $\forall 0 \leq \delta < 1, p^3 n(G'')^\delta \geq \frac{\chi(G)p^5 n(G'')^\delta}{n(G)}$. But $n(G)/p^5 = r = n(G'')$. Therefore $\forall 0 \leq \delta < 1, p^3 n(G'')^\delta \geq \frac{\chi(G)}{n(G'')^{1-\delta}}$. Therefore, for any graph $G^\oplus$ corresponding to instances of "SAT" either satisfiable or not, we have:

$$\forall 0 \leq \delta < 1, \left(\frac{\chi(G)}{n(G'')^{1-\delta}}\right)^d \leq \chi_g(G^\oplus) \leq (\chi(G))^d \tag{3}$$

This relationship is central for our proof; although it is much weaker than equation 1, it is still sufficient to prove theorem 1. However, there are two problems left : how can we use 3 to prove $\rho^\oplus$, and can we choose d constant, so that τ_3^d is Ptime ? We first solve the first problem. From any "SAT" instance (theorem 2) transformed in a graph G using reductions τ_1 and τ_2, depending on whether it is satisfiable or not, there exists $g > 0$ (it is a function of the "SAT" instance, [Lund and Yannakakis, 1993]) such that either $\chi(G) = p^3$, or $\chi(G) > p^3 n(G'')^\delta$, $\forall 0 < \delta \leq 1$.
For any satisfiable instance of "SAT", we get from inequations 3 : $\chi_g(G^\oplus) \leq \chi(G)^d = (p^3)^d$.
For any unsatisfiable instance of "SAT", we get from inequations 3 : $\chi_g(G^\oplus) \geq \left(\frac{\chi(G)}{n(G'')^{1-\delta}}\right)^d \geq (p^3)^d \times (n(G'')^{2\delta-1})^d$. In order to prove theorem 1, it is sufficient to show a "gap" greater than $\rho^\oplus$ between $(p^3)^d$ and $(p^3)^d \times (n(G'')^{2\delta-1})^d$. That is, we need to find d such that

$$(n(G'')^{2\delta-1})^d > \rho^\oplus \quad \Rightarrow \quad d > \frac{\log \rho^\oplus}{\log(n(G'')^{2\delta-1})} \tag{4}$$

under the constraint $\delta > 1/2$. And we need to show that d is constant.

7

We now check the constant value for d satisfying inequation 4. [Lund and Yannakakis, 1993] choose for p a prime at least as large as $\sup\left\{\sup_i |C_i|, \frac{r}{\alpha(G'')}, \sqrt{r}\right\}$, which is $n(G'')$ in our case. We also have

Proposition 3 [de la Higuera, 1996] *There exists a constant $0 < \alpha < 1$ such that for any positive integer M, there exists a prime falling in the interval $[M; M^{1+\alpha}]$. Furthermore, we can fix $\alpha = 11/20$.*

In our case, we can therefore suppose that $n(G'') \leq p \leq n(G'')^{1+\alpha}$. We fix $\frac{49}{50} < \delta < 1$ if $c \leq 1$, and $\frac{49}{50} + \frac{113(c-1)}{40} < \delta < 1$ otherwise (remark that $\frac{1}{2} < \delta < 1$). Fix

$$d = \lceil \frac{f(a,b,c,\alpha,\delta) + (6+5\alpha)b}{2\delta - 1} \rceil \tag{5}$$

with

$$f(a,b,c,\alpha,\delta) = 2 \times \left[\frac{1}{a(12+10\alpha) + \max\{0; (c-1)(4+3\alpha)\}} - \frac{1}{2\delta - 1}\right]^{-1} \times \left[\frac{(6+5\alpha)b}{2\delta - 1} + 1\right]$$

Arithmetic calculation gives

Fact 1 *The choice of δ leads that $f(a,b,c,\alpha,\delta)$ is positive. Furthermore,*

$$f(a,b,c,\alpha,\delta) - d\left(a(12+10\alpha) + \max\{(c-1)(4+3\alpha); 0\}\right) > 0$$

Fact 1 leads to $d^b < n(G'')^{f(a,b,c,\alpha,\delta) - d(a(12+10\alpha)+(c-1)(4+3\alpha))}$, at least for sufficient large-sized graphs. Proposition 2 leads to

$$\chi_g(G^{\oplus}) \leq \chi(G)^d \leq \left(\frac{n(G)}{p^2}\right)^d = (rp^3)^d \leq n(G'')^{d(4+3\alpha)} \tag{6}$$

The fact that vertices of G can be partitionned into independent sets of size p^2 (proposition 2) leads to (for convenience, we fix $e(G)$ and $n(g)$ to be respectively the number of edges and the number of vertices of G) : $e(G) \leq p^4 \frac{(rp^3)!}{(rp^3-2)!2!} = \frac{rp^7(rp^3-1)}{2}$. From this and the fact that $n(G) = rp^5 \leq n(G'')^{6+5\alpha}$, we get $e(G) + n(G) \leq \frac{rp^7(rp^3-1)}{2} + rp^5$, and therefore $e(G) + n(G) < r^2p^{10} = n(G'')^{12+10\alpha}$. Putting it altogether, we get that

$$\rho^{\oplus} = \left(\sup_{1 \leq i \leq n}\{|E_i| + |V_i|\}\right)^{da} \left(d \sup_{1 \leq i \leq n}\{|V_i|\}\right)^b (\chi_g(G^{\oplus}))^{c-1}$$

implies whenever $c \geq 1$

$$\begin{aligned}\rho^{\oplus} < \ & n(G'')^{(12+10\alpha)ad} \times n(G'')^{f(a,b,c,\alpha,\delta)-(12+10\alpha)ad-d(c-1)(4+3\alpha)} \\ & \times n(G'')^{(6+5\alpha)b} \times n(G'')^{d(c-1)(4+3\alpha)}\end{aligned}$$

8

and therefore $\rho^{\oplus} < n(G'')^{f(a,b,c,\alpha,\delta)+(6+5\alpha)b}$. When $c < 1$, similar calculation leads again to $\rho^{\oplus} < n(G'')^{f(a,b,c,\alpha,\delta)+(6+5\alpha)b}$. We get

$$\frac{\log \rho^{\oplus}}{\log(n(G'')^{2\delta-1})} < \frac{f(a,b,c,\alpha,\delta)+(6+5\alpha)b}{2\delta-1} \tag{7}$$

But the choice of d also gives

$$\frac{f(a,b,c,\alpha,\delta)+(6+5\alpha)b}{2\delta-1} \leq d \tag{8}$$

d is therefore constant, and satisfies inequation 4. The proof of theorem 1 is completed. From the reduction in proof of proposition 1 exposed in the appendix, from the fact that $|LS| \leq \left(\sup_{1\leq i\leq n}\{|E_i|+|V_i|\}\right)^d$, $n \leq d \sup_{1\leq i \leq n}\{|V_i|\}$, and the target concept's size satisfies $|t| = \chi_g(G^{\oplus})$, we obtain

Theorem 3 Non-approximability of DNF *Unless $NP \subseteq ZPP$, there cannot exist an Occam's razor for DNF finding formulae whose size does not exceed $|LS|^a n^b |t|^c$, where $|t|$ is the size of the target concept, and $|LS|$ the size of the learning sample. The result is true even if we suppose that the target concept belongs to monotone-DNF. a, b, c are any constants satisfying:*

$$\frac{1}{19} > a \geq 0 \ ; \ b \geq 0 \ ; \ 1 + \frac{1}{145} > c \geq 0$$

4 Conclusion

Recall that an Occam's razor for a class of Boolean formulae C is an algorithm that, given a learning sample LS whose labels are given by some unknown target concept $t \in C$, can produce in time polynomial in $|LS|$, n, $|t|$ a formula $h \in C$ satisfying to the two following conditions : h is consistent with LS and $|h| \leq |LS|^a(n|t|)^b$, with $a, b > 0$ and $a < 1$. With our complexity hypothesis, our result does not rule out any possibility of deterministic Occam's razors, even if efficient Occam's razors are proven impossible. However, the advantage of our reduction technique is that, the higher the time alloted for τ_3^d (thus, the higher d), the higher the non-approximability ratio. [Pillaipakkamnatt and Raghavan, 1994] cite the complexity hypothesis $NP \not\subseteq \mathrm{DTIME}(poly(2^{n^\epsilon}))$ (for some $\epsilon > 0$). With such hypotheses, involving zero-error probabilistic time algorithms, our technique almost removes *any* possibility of Occam's razors for DNF.

References

[Aizenstein and Pitt, 1992] H. Aizenstein and L. Pitt. Exact learning of read-k-disjoint DNF and not-so-disjoint-DNF. In *Proc. of the 5 th International Conference on Computational Theory*, pages 71–76, 1992.

[Aizenstein and Pitt, 1995] H. Aizenstein and L. Pitt. On the learnability of Disjunctive Normal Form formulas. *Machine Learning*, 19:183–208, 1995.

[Balcazar *et al.*, 1988] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity I.* Springer Verlag, 1988.

[Berggren, 1993] U. Berggren. Linear time deterministic learning of k-term-DNF. In *Proc. of the 6 th International Conference on Computational Theory*, pages 37–40, 1993.

[Blum *et al.*, 1994] A. Blum, R. Khardon, E. Kushilevitz, L. Pitt, and D. Roth. On learning read-k-satisfy-j DNF. In *Proc. of the 7 th International Conference on Computational Theory*, pages 110–117, 1994.

[Blumer *et al.*, 1987] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, pages 377–380, 1987.

[Brunk and Pazzani, 1991] C. Brunk and M. Pazzani. Noise-tolerant relational concept learning. In *Proc. of the 8 th International Conference on Machine Learning*, 1991.

[Bshouty *et al.*, 1995] N. H. Bshouty, Z. Chen, S. E. Decatur, and S. Homer. On the learnability of z_n-DNF formulas. In *Proc. of the 8 th International Conference on Computational Theory*, pages 198–205, 1995.

[Cohen, 1993] W. W. Cohen. PAC-learning a restricted class of recursive logic programs. In *Proc. of AAAI-93*, pages 86–92, 1993.

[Cohen, 1995] W. W. Cohen. Fast effective rule induction. In *Proc. of the 12 th International Conference on Machine Learning*, pages 115–123, 1995.

[de la Higuera, 1996] C. de la Higuera. Characteristic sets for polynomial grammatical inference, 1996. Draft accepted to Machine Learning.

[de Raedt, 1992] L. de Raedt. Iterative concept learning and construction by analogy. *Machine Learning*, pages 107–150, 1992.

[Feige and Kilian, 1996] U. Feige and J. Kilian. Zero knowledge and the chromatic number. draft, 1996.

[Garey and Johnson, 1979] M.R. Garey and D.S. Johnson. *Computers and Intractability, a guide to the theory of NP-Completeness.* Bell Telephone Laboratories, 1979.

[Goldman and Mathias, 1992] S. A. Goldman and H. D. Mathias. Learning k-term-DNF formulas with an incomplete membership oracle. In *Proc. of the 5 th International Conference on Computational Theory*, pages 77–84, 1992.

[Håstad, 1996] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *FOCS'96*, pages 627–636, 1996.

[Holte, 1993] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, pages 63–91, 1993.

10

[Kearns and Vazirani, 1994] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory.* M.I.T. Press, 1994.

[Kearns *et al.*, 1987] M.J. Kearns, M. Li, L. Pitt, and L. Valiant. On the learnability of boolean formulae. *Proceedings of the Nineteenth Annual A.C.M. Symposium on Theory of Computing*, pages 285–295, 1987.

[Khardon, 1994] R. Khardon. On using the fourier transform to learn disjoint DNF. *Information Processing Letters*, pages 219–222, 1994.

[Lavrac *et al.*, 1991] N. Lavrac, S. Dzeroski, and M. Grobelnik. Learning non-recursive definitions of relations with linus. In *European Working Session in Learning*, 1991.

[Lund and Yannakakis, 1993] K. Lund and M. Yannakakis. On the hardness of approximating minimization problems. In *Proc. of the 25 th Symposium on the Theory of Computing*, pages 286–293, 1993.

[Mansour, 1992] Y. Mansour. An $O(n^{log\ log\ n})$ algorithm for dnf under the uniform distribution. In *Proc. of the 5 th International Conference on Computational Theory*, pages 53–61, 1992.

[Muggleton and Feng, 1994] S. Muggleton and C. Feng. Efficient induction of logic programs. In *Inductive Logic Programming*, 1994.

[Nock and Gascuel, 1995] R. Nock and O. Gascuel. On learning decision committees. In *Proc. of the 12 th International Conference on Machine Learning*, pages 413–420, 1995.

[Pagallo and Haussler, 1990] J. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 1990.

[Pillaipakkamnatt and Raghavan, 1994] K. Pillaipakkamnatt and V. Raghavan. On the limits of proper learnability of subclasses of DNF formulae. In *Proc. of the 7 th International Conference on Computational Theory*, pages 118–129, 1994.

[Pitt and Valiant, 1988] L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *J. ACM*, pages 965–984, 1988.

[Quinlan, 1990] J. R. Quinlan. Learning logical definition from relations. *Machine Learning*, pages 239–270, 1990.

[Quinlan, 1994] J. R. Quinlan. *C4.5 : programs for machine learning.* Morgan Kaufmann, 1994.

[Quinlan, 1995] J. R. Quinlan. MDL and categorical theories (continued). In *Proc. of the 12 th International Conference on Machine Learning*, pages 464–470, 1995.

[Rouveirol, 1992] C. Rouveirol. ITOU: induction of first-order theories. *Inductive Logic Programming*, 1992.

[Thrun *et al.*, 1991] S. B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Dzeroski, S. E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R. S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich, H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, and J. Zhang. The MONK's problems: a performance comparison of different lear ning algorithms. Technical Report CMU-CS-91-197, Carnegie Mellon University, 1991.

[Valiant, 1984] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.

[Valiant, 1985] L. G. Valiant. Learning disjunctions of conjunctions. In *Proc. of the 9 th IJCAI*, pages 560–566, 1985.

[Wnek and Michalski, 1991] J Wnek and R. Michalski. Hypothesis-driven constructive induction in AQ17. In *Proc. of the 12 th IJCAI*, 1991.

5 Appendix

Proof of proposition 1.

Let $G^{\oplus} = (X^{\oplus}, E^{\oplus})$ a given graph instance of Opt(Multi independent-set cover). For any $G_i = (X_i, E_i)$ subgraph of $G^{\oplus}$ induced by X_i, we create two sets of word over $\{0,1\}^{|X_i|}$: let

- $N_i = \{\vec{0}_{j,k}, \forall (j,k) \in E_i\}$ ($\vec{0}_{j,k}$ is the only-"1" vertor having zeroes at indices i and j).
- $P_i = \{\vec{0}_j, \forall j \in [|X_i|]\}$ ($\vec{0}_j$ is the only-"1" vertor having one "0" at index j).

Let $LS_i = P_i \cup N_i$. Examples are the elements of the set $LS = \times_{j=1}^{d} LS_j$. $e = \{e_1, ..., e_d\}$ is a positive example iff $\forall 1 \leq i \leq d, e_i \in P_i$. The class of any example is therefore given by a $\wedge$-like rule.
Suppose that there exists a feasible solution $s' = \{s'_1, ..., s'_k\}$ to Opt(Multi independent-set cover), whose cost is k. Each s'_i représents a partial cover by independent sets of $X^{\oplus}$. For any set s'_i of s', create a monotonous monomial having "*" in positions corresponding to the vertices of s'_i. This monomial is not satisfied by any negative example. Otherwise, s'_i would not represent a partial cover by independent sets of $X^{\oplus}$. Indeed, each of the edges of the negative example (there exists at least one because of the definition of the negative examples) would lie in the subgraph of $G^{\oplus}$ induced by s'_i, so there would exist a $j \in [d]$ such that the subgraph induced by $s'_i \cap X_j$ is not an independent set, which is impossible by hypothesis. Furthermore, the d-tuples of $\prod_{i=1}^{d} X_i$ are in bijection with positive examples, and for each of these d-tuples that is covered by a set s'_j for example corresponds a positive example satisfying the monomial created from s'_j. The DNF constructed is consistent with LS, and it contains k monomials, as claimed.
Suppose that there exists a DNF f consistent with LS. For any monomial T_i of f, note it $T_i = \left(T_i^1, ..., T_i^d\right)$, which means that part T_i^j is described over the set of variables that were created from X_j. Suppose without loss of generality that it monotonous, since otherwise, $\forall 1 \leq j \leq d$,

12

if T_i^j is not monotonous, then it contains at most one "0" (otherwise, it cannot be satisfied by any positive example), and we can replace T_i^j by the monotonous word over $\{*, 1\}^{|X_j|}$ having only one "*" in the position of the "0". By this way, we keep the consistency of f. Now, create s', feasible solution to Opt(Multi independent-set cover), by constructing for each monomial T_i of f a set s'_i obtained by grouping all vertices corresponding to the "*" of T_i. Each s'_i induces an independent set in any subgraph of $G^{\oplus}$ induced by $s'_i \cap X_j, \forall j \in [d]$. Indeed, suppose that this not the case, and let $s'_i = \{s'^1_i, ..., s'^d_i\}$ which means that the vertices of s'^j_i belong to X_j ; there exists a $j \in [d]$ such that s'^j_i is not an independent set. So, two of its vertices are linked by an edge. Create the following word over $\{0,1\}^{\sum_{i=1}^{d}|X_i|}$ described respectively over variables from X_1, X_2, ..., X_d in this order:

- $\forall k \in [d] \backslash j$, part k is described over $\{0,1\}^{|X_k|}$, it contains $|X_k| - 1$ "1", and one "0" corresponding to an aribtrary element of s'^k_i.
- part j is described over $\{0,1\}^{|X_j|}$, it contains $|X_j| - 2$ "1", and two "0" corresponding to two vertices of s'^j_i linked by an edge in X_j.

The word created is a negative example, and it satisfies T_i. f is inconsistent, which is impossible by hypothesis. Furthermore, since any positive example satisfies a monomial of f, we clearly obtain $\forall (x_1, x_2, ..., x_d) \in \prod_{i=1}^{d} X_i, \exists j \in [k] : \forall l \in [d], x_l \in s'_j$. The proof of proposition 1 is achieved.

Proof of [4] in proposition 2.

Obtained from definition 3 : the vertices of G can be partitionned into independent sets of size p^2. Take a vertex v in G''. For any $i = 1, ..., r$, define the following subgraphs of G having p^2 vertices each.

- $G_{<i,v>}$ is induced by vertices $\{< i, v, y, vy >: y \in Z_p^2\}$
- for any $z \in Z_p^2 \backslash (0,0)$, $Shift(G_{<i,v>}, z)$ is induced by the set $\{< i, v, y, w+z >:< i, v, y, w > \in G_{<i,v>}\}$

It comes that the vertices of G are exactly the vertices of

$$\cup_{v \in G''; i=1,...,r} \left(G_{<i,v>} \cup_{z \in Z_p^2 \backslash (0,0)} Shift(G_{<i,v>}, z) \right) \tag{9}$$

Consider definition 3. Putting $s = 0$ and $z = (0,0)$, it comes that $G_{<i,v>}$ satisfies conditions of definition 3 and is therefore not linked. Putting $s = 0$ and z from $Shift(G_{<i,v>}, z)$, it comes that any couple of vertices of $Shift(G_{<i,v>}, z)$ satisfies conditions of definition 3 and is not linked. It comes that G is always $n(G)/p^2$ colorable, and $\chi(G) \leq n(G)/p^2$, as claimed. The proof of [4] in proposition 2 is achieved.

A.2 [Nock and Jappy, 1998a]

Function-free Horn Clauses are Hard to Approximate

Richard Nock and Pascal Jappy

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier,
161, rue Ada,
34392 Montpellier, France
{nock,jappy}@lirmm.fr

Abstract. In this paper, we show two hardness results for approximating the best function-free Horn clause by an element of the same class. Our first result shows that for some constant $k > 0$, the error rate of the best k-Horn clause cannot be approximated in polynomial time to within any constant factor by an element of the same class. Our second result is much stronger. Under some frequently encountered complexity hypothesis, we show that if we replace the constant number of Horn clauses by a small, poly-logarithmic number, the constant factor blows up exponentially to a quasi-polynomial factor $n^{\log^k n}$, where n is the number of predicates of the problem, a measure of its complexity. Our main result links the difficulty of error approximation with the number of clauses allowed. We finally give an outline of the incidence of our result on systems that learn using ILP (Inductive Logic Programming) formalism.

1 Introduction and motivation

ILP is an active research branch at the crossroads of of Machine Learning and Logics. It aims at learning concepts expressed as (variously) restricted Horn Clause Programs from examples, and in the presence of background knowledge. Many experimental applications are available, that have been applied to domains such as biology, chess playing and natural langage analysis. Theoretical work has allowed to establish learnability results for some subclasses of first order Horn clauses. Early studies were undertaken in the Identification in the limit model [7], but most work has focused on Approximately Correct (PAC) learnability [15], [10] which is thought to better quantify the complexity of learning in terms of computational effort and number of examples required. In ILP, this latter problem is intractable for very general classes such as unconstrained Horn clauses (see [11] for a detailed presentation of computational hardness results). So, in order to achieve positive results, several restrictions of Horn Clause programs have been considered [13], [4], [5], and [6].

However, conflicts between PAC results and practical ones have led researchers to look for other learnability models [12]. In a previous paper, we highlighted divergences between PAC and robust learning [8] results for some of the main

ILP classes. Whereas PAC learning makes the strong assumption that any target concept can be represented in the hypothesis class $\mathcal{H}$, (which is very rarely acceptable in practice), robust learning studies the degradation in prediction performance of a hypothesis class $\mathcal{H}$ when it is not known a priori whether it contains the target concept's class. This makes this model a stricter one but it is closer to practical requirements. The commonpoint to both PAC and robust learning models is the sufficiency of worst case analyses to obtain negative results. Our result in [9] states that, even when considering a simple subclass of ILP formalism and even when looking for a single Horn clause, no polynomial-time algorithm can produce a formula whose error comes close to the error of the optimal single Horn clause. In this paper, we go further in worst-case analyses. We show that the condition on the error can be replaced by a much weaker one without losing negative results. We show that no polynomial-time algorithm can produce a formula approximating the error of the optimal one to within very large factors. The rest of this paper is organised as follows: in section 2, we present the ILP background we need for our results, and the link between ILP and structural complexity. In section 3 and 4 we prove that approximating function-free Horn clause is hard. Finally, in section 5, we highligh some relevant subclasses of ILP formalism for which our results are valid.

2 An ILP approximation problem

For a complete formalization of the ILP background needed for this article, we refer the reader to [9]. Given a Horn clause langage $\mathcal{L}$ and a correct inference relation on $\mathcal{L}$, an ILP learning problem can be formalized as follows. Assume a background knowledge $\mathcal{BK}$ expressed in a langage $\mathcal{LB} \subseteq \mathcal{L}$, and a set of examples $\mathcal{E}$ in a langage $\mathcal{LE} \subseteq \mathcal{L}$. The goal is to produce a hypothesis h in a hypothesis class $\mathcal{H} \subseteq \mathcal{L}$ consistent with $\mathcal{BK}$ and $\mathcal{E}$ such that h and the background knowledge cover all positive examples and none of the negative ones. The choice of the representation langages for the background knowledge and the examples, and the inference relation greatly influence the complexity (or decidability) of the learning problem. A common restriction for both $\mathcal{BK}$ and $\mathcal{E}$ is to use ground facts. As in [11], we use θ-subsumption as inference relation. Its main drawback being that it does not allow the use of background knowledge, other subsumption relations have been defined to do so, in particular generalized subsumption [2], and are thus preferred in ILP. We now state a useful lemma

Lemma 1 *Learning a Horn clause program from a set of ground background knowledge $\mathcal{BK}$ and ground examples $\mathcal{E}$, the inference relation being generalized subsumption, is equivalent to learning the same program with θ-subsumption, and empty background knowledge and examples defined as ground Horn clauses of the form $e \leftarrow b$, where $e \in \mathcal{E}$ and $b \in \mathcal{BK}$.*

This lemma allows us to incorporate the background knowledge in the new examples (and is thus empty). Examples and clauses are defined by predicates. To the variables of these predicates that are in the clauses built correspond constant symbols in the examples. θ-subsumption relative to the examples aims at finding adequate substitutions of variables with constant symbols. It can be defined in a general way as follows.

Definition 1 (θ-subsumption) *A clause C_1 θ-subsumes a clause C_2 if there exists a substitution θ such that $C_1\theta \subseteq C_2$.*

Our ILP problem can be presented as an optimization problem as follows, respecting the formalization of [8] :

Name : *Opt*(Weighted-Approx(g(.)-function-free Horn clauses)).
Instance : A set of examples $\mathcal{E} = S^+ \cup S^-$, an integer weight $w(x_i) > 0$ for each example $x_i \in S^+ \cup S^-$.
Feasible Solutions : $h \in g$(.)-function-free Horn clause.
Cost Function : $\sum_{(x \in S^+ \wedge h(x)=0) \vee (x \in S^- \wedge h(x)=1)} w(x)$.

g(.) is a function defining the maximum size (clause number) of the function-free Horn clauses constructed. It is worthwile remarking that a machine learning algorithm is ran practically on a set of examples often called the "learning sample", and aims at finding a low-error formula, without prior knowledge on the concept from which these examples were taken. Therefore, what does such an algorithm is trying to find a feasible solution to the previous problem having a low cost. Proving lower bounds on the costs of polynomial-time algorithms for this problem is therefore of practical interest.

3 Result on k-function-free Horn clauses

In this section we state and prove a first non-approximability result, dealing with small-sized function-free Horn clauses.

Theorem 1 *If $NP \not\subset P$, $\forall k \geq 3$, Opt(Weighted-Approx(k-function-free Horn clauses)) is not approximable to within any constant $d > 0$.*

We make a reduction from a minimization problem previously studied in [8]:

Name : *Opt*(Aggravated 3-SAT).
Instance : A set of variables $U = \{x_1, \overline{x}_1, ..., x_n, \overline{x}_n\}$, a collection of 3-clauses over U, a subset $U' \subseteq U$, an assignment satisfying all clauses.
Feasible Solutions : An assignation of the variables of U satisfying all clauses.
Cost Function : The number of variables from U' assigned to **True** .

The satisfiability constraint implies that a solution always exists; the difficulty of the problem therefore relies only is the minimization of the cost function, and not in finding feasible solutions, which would be an artifact of the problem's hardness. We use the following result on *Opt*(Aggravated 3-SAT):

Theorem 2 [8] *If $NP \not\subset P$, Opt(Aggravated 3-SAT) is not approximable to within any constant $d > 0$.*

In order to prove our result, we need to obtain an intermediate inapproximability result for the following minimization problem:

Name : *Opt*(Aggravated k-Colorability).
Instance : A graph $G = (X, E)$. A subset $X' \subset X$. A vertex $x \in X \backslash X'$. A trivial k-Colorability of G.
Feasible Solutions : A valid k-Colorability of G.
Cost Function : The number of elements of X' having the same color as x.

Due to space limitations, undetailed proofs can be found in [14].

Theorem 3 *If $NP \not\subset P$, Opt(Aggravated k-Colorability) is not approximable to within any constant $d > 0$.*

Proof sketch. The reduction is made from the problem *Opt*(Aggravated 3-SAT). $C = \{C_1, ..., C_p\}$ denotes the set of 3-clauses instance of *Opt*(Aggravated 3-SAT). We transform it into a new set $C' = \{C'_1, ..., C'_p, C'_{p+1}, C'_{p+2}, C'_{p+3}, C'_{p+4}\}$ in the following way : let $L = U = \{x_1, \overline{x}_1, ..., x_n, \overline{x}_n\}$ stands for the variables set of the 3-SAT instance. Let $L' = \{x_1, \overline{x}_1, ..., x_n, x_{n+1}, \overline{x}_{n+1}, x_{n+2}, \overline{x}_{n+2}, x_{n+3}, \overline{x}_{n+3}\}$ be our new set of variables, and define $\forall 1 \leq i \leq p, C'_i = C_i \vee x_{n+1}$, $C'_{p+1} = \overline{x}_{n+1} \vee x_{n+2} \vee x_{n+3}$, $C'_{p+2} = \overline{x}_{n+2} \vee x_{n+3}$, $C'_{p+3} = x_{n+2} \vee \overline{x}_{n+3}$, and $C'_{p+4} = \overline{x}_{n+2} \vee \overline{x}_{n+3}$. The length of each new clause in C' is either 2, 3 or 4. We now state a number of facts useful to prove theorem 3. We let $[n]$ denote the set $\{1, 2, ..., n\}$.

Fact 1 *C' satisfies two properties : (1) if C' is satisfiable, x_{n+1}, x_{n+2} and x_{n+3} are all* **False** *; (2) C is satisfiable iff C' is satisfiable.*

The graph G we construct, instance of *Opt*(Aggravated k-Colorability), has the modular structure which we now describe.
Step 1: each of the $2n + 6$ variables of L' is represented by a vertex in G:

$$\{x_1, \overline{x}_1, ..., x_n, \overline{x}_n, x_{n+1}, \overline{x}_{n+1}, x_{n+2}, \overline{x}_{n+2}, x_{n+3}, \overline{x}_{n+3}\}$$

We call these "variable vertices". We then place $n + 3$ edges $(x_i, \overline{x}_i), \forall i \in [n+3]$ in G. Each couple of variable vertices $(x_i, \overline{x}_i), i \in [n]$ represent the two possible truth assignment of the variables $x_i, \overline{x}_i$ of U: either **True** for x_i (and **False** for $\overline{x}_i$), or **False** for x_i (and **True** for $\overline{x}_i$).
Step 2: for each clause $C'_i, i \in [p+4]$, we create a corresponding subgraph H_i. Each of these subgraphs uses a basic buiding block shown in figure 1 (K_k is the complete graph on k vertices). In this figure, at least either of a or b is a variable vertex, and y is a new vertex. $\forall i \in [p+4]$, each clause C'_i of size 2, 3 or 4 is represented in G by a subgraph H_i using 1, 2 or 3 basic building blocks respectively. As shown in figure 1, to each clause C'_i corresponds a distinct vertex noted $y_{6,i}$. The only vertices shared by these subgraphs $H_i, i \in [p+4]$ are variable vertices.
Step 3: for each clause $C'_i, i \in [p+1]$, we create a corresponding subgraph H'_i isomorphic to H_i. $\forall i \in [p+1]$, the only vertices shared by H_i and H'_i are the variable vertices corresponding to C'_i. Vertices $y_{6,i}$ in H_i are renamed $y'_{6,i}$ in H'_i. Let $\chi(x)$ denote the color of any vertex x of G.

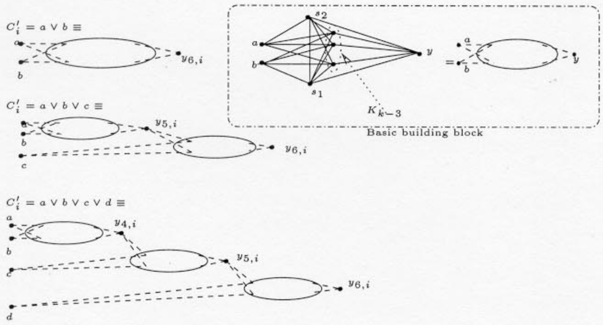

Fig. 1. Basic building block for $H_i, i \in [p+4]$, and subgraphs H_i generated by 2-, 3- and 4-clauses.

Fact 2 *$\forall i \in [p+4], \forall j \in [p+1]$, for any valid k-coloring of the vertices of H_i (resp. H'_j), there exists one which gives to $y_{6,i}$ (resp. $y'_{6,j}$) a color shared by either*

- *a or b if C'_i (resp. C'_j) is of size 2.*
- *a or b or c if C'_i (resp. C'_j) is of size 3.*
- *a or b or c or d if C'_i (resp. C'_j) is of size 4.*

Fact 3 *$\forall i \in [p+4], \forall j \in [p+1]$, any valid k-coloring of the vertices of H_i (resp. H'_j) assigning the same color to each variable vertex of the clause C'_i (resp. C'_j), forces $y_{6,i}$ (resp. $y'_{6,j}$) to have this color.*

Step 4: The subgraphs constructed in steps 1, 2 and 3 are linked to a subgraph W according to figure 2. This terminates the construction of G which now contains $6pk + 8k + 2n + 6$ vertices and is therefore of polynomial size. We have:

Fact 4 *$\chi(x_{n+1}) = \chi(v_2)$ and $\chi(\overline{x}_{n+1}) = \chi(v_1)$.*

Fact 5 *In a valid k-coloring of G, let $\chi_1, ..., \chi_{k-2}$ denote the set of colors used to color K_{k-2} in W, and $\chi_1, ..., \chi_k$ the total set of colors. We have*

$$\forall i \in [n+3], \{\chi(x_i), \chi(\overline{x}_i)\} = \{\chi_{k-1}, \chi_k\} = \{\chi(v_1), \chi(v_2)\}$$

Fact 6 *$\forall i \in [p]$, let $C'_i = x_{i_1} \vee x_{i_2} \vee x_{i_3} \vee x_{n+1}$ denote a 4-clause of C'. We have*

$$(\chi(x_{i_1}) = \chi(v_1)) \vee (\chi(x_{i_2}) = \chi(v_1)) \vee (\chi(x_{i_3}) = \chi(v_1))$$

We now let X' in the instance of *Opt*(Aggravated k-Colorability) denote the set of variable vertices built from U'. Recall that to one variable in U' corresponds one variable vertex. Therefore, X' does not contain any of the variables from

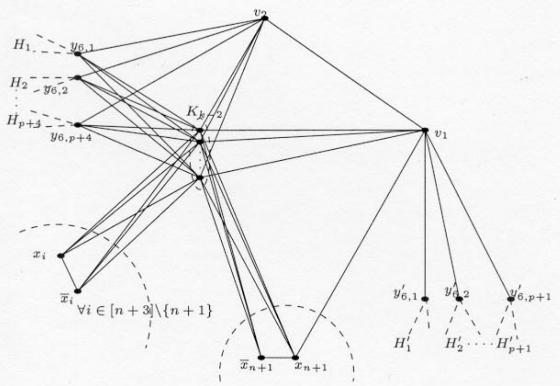

Fig. 2. The subgraph W of G.

the set $\{x_i, \overline{x}_i\}_{i \in \{p+1,p+2,p+3,p+4\}}$. We fix the special vertex to v_1. Proof of the reduction of theorem 3 can be found in [14]. □

We now prove theorem 1 using the following reduction. For any graph G instance of *Opt*(Aggravated k-Colorability), we create a set of examples $\mathcal{E}$ described over a set of predicates $a_i(.), i \in [n]$, and constant symbols $\{l_i\}, i \in [n]$; $\{m_{i,j}\}, (i,j) \in E$; $\{s_{i,x}\}, i \in X'$ (there are $n + |E| + |X'|$ constant symbols). We let x denote the special vertex of the instance of *Opt*(Aggravated k-Colorability).

- $S^+ = \{q(l_i) \leftarrow \wedge_{j \neq i} a_j(l_i) : 1 \leq i \leq n\}$. The weight of these positive examples is $w^+ = n$.
- $S_1^- = \{q(m_{i,j}) \leftarrow \wedge_{k \notin \{i,j\}} a_k(m_{i,j}) : (i,j) \in E\}$. The weight of these negative examples is $w^- = w^+ = n$.
- $S_2^- = \{q(s_{i,x}) \leftarrow \wedge_{k \notin \{i,x\}} a_k(s_{i,j}) : (i,x) \notin E \wedge i \in X'\}$. The weight of these negative examples is $w'^- = 1$.

The proof of theorem 1 follows from the proof of the two following propositions:

Proposition 1 *From any k-function-free Horn clauses making t errors on $S^+ \cup S_1^- \cup S_2^-$, we can build in polynomial time a feasible solution to Opt(Aggravated k-Colorability) which gives the same color as x to at most t elements of X'.*

Proof. Suppose that $t \geq n$. In that case, we can use the trivial coloring of the instance of *Opt*(Aggravated k-Colorability). Since $X' \subset X$, there are trivially at most $t \geq n$ elements of X' colored by the same color as x.
Suppose now that $t < n$. In that case, any element of weight n is well classified. Let $\{h_1, h_2, ..., h_k\}$ denote the set of clauses solution to *Opt*(Weighted-Approx(k-function-free Horn clauses)). We can suppose without loss of generality that any

predicate is absent of at most one clause. Otherwise, we can add this predicate to all clauses but one (in which it is absent), and this does not increase the number of errors since it only forces all positive examples to θ-subsume exactly one clause among the k. The colorability assigned is the following one : $\forall i \in [n], \forall j \in [k]$, if $a_i(.) \not\subset h_j$, then give color j to vertex x_i.
This is a valid k-colorability, otherwise some examples of weight n would be misclassified. The t errors are made on examples of weight 1. These examples represent distinct vertices having the same color as x. Note that a single clause is responsible for all the errors : $h_{\chi(x)}$.

Proposition 2 *Any feasible solution to Opt(Aggravated k-Colorability) which gives the same color as x to at most t elements of X' can be transformed in polynomial time into a feasible solution to Opt(Weighted-Approx(k-function-free Horn clauses)) which makes at most t errors over the examples.*

Proof. The k clauses of the solution to *Opt*(Weighted-Approx(k-function-free Horn clauses)) are defined by:

$$\forall j \in [k], h_j \equiv q(X) \leftarrow \wedge_{i \in [n]:\chi(x_i) \neq j} a_i(X)$$

These clauses do not make errors on examples of weight n. The only errors made are on examples of weight 1 corresponding to vertices having the same color as x. Note that in our construction, only one clause makes all the errors : $h_{\chi(x)}$.

Theorem 1 now follows from propositions 1 and 2.

4 Result on function-free Horn clauses having polylogarithmic size

Let QP denote the class of problems admitting quasi-polynomial time deterministic algorithms. A function of n is quasi-polynomial iff $f(n) \leq n^{\log^c n}$ for some constant c. Many results have introduced the class QP, such as for example [1], to point out the fact that hard-to-solve or approximate problems (such as *Opt*(Aggravated 3-SAT), and therefore *Opt*(Weighted-Approx(k-function-free Horn clauses))) might not even admit quasi-polynomial time approximation algorithms. We are going to use this fact to prove our next result. We now prove theorem 4 below.

Theorem 4 *If $NP \not\subset QP$, $\forall d > 0$ a constant, Opt(Weighted-Approx($\log^{d+2} n$-function-free Horn clauses)) is not approximable to within $n^{\log^d n}$.*

In order to do this, we highlight a correlation between the size of the formula and its error rate. To that effect, we multiply $\log^{d+2} n = K$ instances of *Opt*(Weighted-Approx(k-function-free Horn clauses)) by concatenating the tail of the examples to form new ones, thus described over a set of $n \times K$ predicates (plus the inferred predicate $q(.)$). Each predicate taken from the initial set serves to create exactly K new predicates. This can be viewed as making set products among the set of tails of the examples of the instance of Opt(Weighted-Approx(k-function-free Horn clauses)) we used to prove theorem 1. One of these examples could be represented *e.g.* as follows:

$$q(l'_i) \leftarrow \left(\wedge_{j=1;j\neq i}^{j=n} a_{j,1}(l'_i)\right) \wedge \left(\wedge_{j=1;j\neq i}^{j=n} a_{j,2}(l'_i)\right) \wedge ... \wedge \left(\wedge_{j=1;j\neq i}^{j=n} a_{j,K}(l'_i)\right)$$

The subscript in $a_{i,j}(.)$ denotes the j^{th} copy of the initial predicate $a_i(.)$. The example represented has each of its K predicate parts coming from the copy of an initial example from S^+ (we shall write that this example comes from $(S^+)^K$). The new examples are described as follows :

- S^+_K is the subset $(S^+)^K$. Their weight is n^K.
- $S^-_{1,K} = (S^+ \cup S^-_1)^K \backslash S^+_K$. their weight is n^K.
- $S^-_{2,K} = (S^-_2)^K$. their weight is 1.

Due to the increase in the number of examples, the initial set of $n + |E| + |X'|$ constant symbols is replaced by a new larger set of size $(n + |E|)^K + |X'|^K$. Note that this new set of examples is created in quasi-polynomial time. For the sake of brevity, we let c_* denote the minimal error of a feasible solution to Opt(Weighted-Approx(k-function-free Horn clauses)), and c'_* the minimal error of a feasible solution to Opt(Weighted-Approx(K-function-free Horn clauses)).

Proposition 3 $c_* < n$

Proof follows from the fact that the instance of Opt(Aggravated k-Colorability) which serves to build the instance of Opt(Weighted-Approx(k-function-free Horn clauses)) is always k-colorable. Proposition 3 comes from the construction technique of proposition 2 : any example of weight n is well classified, and there are at most $n-1$ examples of weight 1. According to propositions 1 and 2, we can suppose that each predicate is absent from at most one clause. Let $c_*(c_{\chi(x)})$ denote the minimal error of the clause that does not contain the predicate $a_x(.)$ corresponding to the special vertex of the instance of Opt(Aggravated k-Colorability). From propositions 1 and 2, proposition 3 can be refined :

Proposition 4 $c_* = c_*(h_{\chi(x)})$

Proposition 5 $c'_* \geq (c_*)^K$.

Proof. Whenever an example from the set $S^+_K \cup S^-_{1,K}$ is badly classified, proposition 3 gives the result: $c'_* \geq n^K > (c_*)^K$. Suppose that all examples from the set $S^+_K \cup S^-_{1,K}$ are well classified. Any error is necessarily due to an example of the set $S^-_{2,K}$. The only type of clause that can cause these errors is a clause of type $q(X) \leftarrow P_1 \wedge P_2 ... \wedge P_K$ where $\forall j \in [K]$, P_j is a clause described over the set of predicates $\{a_{i,j}\}_{i\in[n]}$ such that P_j makes errors on S^-_2. Since the error of each P_j is at least $c_*(h_{\chi(x)})$, which is c_* (proposition 4), the error of the conjuction ($\wedge$) is at least the product of the minimal error of each part, $(c_*)^K$. The overall error of the set of clauses is thus at least equal to this quantity.

Proposition 6 *Let h denote a feasible solution to Opt(Weighted-Approx(K-function-free Horn clauses)), whose error is a. Then we can find a feasible solution to Opt(Weighted-Approx(k-function-free Horn clauses)) whose error is no more than $a^{\frac{1}{K}}$.*

Proof. Recall that the graph we constructed from the instance of the problem *Opt*(Aggravated 3-SAT) is always k colorable. Therefore, there always exist a set of k-function-free Horn clauses consistent with $S^+ \cup S_1^-$ (proposition 2), having error $< n$. In h, whenever an example from the set $S_K^+ \cup S_{1,K}^-$ is badly classified, proposition 3 gives the result since the error of h is at least $c'_* \geq n^K$. Suppose that all examples from the set $S_K^+ \cup S_{1,K}^-$ are well classified. Any error is necessarily due to an example of the set $S_{2,K}^-$. The only type of clause that can cause these errors is a clause of type $q(X) \leftarrow P_1 \wedge P_2 ... \wedge P_K$ where $\forall j \in [K]$,

1. P_j is a subset of predicates described over the set of predicates $\{a_{i,j}\}_{i\in[n]}$, and
2. the clause isomorphic to $q(X) \leftarrow P'_j$ described over the set $\{a_i\}_{i\in[n]}$, obtained by replacing each $a_{l,j} \in P_j$ by a_l in P'_j, makes errors on S_2^-.

Note that the error of the conjuction ($\wedge$) is the product of errors of each part $P_j, j \in [K]$ on S_2^-. So, the part P_* over $P_1, ..., P_K$ leading to the least number of errors on S_2^- makes an error that is at most $a^{\frac{1}{K}}$. Now, construct the set h' of $(k-1)$-function-free Horn clauses of proposition 2 with all clauses except $h_{\chi(x)}$, and add (for $h_{\chi(x)}$) the clause corresponding to the part P_* (it is $q(X) \leftarrow P'_*$ as described in point 2 above). The overall error of h' does not exceed $a^{\frac{1}{K}}$.

Proposition 7 $c'_* \leq (c_*)^K$

The proof of this proposition follows simply if we calculate the K-time cross-product of the solution realizing the cost c_*. We obtain a set of k^K clauses, and the construction can be realized in quasi-polynomial time.
From this, it comes that $c'_* = (c_*)^K$. We now prove theorem 4 *ad absurdum*. Suppose that *Opt*(Weighted-Approx($\log^{d+2} n$-function-free Horn clauses)) is approximable to within $n^{\log^d n}$. In quasi-polynomial time, from any instance of *Opt*(Weighted-Approx(k-function-free Horn clauses)), we build an instance of *Opt*(Weighted-Approx(K-function-free Horn clauses)) following the procedure described at the beginning of this section. We can find an element of K-function-free Horn clauses whose error does not exceed $(Kn)^{\log^d(Kn)} c'_*$ (hypothesis). Thus, we can find a solution to *Opt*(Weighted-Approx(k-function-free Horn clauses)) whose cost is approximately no more than $\left((Kn)^{\log^d(Kn)} c'_*\right)^{\frac{1}{K}}$ (proposition 6). But (using propositions 5 and 7)

$$\left((Kn)^{\log^d(Kn)} c'_*\right)^{\frac{1}{K}} = \left((Kn)^{\log^d(Kn)} c'_*\right)^{\frac{1}{\log^{d+2} n}} \leq (Kn)^{\frac{1}{\log(Kn)}} c_* = \mathcal{O}(c_*)$$

This contradicts theorem 1, since we obtain an approximation of *Opt*(Weighted-Approx(k-function-free Horn clauses)) to within a constant factor.

5 Consequences on learnability

In this paper, we have essentially presented two structural complexity results. Their purpose, is to prove that Horn Clauses display very severe error rates when used for learning in complex domains, and they extend our previous results [9]. It should be noted that as in this previous work, in order to obtain general properties, we have studied general function-free Horn-clauses but the proofs of our theorems are made in such a manner (using simple ILP formalisms) as to remain valid for the more specialized classes encoutered in ILP. So, both our non-approximability results also apply to many subsets that have led to theoretical studies in the PAC-learning model. Classes for which this result applies are subclasses of the following classes where the number of clauses is limited to the values of theorems 1 and 4 : ij-determinate non recursive Horn clauses [13] where i and j are any integer constants satisfying $i \geq 0$ and $j > 0$, and l-local Horn clauses [3] where l is any integer constant satisfying $l > 0$.

References

1. S. Arora. Probabilistic checking of proofs and hardness of approximation problems. Technical Report CS-TR-476-94, Princeton University, 1994.
2. W. Buntine. Generalized subsumption and its applications to induction and redundancy. *Artificial Intelligence*, 36:149–176, 1988.
3. W.W. Cohen. Pac-learning nondeterminate clauses. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI'94*, pages 676–681, 1994.
4. W.W. Cohen. Pac-learning recursive logic programs: Efficient algorithms. *Journal of Artificial Intelligence Research*, 2:501–539, 1995.
5. W.W. Cohen. Pac-learning recursive logic programs: Negative results. *Journal of Artificial Intelligence Research*, 2:541–571, 1995.
6. S. Dzerovski, S.H. Muggleton, and S. Russel. Pac-learnability of determinate logic programs. In *Proceedings of COLT-92*, pages 128–137, 1992.
7. E.M. Gold. Language indentification in the limit. *Information and Control*, 10:447–474, 1967.
8. K-U. Höffgen and H.U. Simon. Robust trainability of single neurons. In *Proc. of the 5 th International Conference on Computational Theory*, 1992.
9. P. Jappy, R. Nock, and O. Gascuel. Negative robust learning results for horn clause programs. In *Proceedings of ICML'96*, pages 258–265, 1996.
10. M. Kearns, M. Li, L. Pitt, and L.G. Valiant. On the learnability of boolean formulae. In *Proceedings of STOCS'87*, pages 285–294, 1987.
11. J.U. Kietz and S. Dzeroski. Inductive logic programming and learnability. *Sigart Bulletin*, 5:22–32, 1994.
12. S.H. Muggleton. Bayesian inductive logic programming. In *Proceedings of the Seventh Workshop on COmputational Learning Theory*, 1994.
13. S.H. Muggleton and C. Feng. *Efficient induction of logic programs*. Inductive Logic Programming. Academic Press, New York, 1992.
14. R. Nock and P. Jappy. On the hardness of approximating function-free horn clauses. Technical Report LIRMM-RR-98017, LIRMM, 1998.
15. L.G. Valiant. A theory of the learnable. *Association for Computing Machinery Communications*, 27:1134–1142, 1984.

A.3 [Nock, 1999]

L'article présenté ici est une version étendue de [Nock, 1999], en cours de soumission.

Complexity in the case against accuracy estimation

Richard Nock [1]

Université des Antilles-Guyane,
GRIMAAG-Département Scientifique Interfacultaire,
Campus de Schoelcher, BP 7209,
97275 Schoelcher, France

Abstract

Some authors have repeatedly pointed out that the use of the accuracy, in particular for comparing classifiers, is not adequate. The main argument concerns some assumptions of seldom validity or correctness underlying the use of this criterion. In this paper, we study the computational burden of the accuracy's replacement for building classifiers in various ways, using the framework of Inductive Logic Programming. Replacement is investigated in three ways: completion of the accuracy with an additional requirement, replacement of the accuracy with a bi-criterion recently introduced from statistical decision theory: the ROC analysis, and replacement of the accuracy by a single criterion. We prove very hard results for most of the possible replacements. A first result shows that allowing the arbitrary multiplication of clauses appears to be totally useless. "Arbitrary" is to be taken in its broadest meaning, in particular exponential. The second point is the sudden appearance of the negative result, which is not a function of the requirements demand. The third point is the equivalence in difficulty of criteria much different from each other. In contrast, the accuracy criterion appears to be tractable in this framework.

[1] Email: `rnock@martinique.univ-ag.fr`. Fax: (+596) 72 73 62. Phone: (+596) 72 73 64. URL: `http://www.martinique.univ-ag.fr/~rnock`.

Preprint submitted to Elsevier Preprint *1 February 2002*

1 Introduction

One of the primary tasks of Machine Learning (ML) and Data Mining (DM) systems is related to classification. This basically consists in giving the most accurate answer to the affectation of some observations (or patterns) to a finite number of classes. Historically, measuring the quality of the system's answer has mostly been a matter of computing its accuracy, that is, the frequency (or probability) of correct predictions made [Domingos, 1999], [Kubat et al., 1998]. The advent of new technological media authorizing the storing of databases of huge sizes, together with the increasing diversity of the problems (and goals) addressed, have favored the emergence of a research trend in ML and DM. This trend discusses, mostly experimentally, this standardized use of the accuracy to assess the merit of a system (see *e.g.* [Kubat et al., 1998]), as well as to compare different classifier learning algorithms to decide which one should be used preferentially (see *e.g.* [Provost et al., 1998]). Following this trend, we question again the replacement of the accuracy itself by other performance measures. However, apart from our purely theoretical standpoint on this question, we think our approach is original and distinguished from the others in that it is, to our knowledge, the first to evaluate the computational burden of the accuracy's replacement.

The primary inadequacy of the accuracy stems from a tacit assumption that the overall accuracy controls by-class accuracies, or similarly that class distributions among examples are constant and relatively balanced, see for example [Provost and Fawcett, 1997]. This is obviously not true : skewed distributions are frequent in agronomy, or more generally in life or earth sciences. As an example, consider the human DNA, in which no more than 6% are coding genes [Provost et al., 1998]. Another example is the oil spill detection problem of [Kubat et al., 1998], in which roughly 4% of the data represent oil slicks, the remaining being lookalikes. Even more extreme cases exist, in information retrieval, in which the minority class can scarcely represent 0.2% of the data [Kubat et al., 1998]. In all these cases, the interesting, unusual class is often the rare one, and the well-balanced hypothesis may simply lead to the elusion of its elements when building a classifier. In [Kubat et al., 1998], a simple classifier labeling all patterns as lookalikes (this is the so-called majority rule) would achieve an accuracy of 96%. As pointed out by [Kubat et al., 1998], this looks like a high accuracy, but the classifier is totally useless since it completely fails to achieve the goal of oil spill detection. On the other hand, a system achieving only 94% detection on oil spills, and 94% detection on lookalikes, would have a worse accuracy, and yet would be deemed highly successful [Kubat et al., 1998].
This last example shows that, in real-world problems, not only is the balanced distributions assumption false, but also of heavy consequences may be the misclassification of some examples, a cost which is not integrated in

the accuracy. Fraud detection is another good example of such a situation [Provost et al., 1998], but there are many others. In database marketing, a prominent application consists in targeting the people likely to respond to a mailing. In that case, the cost of mailing to a non-respondent is small, but the cost of not mailing to someone who would respond is the entire profit lost [Domingos, 1999]. Solving the cost problem by the integration of the costs in the accuracy, to shift its behavior towards the crucial examples, is also far from being obvious, as it involves "multiple considerations whose units are incommensurable" [Kubat et al., 1998].
Furthermore, the accuracy may be inadequate in some cases because other parameters are to be taken into account. Some works [Kukar et al., 1997] report the need to add an information measure to the accuracy, to eliminate the influence of prior probabilities. Constraints on size parameters (see [Nock and Gascuel, 1995], [Nock and Jappy, 1998]) are sometimes to be used because we want to obtain small formulas, to ease their interpretability by the system user.
Finally, some works also report the experimental convenience that reducing the size of the data itself can have when simply optimizing the accuracy [Sebban and Nock, 2002]. Indeed, it is well known in ML and DM that removing some parts of the data, such as features (or variables), is a good experimental solution to reduce the size of the models built afterwards, while avoiding to damage too much their accuracy. Sometimes, it can even provide a way to improve their accuracy, for hard problems. Whereas experiments show that reducing features can be a good criterion to optimize when dealing with the accuracy's optimization, one may wonder the way these two constraints computationally interact.

To examine the possible influence of all these replacement criteria, we have chosen as our framework a field particularly sensitive to the computational complexity factor, Inductive Logic Programming (ILP). ILP is a rapidly growing research field, concerned with the use of variously restricted subclasses of Horn clauses to build ML algorithms. According to a census of [Srinivasan, 1998], in 1998, almost seventy applications were using ILP formalism, twenty of which were science applications, which can be partitioned into biological (four) and drug design (sixteen) applications. With the increasing popularity of ILP, in particular to address complex domains, this number has certainly increased since. ILP-ML algorithms have been applied with some success in areas of biochemistry and molecular biology [Srinivasan, 1998]. Using ILP formalism, we argue that the replacement of the accuracy raises computational complexity issues. This is all the more important in ILP studies, as ILP is a field which can be naturally concerned with intractability or even undecidability issues [Kietz, 1993], and keeping tractability is of primary importance to keep the ILP formalism power affordable to practical learning systems. For this reason, ILP is certainly a domain of choice for a computational study of the accuracy's replacement. More precisely, here is the structuring of the argument.

First, to address the replacement problem, we explain that the single accuracy requirement can be completed by an additional requirement to provide more adequate criteria. We integrate various constraints over two important kind of parameters: by-class error functions, and representation parameters such as feature selection ratios, size constraints. These criteria are inspired by the works of [Kubat et al., 1998], [Nock and Gascuel, 1995], [Nock and Jappy, 1998], [Sebban and Nock, 2002]. We do not integrate in our criteria the information measure of [Kukar et al., 1997], as it is mainly designed to handle classifiers with probabilistic answers, and is therefore not suited to ordinary Horn clauses.
Then, we study the replacement of the accuracy criterion using a general method derived from statistical decision theory, based on a specific bi-criteria optimization (see *e.g.* [Provost and Fawcett, 1997], [Provost and Fawcett, 1998], [Provost et al., 1998]).
Finally, we investigate the replacement of the error by a single replacement criterion. Two candidates we study are criteria proposed in [Provost and Fawcett, 1997], and used in [Kubat et al., 1998].
In this paper, we show that any of such integration leads to a very negative structural complexity result, which is not faced by the accuracy optimization alone. The result has a side effect which can be presented as a "loss" in the formalism's expressiveness, a seldom property in classical ML complexity issues. Indeed, it authorizes the construction of arbitrary large (even exponential-sized) sets of Horn clauses, but which we prove having no more expressive power than a single Horn clause. We prove a threshold in intractability since it appears immediately with the additional requirement, and is not a function of the tightness of it. Furthermore, the effects of the constraints on optimal accuracies vanish as the number of predicates increases, as optimal accuracies with or without the additional constraints are asymptotically equal. This phenomenon tends to strengthen the threshold effect in intractability. Finally, for some criteria, their blending with the accuracy brings the most negative result: not only does the intractability appears immediately with the criterion, but also the error cannot be dropped down under that of the unbiased coin.

The reductions are presented for a subclass of Horn formalism simple enough to be an element of the intersection of all classically encountered in theoretical ILP studies or practical ILP learning systems. As a consequence, our results also hold in all these other settings. The following section details the bases of learnability and ILP. It is followed by a section introducing the possible criteria to address the accuracy's drawbacks, and the tools used in our proofs. Then, all the results are presented in the last section, along with some possible extensions to other formalisms, or to learning models. For the reader's convenience, the proofs and technical aspects, not necessary to understand the results, have been bulked in two appendices. The first one presents a synthetic view of all proofs, the second one presents in-depth reductions. In order not to laden the paper with a collection of extensive proofs, some cases have been voluntarily omitted.

2 Learnability and ILP

Denote as $\mathcal{C}$ and $\mathcal{H}$ two classes of concepts representations, respectively called *target* class and *hypothesis* concept class. Informally, our objective is to build a concept from the hypothesis class, approximating as best as possible an unknown concept c, called the target concept, element of $\mathcal{C}$. In real-world domains, we do not know the target concept's class, that is why we have to make *ad hoc* choices for $\mathcal{H}$ with a powerful enough formalism, yet ensuring tractability. Even if some benchmarks problems appear to be easily solvable [Holte, 1993], ML applications, and particularly ILP, face more difficult problems [Srinivasan, 1998], for which the choice of $\mathcal{H}$ is crucial.
After the choice of $\mathcal{H}$, approximating the target concept can only be achieved by catching a glimpse of it, through its extensional representation, *i.e.* by drawing examples, classified according to c. Generally, the data collected can only account for a small part of this very large set, and the objective is then to build the intensional representation of some hypothesis, whose extensional representation shall hopefully match as best as possible the target concept's. Most of the studies dealing with the accuracy's replacement, as well as computational complexity results in ML, have been investigated with two classes [Provost et al., 1998], [Provost and Fawcett, 1998]. We also consider a two-classes setting. It is not really important for us, as results already become hard in that setting.
We shall see later in this section how examples and concepts are described in the context of ILP. Before, it is important to clarify the way we "collect" the examples, and its use to obtain either positive, or negative results, in ML or DM. Theoretically speaking, a large part of the modern approaches to obtain positive results for ML/DM algorithms draw their roots in two fundamental bodies, the so-called Probably Approximately Correct (PAC) learning model of [Valiant, 1984], and the Statistical Learning Theory, fathered by [Vapnik, 1998]. The principle is that the examples are drawn from some unknown, but fixed distribution D, and labeled according to an unknown $c \in \mathcal{C}$. If we suppose that the representation space is discretized, then we can denote the accuracy of some $h \in \mathcal{H}$ with respect to (w.r.t.) c by $P_D(h = c) = \sum_{h(x)=c(x)} D(x)$ (here, x is an observation, and $h(x)$, $c(x)$ are respectively the classes given to x by h and c). Note that this quantity is measured over the whole set of possible examples, a domain to which we do not

have access to, as specified before. We have only access to its estimator over the sample collected. This raises statistical issues to evaluate the quality of this estimator (and h), issues discussed in many papers (see *e.g* [Vapnik, 1998]). The objective of this paper is not to discuss the statistical burden of the theory, but its computational issues.
Proving most negative, complexity-theoretic results for ML/DM, follows a quite standardized approach. It consists in building a particular set of examples, supposed to be the set collected, and giving a frequency distribution over these examples "mimicking" D (a seminal paper to this approach is [Kearns et al., 1987]), and then proving the result on the basis of this particular instance of the problem. Our results also rely on this scheme. It is important to note that the negative results are, in that case, complexity-theoretic, *i.e.* they raise the hardness of finding efficient (*e.g.* polynomial) algorithms to address the problem. As briefly exposed before, they do not address the statistical hardness of building h, since the instance built boils down to having access to the whole domain knowledge (all examples that are not present in the set are supposed to have zero probability of occurrence).
An interesting fact in negative computational results in ML/DM is that they may have two consequences. The first is what motivates this paper, *i.e.* the inexistence of affordable practical algorithms to solve these problems. The second is the extension of these results to negative results for learning in models derived from the PAC model of [Valiant, 1984]. Some of our results can be extended to negative results on the PAC-derived robust learning model of [Höffgen and Simon, 1992], [Jappy et al., 1996]. This is described later.

We now introduce our formalism for the examples and the hypotheses, ILP. The field of ILP is concerned with the induction of first-order Horn clauses from examples and background knowledge. A Horn clause has the following form:

$$q(...) \leftarrow a_1(...) \wedge a_2(...) \wedge ... \wedge a_n(...)$$

$q(...)$ is called the head of the clause and the conjunction $a_1(...) \wedge a_2(...) \wedge ... \wedge a_n(...)$ is called the body of the clause. A Horn clause with no body is unit. A clause with no variable is ground. Given a Horn clause language $\mathcal{L}$ and a correct inference relation on $\mathcal{L}$, the problem can be formalized in a general way as follows ([Jappy et al., 1996]):

Definition 1 *Given:*

- *A background knowledge* $\mathcal{BK}$ *expressed in a language* $\mathcal{LB} \subseteq \mathcal{L}$,
- *A set of examples* $\mathcal{S}$ *in a language* $\mathcal{LS} \subseteq \mathcal{L}$, *consisting of positive examples,* $\mathcal{S}^+$, *and negative examples,* $\mathcal{S}^-$, *such that* $\mathcal{B} \not\models \mathcal{S}^+$ *(*$\mathcal{B}$ *does not entail the positive examples) and* $\mathcal{B}, \mathcal{S} \not\models \Box$ *(*$\mathcal{S}$ *is consistent with* $\mathcal{B}$*).*

6

- *A hypothesis class $\mathcal{H}$ described over a language $\mathcal{LH} \subseteq \mathcal{L}$,*

find a hypothesis $h \in \mathcal{H}$ such that

$$\mathcal{B} \wedge h \models \mathcal{S}^+ \quad (1)$$
$$\mathcal{B} \wedge h \not\models \mathcal{S}^- \quad (2)$$

i.e., $\mathcal{B}$ and h explain the positive examples whereas they do not explain the negative examples.

We now give some precisions on this definition. The background knowledge in ILP is usually restricted in order to avoid undecidability problems about the deduction process [Kietz, 1993], [Dzeroski et al., 1992]. A usual restriction makes use of ground background knowledge, that is, consisting of ground unit clauses. A clause is ground if it does not contain any variables. Therefore, to ensure tractability, we suppose that the background knowledge consists of ground unit clauses, and examples are ground unit clauses too. Another restriction commonly encountered consists in preventing the use of function symbols of arity > 0:

Definition 2 *A clause is called function-free iff all its arguments are either variables or constants (function symbols of arity 0).*

As in [Kietz and Dzeroski, 1994], we use θ-subsumption as the inference relation. θ-subsumption is a correct and complete inference procedure between function-free Horn clauses ($h \models h'$ iff $h\theta \subseteq h'$). This leads however to a modification of the learning problem, as stated by the following lemma:

Lemma 3 *[Kietz, 1993] The learning problem is equivalent to learning the same program with θ-subsumption, and empty background knowledge and examples defined as ground Horn clauses of the form $e \leftarrow b$, where $e \in \mathcal{S}$ and $b \in \mathcal{BK}$.*

This lemma allows us to incorporate the background knowledge in the new examples (and is thus empty). Our result make use of a simple subclass of Horn clause formalism. Its property is that it is an element of common subclasses of Horn clause formalisms usually encountered in practice or in theoretical learning studies. Therefore, since our results are essentially negative, they hold also for all these other subclasses. The most important property of our subclass is that the predicates arity is one. Therefore,

- we can suppose that the Horn clauses contain the same variable, say X. In other words, the clauses are constrained.
- the clauses are 01-determinate as defined in [Kietz, 1993], [Dzeroski et al., 1992]. In other words, the maximum predicate arity is 1 and the depth of each term

is that of the head, 0. This represents the easiest case of determinacy.
- $\mathcal{BK}$ does not contain the predicate to be inferred, and the Horn clauses are non recursive.

The principle of our negative results, from an ILP point of view, is quite simple: we create a formalism so simple such that, given the constraints, there cannot always exist a set of Horn clauses solution of the learning problem. From that, the goal of the learning problem is relaxed to that of an approximation problem well known in robust learning [Jappy et al., 1996]: find a hypothesis $h \in \mathcal{H}$ such that

$$\mathcal{B} \wedge h \models \mathcal{S}^+ \quad (3)$$
$$\mathcal{B} \wedge h \not\models \mathcal{S}^- \quad (4)$$

for the largest part of the examples in $\mathcal{S}$.

3 Replacement criteria and the hardness technique

3.1 Extending the accuracy

For any fixed positive rational ν, we use the following adequate notion of distance [Boucheron, 1992] between two reals u, v: $d_\nu(u,v) = \frac{|u-v|}{u+v+\nu}$. We also use eight rates on the examples (definitions differ slightly from [Provost et al., 1998]):

$$TP = \sum_{h(x)=1=c(x)} D(x) \quad ; \quad TPR = \frac{TP}{\sum_{c(x)=1} D(x)}$$

$$FP = \sum_{h(x)=1\neq c(x)} D(x) \quad ; \quad FPR = \frac{FP}{\sum_{c(x)=0} D(x)}$$

$$TN = \sum_{h(x)=0=c(x)} D(x) \quad ; \quad TNR = \frac{TN}{\sum_{c(x)=0} D(x)}$$

$$FN = \sum_{h(x)=0\neq c(x)} D(x) \quad ; \quad FNR = \frac{FN}{\sum_{c(x)=1} D(x)}$$

In many DM/ML domains, the user's desiderata are often the optimization of more than one basic criterion (accuracy, precision, recall, sizes, etc...). Various composed criteria exist, combining some of these, but it is hard to obtain a suitable combination into one criterion, so as to optimize in one step more than one of these basic demands. The accuracy is typical, but others are well known, such as the geometrical mean of [Kubat et al., 1998], which ignores precision. Some authors, such as [Kukar et al., 1997], have proposed to take into account more than one criterion, such as information measures for probabilistic classifiers. In order to complete the accuracy requirements, we imagine seven types of additional constraints aiming at controlling the well-balanced drawback of the accuracy alone [Provost and Fawcett, 1997], or precision or recall measures [Provost and Fawcett, 1997,Kubat et al., 1998], or size parameters [Nock and Gascuel, 1995], [Sebban and Nock, 2002]. Each of them is parameterized by a number ζ (between 0 and 1), and defines a subset of $\mathcal{H}$, which shall be parameterized by D if the distribution controls the subset through the constraint. The first three subsets of $\mathcal{H}$ contain hypotheses for which the FP and FN are not far from each other, or a one-side error is upper bounded:

$$\mathcal{H}_{D,1}(\zeta) = \{h \in H | d_\nu (FP, FN) \leq \zeta\} \quad (5)$$
$$\mathcal{H}_{D,2}(\zeta) = \{h \in H | FN \leq \zeta\} \quad (6)$$
$$\mathcal{H}_{D,3}(\zeta) = \left\{h \in H | FN \leq \frac{1}{\zeta} FP\right\} \quad (7)$$

The two following subsets are parameterized by constraints equivalent to some frequently encountered in the information retrieval community [Slattery and Craven, 1998], respectively (1 minus) the precision and (1 minus) the recall criteria.

$$\mathcal{H}_{D,4}(\zeta) = \left\{h \in H | \frac{FP}{TP + FP} \leq \zeta\right\} \quad (8)$$
$$\mathcal{H}_{D,5}(\zeta) = \left\{h \in H | \frac{FN}{TP + FN} \leq \zeta\right\} \quad (9)$$

Now, we give two more constraints specific to Horn clauses. Horn clauses shall be extensively defined in a section devoted to ILP formalism. We give some preliminary and necessary definitions for the two constraints we define. A Horn clause (a definite program clause) [Dzeroski et al., 1992] has the following form:

$$q(...) \leftarrow a_1(...) \wedge a_2(...) \wedge ... \wedge a_n(...)$$

Here, q, a_1, a_2, ..., a_n are predicate symbols. Define #`Predicates`(h) as the total number of different predicates of h, #`Whole_predicates`(h) as the overall number of predicates of h (if one predicate is present k times, it is counted k times), and #`Total_predicates` as the total number of different available predicates to build a Horn clause for our specific problem. The two last subsets of $\mathcal{H}$ are parameterized by formulas respectively having a sufficiently small fraction of the available predicates, or having a sufficiently small overall size:

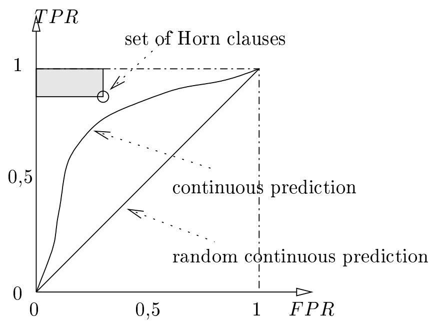

Fig. 1. The ROC analysis of a learning algorithm, with three general classifications: discrete, continuous, and default.

$$\mathcal{H}_6(\zeta) = \left\{ h \in H | \frac{\#\texttt{Predicates}(h)}{\#\texttt{Total\_predicates}} \leq \zeta \right\} \quad (10)$$

$$\mathcal{H}_7(\zeta) = \left\{ h \in H | \frac{\#\texttt{Whole\_predicates}(h)}{\#\texttt{Total\_predicates}} \leq \zeta \right\} \quad (11)$$

The division by the total number of different predicates in $\mathcal{H}_7(\zeta)$ is made only for technical reasons: to obtain hardness results for small values of ζ. The first problem we address can be summarized as follows:
Problem 1: **Given ζ and $i \in \{1, 2, ..., 7\}$, can we find an algorithm returning a set of Horn clauses from $\mathcal{H}_{(D,)i}(\zeta)$ whose error is no more than a given γ, if such a hypothesis exists ?**

3.2 Replacing the accuracy: the ROC analysis

Receiver Operating Characteristic (ROC) analysis is a traditional methodology from signal detection theory [Egan, 1975]. It has been used in machine learning recently [Provost and Fawcett, 1998], [Provost and Fawcett, 1997], [Provost et al., 1998] in order to correct the main drawbacks of the accuracy. In ROC space (this is the coordinate system), we visualize the performance of a classifier by plotting TPR on the Y axis, and FPR on the X axis. Figure 1 presents the ROC analysis, along with three possible outputs which we present and analyze now. If a classifier produces a continuous output (such as an estimate of posterior probability of an instance's class membership [Provost et al., 1998], or a real-valued confidence such as in AdaBoost [Schapire and Singer, 1998]), for any possible value of FPR, we can get a value for TPR, by thresholding the output between its extreme bounds. If a classifier produces a discrete output (such as Horn clauses), then the classifier gives rise to a single point. If the classifier is the random choice of the class, either (if it is continuous) the curve is the line $y = x$, or (if it is discrete) there is a single dot, on the line

10

$y = x$. One important thing to note is that the ROC representation gives the behavior of an algorithm without regarding the class distribution or the error cost [Provost and Fawcett, 1997]. And it allows to choose the best of some classifiers, by the following procedure. Fix as K^+ the cost of misclassifying a positive example, and K^- the cost of misclassifying a negative example (these two costs depend on the problem). Then the *expected* cost of some classifier represented by point (FPR, TPR) is given by the following formula:

$$\sum_{c(x)=1} D(x) \times (1 - TPR) \times K^+ + \sum_{c(x)=0} D(x) \times FPR \times K^- \tag{12}$$

Two algorithms, whose corresponding point are respectively (FPR_1, TPR_1) and (FPR_2, TPR_2), have the same expected cost iff

$$\frac{TPR_2 - TPR_1}{FPR_2 - FPR_1} = \frac{\sum_{c(x)=1} D(x)K^+}{\sum_{c(x)=0} D(x)K^-} \tag{13}$$

This gives the slope of an *isoperformance* line, which only depends on the relative weights of the examples, and the respective misclassification costs. Given one point on the ROC, the classifiers performing better are those on the "northwest" of the isoperformance line with the preceding slope, and to which the point belongs. If we want to find an algorithm A performing *surely* better than an algorithm B, we therefore should strive to find A such that its point lies into the rectangle whose opposite vertices are the (0,1) point (the perfect classification) and B's point (a grey rectangle is shown on the top left of figure 1). From that, the second problem we address is the following:
<u>Problem 2</u>: **Given one point (TPR_x, FPR_x) on the ROC, can we find an algorithm returning a set of Horn clauses whose point falls into the rectangle with opposite vertices (0,1) and (TPR_x, FPR_x), if such a hypothesis exists** ?
Note that the problem we address is based on weak constraints: indeed, we only require the algorithm to work on a *single* point (TPR_x, FPR_x).

3.3 Replacing the accuracy by a single criterion

The ROC analysis is based on two criteria, controlling FPR and TPR. The question of whether the accuracy can be replaced by a single criterion instead of two has been raised in [Provost and Fawcett, 1997]. Some researchers [Provost and Fawcett, 1997] propose the use of the following criterion: $(1 - FPR) \times TPR$. A geometric interpretation of the criterion is the following [Provost and Fawcett, 1997]: it corresponds to the area of a rectangle whose opposite vertices are (FPR, TPR) and $(1, 0)$. The typical isoperformance

curve is now a hyperbola. The third problem we address is therefore:
Problem 3: **Given γ, can we find an algorithm returning a set of Horn clauses such that $(1 - FPR) \times TPR \geq \gamma$, if such a hypothesis exists ?**
In [Kubat et al., 1998], a criterion is maximized which is the square-root of our criterion. Because of the monotonicity properties of this function, our negative results on problem 3 shall also hold for the criterion of [Kubat et al., 1998].

3.4 Basic tools for the hardness results

Concerning problem 1, fix $a \in \{1, 2, 3, 4, 5, 6, 7\}$. We want to approximate the best concept in $\mathcal{H}_{(D,)a}(\zeta)$ by one still in $\mathcal{H}_{(D,)a}(\zeta)$. However, the best concept in $\mathcal{H}_{(D,)a}(\zeta)$ generally does not have an error equal to the optimal one over $\mathcal{H}$ given D, $opt_{\mathcal{H}_D}(c)$. In fact, it has an error that we can denote

$$\begin{aligned} opt_{\mathcal{H}_{(D,)a}(\zeta)}(c) &= \min_{h' \in H_{(D,)a}(\zeta)} \sum_{h(x) \neq c(x)} D(x) \\ &\geq opt_{\mathcal{H}_D}(c) \end{aligned}$$

The goodness of the accuracy of a concept taken from $\mathcal{H}_{(D,)a}(\zeta)$ should be appreciated with respect to this "constrained" optimum. Our results on problem 1 are all obtained by showing the hardness of solving the following decision problem:

Definition 4 Approx-Constrained($\mathcal{H}, (a, \zeta)$)

- **Name** *: Approx-Constrained($\mathcal{H}, (a, \zeta)$).*
- **Instance** *: A set of negative examples $\mathcal{S}^-$, a set of positive examples $\mathcal{S}^+$, a rational weight $0 < w(x_i) = \frac{n_i}{d_i} < 1$ for each example x_i, a rational $0 \leq \gamma < 1$. We assume that $\sum_{x \in \mathcal{S}^+ \cup \mathcal{S}^-} w(x_i) = 1$.*
- **Question** *: Does there exist a hypothesis $h \in \mathcal{H}_{(D,)a}(\zeta)$ that satisfies $\sum_{h(x) \neq c(x)} w(x) \leq \gamma$?*

Define as n_e the size of the largest example we dispose of. Note that when the constraint is too tight, it can be the case that no hypothesis can actually satisfy it, and therefore

$$\mathcal{H}_{(D,)a}(\zeta) = \emptyset \tag{14}$$

Define as $|h|$ the size of some $h \in \mathcal{H}$ (in our case, it is the number of Horn clauses of h). In the non-empty subset of $\mathcal{H}$ where formulas are the most constrained (*i.e.* strengthening further the constraint gives an empty subset), define $n^*_{\mathcal{H}_{(D,)a}(\zeta)}$ as the size of the *smallest* hypothesis in $\mathcal{H}_{(D,)a}(\zeta)$ (therefore, it is the smallest hypothesis which satisfies the constraint). Then, our reductions all satisfy

$$n^*_{\mathcal{H}_{(D,)a}(\zeta)} \leq (n_e)^3 \tag{15}$$

Note that the constraint makes generally

$$opt_{H_{(D,)a}(\zeta)}(c) > opt_{\mathcal{H}_D}(c) \tag{16}$$

which might seem to be a negative effect of the constraints. However, the reductions all satisfy

$$d_\nu\left(opt_{\mathcal{H}_D}(c), opt_{\mathcal{H}_{(D,)a}(\zeta)}(c)\right) = o(1) \tag{17}$$

i.e. asymptotic optimal accuracies coincide with or without the constraints; here, the limit is taken as the number of distinct predicates of the problem grows towards infinity. In addition, a principal corollary to all our results is that we can suppose that the whole time used to *write* the total set of Horn clauses is assimilated to $\mathcal{O}(n_e)$, for *any* set. By writing time, we mean time of any procedure consisting only in writing down clauses. Examples of such a procedure are "write down all clauses having k literals", or even "write down *all* Horn clauses". Such procedures can be viewed as *for-to*, or *repeat* algorithms. This property authorizes the construction of Horn clause sets having arbitrary sizes, even exponential.
Problem 2 is addressed by studying the complexity of the following decision problem.

Definition 5 ***Approx-Constrained-ROC(*** $\mathcal{H}, \gamma_{FPR}, \gamma_{TPR}$ ***)***

- **Name** *: Approx-Constrained-ROC(*$\mathcal{H}, \gamma_{FPR}, \gamma_{TPR}$*).*
- **Instance** *: A set of negative examples $\mathcal{S}^-$, a set of positive examples $\mathcal{S}^+$, a rational weight $0 < w(x_i) = \frac{n_i}{d_i} < 1$ for each example x_i. We assume that $\sum_{x \in \mathcal{S}^+ \cup \mathcal{S}^-} w(x_i) = 1$.*
- **Question** *: Does there exist a hypothesis $h \in \mathcal{H}$ satisfying $1 - FPR \geq 1 - \gamma_{FPR}$ and $TPR \geq \gamma_{TPR}$?*

Concerning problem 3, the reductions study a single replacement criterion Γ, and the following decision problem.

Definition 6 ***Approx-Constrained-Single(*** $\mathcal{H}, \Gamma, \gamma$ ***)***

- **Name** *: Approx-Constrained-Single(*$\mathcal{H}, \Gamma, \gamma$*).*
- **Instance** *: A set of negative examples $\mathcal{S}^-$, a set of positive examples $\mathcal{S}^+$, a rational weight $0 < w(x_i) = \frac{n_i}{d_i} < 1$ for each example x_i. We assume that $\sum_{x \in \mathcal{S}^+ \cup \mathcal{S}^-} w(x_i) = 1$.*

13

- **Question** : *Does there exist a hypothesis $h \in \mathcal{H}$ satisfying $\Gamma(h) \leq \gamma$?*

4 Results

For the sake of simplicity in stating our results, we abbreviate "Function free Horn Clauses" by the acronym "FfHC".

4.1 Extending the accuracy

Theorem 7 *We have:*
[1] $\forall 0 < \zeta < 1$, *Approx-Constrained*(*FfHC*, $(1, \zeta)$) *is Hard, when* $\nu < (1-\zeta)/\zeta$
[2] $\forall 0 < \zeta < \frac{1}{2}$, *Approx-Constrained*(*FfHC*, $(2, \zeta)$) *is Hard.*
[3] $\forall 0 < \zeta < 1$, *Approx-Constrained*(*FfHC*, $(3, \zeta)$) *is Hard.*
[4] $\forall 0 < \zeta < 1$, *Approx-Constrained*(*FfHC*, $(4, \zeta)$) *is Hard.*
[5] $\forall 0 < \zeta < 1$, *Approx-Constrained*(*FfHC*, $(5, \zeta)$) *is Hard.*
[6] $\forall 0 < \zeta < 1$, *Approx-Constrained*(*FfHC*, $(6, \zeta)$) *is Hard.*
[7] $\forall 0 < \zeta < 1$, *Approx-Constrained*(*FfHC*, $(7, \zeta)$) *is Hard.*

At that point, the notion of "hardness" needs to be clarified. By "Hard" we mean "cannot be solved in polynomial time under some particular complexity assumption". The hypothesis we use is the same as [Hastad, 1996] ($NP \not\subset ZPP$), which involves randomized complexity classes. We refer the reader to the paper of [Hastad, 1996] for further details, not needed here.
Due to the fact that all proofs are essentially based on the same properties, only proof of point [1] is presented in details in appendix 2; the other results presented strictly use the same type of reduction, and are sketched. Also, in appendix 1, we give the proof that all distributions under which our negative results are proven lead to trivial positive results for the same problem when we remove the additional constraint, and optimize the accuracy alone. Negative results for optimizing the accuracy itself would naturally hold when considering the additional constraints, unless pathological situations are created, such as when the constraint is so tight and removes so many hypotheses that the set of constrained hypotheses has small size (*e.g.* polynomial), and can be explored in polynomial time. We therefore prove that for reasonable constraints, optimizing the accuracy under constraint is a strictly more difficult problem, with non-trivial additional drawbacks. Furthermore, the upperbound error value (γ in definition 4) in constraints $4, 5, 6$ can be fixed arbitrarily in $]0, 1/2[$, which shows that almost removing the accuracy's constraint does not make the problem easier: requiring the Horn clauses to perform slightly better than the unbiased coin leads also to intractability.

14

4.2 Replacing the accuracy: the ROC analysis

In this section, we show that the classical ROC components as described by [Provost et al., 1998], [Provost and Fawcett, 1998] lead to the same results as those we claimed for the preceding bi-criteria optimizations. The problem is all the more difficult as the difficulty appears as soon as we choose to use ROC analysis, and is not a function of the ROC bounds.

Theorem 8 $\forall 0 < \gamma_{FPR}, \gamma_{TPR} < 1$, *Approx-Constrained-ROC(FfHC,$\gamma_{FPR}, \gamma_{TPR}$) is Hard.*

The distribution under which the negative result is proven is an easy distribution for the accuracy's optimization alone, similarly to those of subsection 4.1.

4.3 Replacing the accuracy by a single criterion

The negative result stated in the following theorem is to be read with all additional drawbacks mentioned for the seven constraints. Again, the distribution under which the theorem is proven is an easy distribution when optimizing the accuracy alone.

Theorem 9 $\exists \gamma_{max} > 0$ *such that* $\forall 0 < \gamma < \gamma_{max}$, *Approx-Constrained-Single(FfHC,$(1 - FPR) \times TPR, \gamma$) is Hard.*

(Proof included in appendix 2). As far as we know, $\gamma_{max} \geq \frac{175}{41616}$ (roughly 4.2×10^{-3}), but we think that this bound can be much improved. The accuracy can sometimes be conveniently replaced by the F_β statistics [Cohen, 1995], which is an accurate composition of precision and recall (see subsection 3.1 for their definition), useful for text categorization problems [Cohen, 1995]. So far, we have not been able to conclude to the hardness of using this criterion in our framework.

4.4 Beyond computational complexity and ILP

It is well-known since [Kearns et al., 1987] that negative results on such problems can sometimes be extended to negative results for PAC-type learning models [Valiant, 1984]. Such a model typically brings a statistical and a computational constraint for an algorithm to be qualified as a learning algorithm. Consider for example definition 4, and the following learning model working in exactly the same setting, but in which we replace the set of examples by a so-

called oracle [Kearns et al., 1987], drawing examples on demand, following a probability distribution D unknown, but fixed. Suppose that the requirement on the constraint defining $\mathcal{H}_{(D,)a}(\zeta)$ remains exactly the same, but the one limiting the accuracy on the "learning" sample is replaced by a condition which states that, with sufficiently high probability ($> 1-\delta$), the accuracy over the whole domain is lower than some threshold ($< opt_{\mathcal{H}_{(D,)a}(\zeta)}(c) + \epsilon$), for some parameters $\epsilon, \delta > 0$. If we require that the computational time be a polynomial in $1/\epsilon, 1/\delta$, as well as in n_e and the (smallest) size of the optimal constrained hypothesis, then the learning model we obtain corresponds to the robust learning model of [Höffgen and Simon, 1992], [Jappy et al., 1996], to which we require the outputs to satisfy a constraint (among our seven first constraints). In that case, following a standardized approach ([Höffgen and Simon, 1992], [Kearns et al., 1987], [Jappy et al., 1996]), it is easy to show that a negative result to definition 4 can be translated to a negative constrained robust learning result.
Apart from the extension of the results to learning models, a natural question is their extension to other formalisms, outside the ILP field. So far, as ILP is a complex formalism, the results can be extended to simpler formalisms such as some subclasses of Boolean formulas like the subclass of DNF (disjunctive normal form formulas, [Valiant, 1985]) containing all monotonous formulas (without negative literals). Note that we do not put any restriction on the size of the formulas, a very seldom result in the huge quantity of theoretical ML results on DNF. Indeed, DNF is one of the most central classes to the PAC learning model of [Valiant, 1984], studied early by Valiant himself [Valiant, 1985], and still raising some of the most important problems in computational learning theory [Klivans and Servedio, 2001], in particular for its learnability or approximability properties. In that setting, removing the monotonicity constraint in our results is certainly a problem which would deserve further investigations.

5 Conclusion

In this paper, we have presented a new approach to the problem of the accuracy's replacement in machine learning, a problem which have recently received attention in a growing number of papers. We have argued that the usual criticisms, against the use of the accuracy for comparing the reliability of classifiers as well as for being optimized to build classifiers, face complexity issues. The case against accuracy, as proposed in [Provost et al., 1998], is therefore more complicated than usually presented in machine learning or data mining papers. This justifies the title of our paper, which can be read in two ways, either presenting new aspects to the difficulty (complexity) of the task to find new criteria to replace the accuracy, or presenting (structural) complexity is-

sues about the possible replacements. One important thing about our results is that the complexity results go beyond the usual intractability results found in machine learning. In our case indeed, there are some side effects, rather surprising, proving that the difficulty of the learning task, when the accuracy is replaced, is accompanied by severe drawbacks on the formalism's expressiveness. In deep contrast, the optimization of the accuracy alone in our setting is trivial, since the optimal solutions can be found directly without any algorithmic effort.
Recently, a new approach to building classifiers has been proposed, which also argue for the replacement of the accuracy for being optimized in the induction of classifiers [Schapire and Singer, 1998], in certain circumstances. This approach, called boosting, has been plebiscited as one of the best currently available in classification [Friedman et al., 2000]. However, in some cases, it raises conjectures about the tractability of the optimization of these new criteria [Schapire and Singer, 1998]. This shall certainly be the subject of future studies.

References

[Boucheron, 1992] Boucheron, S. (1992). *Théorie de l'apprentissage, de l'approche formelle aux enjeux cognitifs*. Hermes.

[Cohen, 1995] Cohen, W. W. (1995). Text categorization and relational learning. In *Proceedings of the 12 th International Conference on Machine Learning*, pages 124–132.

[Domingos, 1999] Domingos, P. (1999). MetaCost: a general method for making classifiers cost-sensitive. In *Proceedings of the 5th International Conference on Knowledge Discovery in Databases*, pages 155–164.

[Dzeroski et al., 1992] Dzeroski, S., Muggleton, S., and Russel, S. (1992). Pac-learning of determinate logic programs. In *Proceedings of the 5 th International Conference on Computational Learning Theory*, pages 128–137.

[Egan, 1975] Egan, J. P. (1975). *Signal detection theory and ROC analyses*. Academic press.

[Friedman et al., 2000] Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive Logistic Regression : a Statistical View of Boosting. *Annals of Statistics*, 28:337–374.

[Garey and Johnson, 1979] Garey, M. and Johnson, D. (1979). *Computers and Intractability, a guide to the theory of NP-Completeness*. Bell Telephone Laboratories.

[Hastad, 1996] Hastad, J. (1996). Clique is hard to approximate within $n^{1-\epsilon}$. In *Proceedings of the 37 th IEEE Symposium on the Foundations of Computer Science*, pages 627–636.

[Höffgen and Simon, 1992] Höffgen, K.-U. and Simon, H. (1992). Robust trainability of single neurons. In *Proceedings of the 5 th International Conference on Computational Learning Theory*.

[Holte, 1993] Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning Journal*, 11:63–91.

[Jappy et al., 1996] Jappy, P., Nock, R., and Gascuel, O. (1996). Negative robust learning results for horn clause programs. In *Proceedings of the 13 th International Conference on Machine Learning*, pages 258–265.

[Kearns et al., 1987] Kearns, M., Li, M., Pitt, L., and Valiant, L. (1987). On the learnability of boolean formulae. In *Proceedings of the 19 th ACM Symposium on the Theory of Computing*, pages 285–295.

[Kietz and Dzeroski, 1994] Kietz, J. and Dzeroski, S. (1994). Inductive logic programming and learnability. *Sigart Bulletin*, 5:22–32.

[Kietz, 1993] Kietz, J.-U. (1993). Some lower bounds for the computational complexity of inductive logic programming. In *Proceedings of the 8 th European Conference on Machine Learning*, pages 115–123.

[Klivans and Servedio, 2001] Klivans, A. R. and Servedio, R. (2001). Learning DNF in time $2^{\tilde{O}(n^{\frac{1}{3}})}$. In *Proceedings of the 33 th ACM Symposium on the Theory of Computing*, pages 258–265.

[Kubat et al., 1998] Kubat, M., Holte, R. C., and Matwin, S. (1998). Machine Learning for the detection of oil spills in satellite radar images. *Machine Learning Journal*, 30:195–215.

[Kukar et al., 1997] Kukar, M., Besic, N., Kononenko, I., Auersperg, M., and Robnik-Sikonja, M. (1997). Prognosing the survival time of the patients with anaplastic thyroid carcinoma using machine learning. In Lavrac, N., Karavnou, E., and Zupan, B., editors, *Intelligent Data Analysis in Medicine and Pharmacology*, pages 116–129. Kluwer Academic Publishers.

[Nock and Gascuel, 1995] Nock, R. and Gascuel, O. (1995). On learning decision committees. In *Proceedings of the 12 th International Conference on Machine Learning*, pages 413–420.

[Nock and Jappy, 1998] Nock, R. and Jappy, P. (1998). On the power of decision lists. In *Proceedings of the 15 th International Conference on Machine Learning*, pages 413–420.

[Provost and Fawcett, 1997] Provost, F. and Fawcett, T. (1997). Analysis and visualization of classifier performance : comparison under imprecise class and cost distributions. In *Proceedings of the 3rd International Conference on Knowledge Discovery in Databases*, pages 43–48.

[Provost and Fawcett, 1998] Provost, F. and Fawcett, T. (1998). Robust classification systems for imprecise environments. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*, pages 706–713.

[Provost et al., 1998] Provost, F., Fawcett, T., and Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15 th International Conference on Machine Learning*, pages 445–453.

[Schapire and Singer, 1998] Schapire, R. E. and Singer, Y. (1998). Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the 11 th International Conference on Computational Learning Theory*, pages 80–91.

[Sebban and Nock, 2002] Sebban, M. and Nock, R. (2002). A hybrid filter/wrapper approach of feature selection using information theory. *Pattern Recognition*, 35:835–846.

[Slattery and Craven, 1998] Slattery, S. and Craven, M. (1998). Combining statistical and relational methods for learning in hypertext domains. In *Proceedings of the 8 th International Conference on Inductive Logic Programming*, pages 38–52.

[Srinivasan, 1998] Srinivasan, A. (1998). Applications of ILP to problems in chemistry and biology. Invited talk at the *Eighth International Conference on Inductive Logic Programming*.

[Valiant, 1984] Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27:1134–1142.

[Valiant, 1985] Valiant, L. G. (1985). Learning disjunctions of conjunctions. In *Proceedings of the 9 th International Joint Conference on Artificial Intelligence*, pages 560–566.

[Vapnik, 1998] Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley.

6 Appendix 1: the global reduction

Reductions are achieved from the NP-Complete problem "Clique" [Garey and Johnson, 1979]:

Definition 10 ***"Clique"***

- **Name** *: Clique.*
- **Instance** *: A graph $G = (X, E)$, an integer k.*
- **Question** *: Does there exists a clique of size $\geq k$ in G?*

Of course, "Clique" is not hard to solve for any value of k. The following lemma establishes values of k for which we can suppose that the problem is hard to solve ($\binom{n}{k} = n!/((n-k)!k!)$ is the binomial coefficient):

Theorem 11 *(i) We can suppose that $\binom{k}{2} \leq |E|$, and k is not a constant, otherwise "Clique" is polynomial. (ii) For any $\alpha \in]0, 1[$, "Clique" is hard for the value $k = \alpha|X|$ or $k = |X|^\alpha$.*

19

PROOF. (i) is immediate ; (ii) follows from [Hastad, 1996]: it is proven that the largest clique size is not approximable to within $|X|^\beta$, for any constant $0 < \beta < 1$. Therefore, the graphs generated have a clique number which is either l, or greater than $l \times |X|^\beta$, with $l < |X|^{1-\beta}$. Therefore, the decision problem is intractable for values of k greater than l, which is the case if $k = \alpha|X|$ or $k = |X|^\alpha$, with $\alpha \in]0,1[$.

The structure of the examples is the same for any of our reductions.

- Define a set of $|X|$ unary literals $a_1(.), ..., a_{|X|}(.)$, in bijection with the vertices of G. To this set of literals, we add two unary literals, $s(.)$ and $t(.)$. The inferred predicate is denoted q. The choice of unary predicates is made only for a simplicity purpose. We could have replaced each of them by l-ary predicates without changing our proof.
- Define a set of constant symbols useful for the description of the examples:

$$\{l_{i,j}, \forall(i,j) \in E\} \cup \{l_1, l_2, l_3, l_4\} \cup \{m_i, \forall i \in \{1, ..., |X|\}\}$$

Examples are described in the following way:

- Positive examples from $\mathcal{S}^+$:

$$\forall(i,j) \in E, p_{i,j} = q(l_{i,j}) \leftarrow \wedge_{k\in\{1,...,|X|\}\setminus\{i,j\}} a_k(l_{i,j}) \wedge t(l_{i,j}) \tag{18}$$
$$p_1 = q(l_1) \leftarrow \wedge_{k\in\{1,...,|X|\}} a_k(l_1) \wedge t(l_1) \tag{19}$$
$$p_2 = q(l_2) \leftarrow a_1(l_2) \tag{20}$$

- Negative examples from $\mathcal{S}^-$:

$$\forall i \in \{1, ..., |X|\}, n_i = q(m_i) \leftarrow \wedge_{k\in\{1,...,|X|\}\setminus\{i\}} a_k(m_i) \wedge t(m_i) \tag{21}$$
$$n'_1 = q(l_3) \leftarrow \wedge_{k\in\{1,...,|X|\}} a_k(l_3) \wedge s(l_3) \wedge t(l_3) \tag{22}$$
$$n'_2 = q(l_4) \leftarrow \wedge_{k\in\{1,...,|X|\}} a_k(l_4) \wedge s(l_4) \tag{23}$$

It comes that we always have $n_{\mathcal{H}_{(D,)a}(\zeta)} = \mathcal{O}(|X|^3)$ (this is the coding size of the positive examples) and $n_e = \mathcal{O}(|X|)$. Non-uniform weights are given to each example, depending on the constraint to be tackled with. The common-point to all reductions is that the weights of all examples n_j (resp. all $p_{i,j}$) are equal (resp. to w^- and w^+). In each reduction, examples and clauses satisfy:

H$_1$ p_2 is forced to be badly classified.
H$_2$ n'_1 is always badly classified.
H$_3$ $w(n'_2)$ ensures that n'_2 is always given the right class, forcing any clause to contain literal $t(.)$.

When we remove n'_2, we also ensure that p_2 is removed too.

Lemma 12 *Any clause containing literal $s(.)$ can be removed.*

20

PROOF. Suppose that one clause contains $s(.)$. Then it can be θ-subsumed by n'_1 and by no other example (even if n'_2 exists, because of $\mathbf{H}_3$); but n'_1 θ-subsumes any clauses and also the empty clause. Therefore, removing the clause does not modify the value of any criteria based on the examples weights. Concerning the sixth (resp. seventh) constraint, the fraction of predicates used after removing the clause is at most the one before, thus, if the clause is an element of $\mathcal{H}_6(\zeta)$ (resp. $\mathcal{H}_7(\zeta)$) before, it is still an element after.

As a consequence, p_1 is always given the positive class (even by the empty clause!).

We now give a general outline of the proof for Problem 1 ; reductions are similar for the other problems. Given $h = \{h_1, ..., h_l\}$ a set of Horn clauses, we define the set

$$\mathcal{I} = \{i \in \{1, ..., |X|\} : \exists j \in \{1, ..., l\}, a_i(.) \notin h_j\}$$

and we fix $|\mathcal{I}| = k'$. In our proofs, we define two functions taking rational values, $E(k')$ and $F_a(k')$ ($k' \in \{1, ..., |X|\}$, $a = 1, 2, 3, 4, 5, 6, 7$). They are chosen such that:

- $E(k')$ is strictly increasing, $\sum_{(x \in \mathcal{S}^+ \wedge h(x)=0) \vee (x \in \mathcal{S}^- \wedge h(x)=1)} w(x) \geq E(k')$ and $E(k) = \gamma$.
- $F_a(k')$ is strictly decreasing, it is a lowerbound of the function inside $\mathcal{H}_{(D,)a}(\zeta)$, and $F_a(k) = \zeta$ (excepted for $a = 3$, $F_3(k) = 1/\zeta$).

$\forall a \in \{1, 2, 3, 4, 5, 6, 7\}$, if there exists an unbounded set of Horn clauses $h \in \mathcal{H}_{(D,)a}(\zeta)$ satisfying $\sum_{(x \in \mathcal{S}^+ \wedge h(x)=0) \vee (x \in \mathcal{S}^- \wedge h(x)=1)} w(x) \leq \gamma$, its error rate implies $k' \leq k$ and constraint implies $k' \geq k$. So $|\mathcal{I}| = k' = k$. The interest of the weights is then to force $\binom{k}{2}$ positive examples from the set $\{p_{i,j}\}_{(i,j) \in E}$ to be well classified, while we ensure the misclassification of at most k negative examples of the set $\{n_i\}_{i \in \{1, ..., |X|\}}$. It comes that the $\binom{k}{2}$ correspond to the $\binom{k}{2}$ edges linking the $|\mathcal{I}| = k$ vertices corresponding to negative examples badly classified. We therefore dispose of a clique of size $\geq k$.
Conversely, $\forall a \in \{1, 2, 3, 4, 5, 6, 7\}$, given some clique of size k whose set of vertices is denoted $\mathcal{I}$, we show that the singleton

$$h = q(X) \leftarrow \wedge_{i \in \{1, ..., |X|\} \setminus \mathcal{I}} a_i(X) \wedge t(X)$$

is an element from $\mathcal{H}_{(D,)a}(\zeta)$ satisfying $\sum_{(x \in \mathcal{S}^+ \wedge h(x)=0) \vee (x \in \mathcal{S}^- \wedge h(x)=1)} w(x) \leq \gamma$.
In this case, $n_{\mathcal{H}_{(D,)a}(\zeta)}$ drops down to $\mathcal{O}(n_e)$.
All distributions used in theorems 7 and 9 are such that $w^+ < w^-/|X|$, at

least for graphs exceeding a fixed constant size. Also, due to the negative examples of weights w^-, if we remove the additional constraints and optimize the accuracy alone, we can suppose that the optimal Horn clause is a singleton: merging all clauses by keeping over predicates $a_j(.)$ only those present in all clauses does not decrease the accuracy. Under such a distribution, the optimal Horn clause necessarily contains all predicates $a_j(.)$, and the problem becomes trivial. The distribution in theorem 8 satisfies $w^+ = w^-$. This is also a simple distribution for the accuracy's optimization alone: indeed, the optimal Horn clause over predicates $a_j(.)$ is such that it contains no predicates $a_j(.)$ that does not appear at least in one positive example. If the graph instance of "Clique" is connex (and we can suppose so, otherwise the problem boils down to find the largest clique in one of the connected components), then the optimal Horn clause does not contain any of the $a_j(.)$.

7 Appendix 2: proofs of negative results

7.1 Proof of point [1], theorem 7

We fix the following weights for positive examples:

$$\begin{aligned} w(p_2) &= \frac{1}{2(1-\zeta)}\left(\zeta\nu + |X|^2 w^-(1+\zeta)\right) \\ \forall (i,j) \in E, w(p_{i,j}) = w^+ &= \frac{w^-}{(|X|+k)^2} \end{aligned}$$

$$\begin{aligned} w(p_1) = &\frac{1}{2}\left(1 - \frac{\zeta\nu}{1-\zeta}\right) \\ &-\frac{1}{2}\left(w^-\left[|X|^2\left(\frac{1+\zeta}{1-\zeta} + |X| - k\right)\right]\right) \\ &-\frac{1}{2}\left(w^+\left[\frac{1-\zeta}{1+\zeta}\left(|X| - \binom{k}{2}\right) + |X|\right]\right) \end{aligned}$$

We fix the following weights for negative examples:

$$\begin{aligned} w(n'_2) &= \frac{1}{2} \\ \forall j \in \{1,...,|X|\}, w(n_j) = w^- &= \frac{1}{|X|^2|E|^2} \end{aligned}$$

$$w(n_1') = \frac{1}{2}\left(\frac{1-\zeta}{1+\zeta}\left(|E| - \binom{k}{2}\right)w^+ + (|X|^2 - k)w^-\right)$$

Fix $\gamma = \frac{w(p_2)+w(n_1')+kw^- + (|E|-\binom{k}{2})w^+}{2}$ (note that $w(n_2')$ ensures that n_2' is given the right class), and $k_{\max} = 1 + \max_{2\leq k''\leq |X|;|E|-\binom{k''}{2}\geq 0} k''$. From the choice of weights, $\mathrm{lcm}\left(\cup_{x_i\in\mathcal{S}^+\cup\mathcal{S}^-} d_i\right) = \mathcal{O}(|X|^8)$ ("lcm" is the least common multiple), which is polynomial. Define the functions:

$$\begin{aligned}
\forall k' \in \{0;1\}, E(k') &= |E|w^+ + k'w^- + w(p_2) + w(n_1)\\
\forall 2 \leq k' \leq k_{\max}, E(k') &= \left(|E| - \binom{k'}{2}\right)w^+ + k'w^- + w(p_2) + w(n_1)\\
\forall k_{\max} < k' \leq |X|, E(k') &= k'w^- + w(p_2) + w(n_1)
\end{aligned}$$

(from the choice of weights, $E(k) = \gamma$),

$$\begin{aligned}
\forall k' \in \{0;1\}, F_1(k') &= \frac{||E|w^+ - k'w^- + w(p_2) - w(n_1)|}{\nu + |E|w^+ + k'w^- + w(p_2) + w(n_1)}\\
\forall 2 \leq k' \leq k_{\max}, F_1(k') &= \frac{|\left(|E| - \binom{k'}{2}\right)w^+ - k'w^- + w(p_2) - w(n_1)|}{\nu + |E|w^+ + k'w^- + w(p_2) + w(n_1)}\\
\forall k_{\max} < k' \leq |X|, F_1(k') &= \frac{|-k'w^- + w(p_2) - w(n_1)|}{\nu + |E|w^+ + k'w^- + w(p_2) + w(n_1)}
\end{aligned}$$

(from the choice of weights, $F_1(k) = \zeta$).
The equation obtained when $k' < k_{\max}$ takes its maximum for integer values when $k' = (|X| + k)^2 + 0.5 \pm 0.5 > |X|$. Furthermore,

$$\forall 1 \leq k_{\max} \leq |X|, \left(|E| - \binom{k_{\max}-1}{2}\right)w^+ < w^-$$

which leads to $E(k_{\max} - 1) < E(k_{\max})$. In a more general way, $E(k')$ is strictly increasing over natural integers. Now remark that the numerator of $F_1(k')$ is strictly decreasing, and its denominator strictly increasing. Therefore, $F_1(k')$ is strictly decreasing. Furthermore

$$d_\nu\left(\sum_{h(x)\neq 1=c(x)} w(x); \sum_{h(x)\neq 0=c(x)} w(x)\right) \geq F_1(k')$$

If $\exists h \in \mathcal{H}_{\{w_i\},1}(\zeta)$ satisfying $\sum_{h(x)\neq c(x)} w(x) \leq \gamma$, the error rate implies $k' \leq k$ and the constraint implies $k' \geq k$. Thus $|\mathcal{I}| = k' = k$. As pointed out in the preceding appendix, this leads to the existence of a clique of size $\geq k$.

Reciprocally, the Horn clause h constructed in Appendix 1 satisfies both relations $h \in \mathcal{H}_{\{w_i\},1}(\zeta)$, and $\sum_{h(x)\neq c(x)} w(x) \leq \gamma$. Indeed, we have

$$\sum_{h(x)\neq 1=c(x)} w(x) = \left(|E| - \binom{k}{2}\right) w^{+} + w(p_2)$$

but also

$$\sum_{h(x)\neq 0=c(x)} w(x) = kw^{-} + w(n_1)$$

Therefore,

$$d_\nu \left(\sum_{h(x)\neq 1=c(x)} w(x); \sum_{h(x)\neq 0=c(x)} w(x) \right) = F_1(k) = \zeta$$

and $h \in \mathcal{H}_{\{w_i\},1}(\zeta)$. We also have $\sum_{h(x)\neq c(x)} w(x) = E(k) = \gamma$.
The reduction is achieved. We end by a remark on $d_\nu \left(opt_{\mathcal{H}_{\{w_i\}}}(c), opt_{\mathcal{H}_{\{w_i\},1}(\zeta)}(c) \right)$.
We have

$$\begin{aligned} |opt_{\mathcal{H}_{\{w_i\}}}(c) - opt_{\mathcal{H}_{\{w_i\},1}(\zeta)}(c)| &\leq 1 - w(p_1) - w(p_2) - w(n_1') - w(n_2') \\ &\leq |E|w^{+} + |X|w^{-} \end{aligned}$$

and

$$\begin{aligned} opt_{\mathcal{H}_{\{w_i\}}}(c) + opt_{\mathcal{H}_{\{w_i\},1}(\zeta)}(c) + \nu &\geq 2 \times (w(p_2) + w(n_1')) + \nu \\ &\geq \frac{\zeta\nu}{(1-\zeta)} + \nu \end{aligned}$$

Therefore, we get

$$d_\nu \left(opt_{\mathcal{H}_{\{w_i\}}}(c), opt_{\mathcal{H}_{\{w_i\},1}(\zeta)}(c) \right) = o(1)$$

7.2 Sketch of proof of points [6] and [7], theorem 7

The proof of these two points is easier than the others. Let us consider the sixth constraint to illustrate it. The function F_6 is exactly a decreasing function of the "holes" k', which we can write

$$\forall k' \in \{0, 1, |X|\}, F_6(k') = \frac{|X| - k'}{|X|}$$

24

Fix γ strictly between 0 and $\frac{1}{2}$ (thus, the error is only slightly better than that of the unbiased coin). Weights are as follows for positive examples (we do not use p_1):

$$\forall (i,j) \in E, w(p_{i,j}) = w^+ = \frac{w^-}{(|X|+k)^2}$$

$$w(p_2) = \gamma - kw^- - \left(|E| - \binom{k}{2}\right) w^+$$

Weights are as follows for negative examples (we do not use n'_1):

$$\forall j \in \{1, ..., |X|\}, w(n_j) = w^- = \frac{1}{|X|^2|E|^2}$$

$$w(n'_2) = 1 - \gamma - (|X| - k)w^- - \binom{k}{2} w^+$$

Fix $k_{\max} = 1 + \max_{2 \leq k'' \leq |X|; |E| - \binom{k''}{2} \geq 0} k''$. From the weights, $\mathrm{lcm}\,(\cup_{x_i \in \mathcal{S}^+ \cup \mathcal{S}^-} d_i) = \mathcal{O}(|X|^8)$, which is polynomial. Define the function:

$$\begin{aligned}
\forall k' \in \{0;1\}, E(k') &= |E|w^+ + k'w^- + w(p_2) \\
\forall 2 \leq k' \leq k_{\max}, E(k') &= \left(|E| - \binom{k'}{2}\right) w^+ + k'w^- + w(p_2) \\
\forall k_{\max} < k' \leq |X|, E(k') &= k'w^- + w(p_2)
\end{aligned}$$

(We have $E(k) = \gamma$). From that, it comes that the predicates which are not used can form a Clique.

There remains to check the constraint values ζ which we allowed to take any value in $]0,1[$. From lemma 10, we may use $k = \theta(|X|^\alpha)$, for any $0 < \alpha < 1$. The fraction of authorized predicates is therefore upperbounded by

$$\frac{|X| - k}{|X|} \leq 1 - \frac{1}{|X|^\alpha} \rightarrow_\infty 1$$

By considering sufficiently large sized graphs, the right side is greater than any chosen constant $0 < \zeta < 1$. Point [7] is achieved in the same way.

25

7.3 Proof of theorem 9

Remark that $TPR \times (1 - FPR) = TPR \times TNR$. Weights are as follows for positive examples (we do not use p_2):

$$\forall (i,j) \in E, w(p_{i,j}) = w^+ = \frac{\gamma}{(|X|-k)w^- \times \left(\binom{k}{2} + \frac{(|X|+1)^2 - \left(k - \frac{|X|+1}{3}\right)^2 - 3|X|}{6}\right)}$$

$$w(p_1) = w^+ \times \left(\frac{(|X|+1)^2 - \left(k - \left(\frac{|X|+1}{3}\right)\right)^2 - 3|X|}{6}\right)$$

Weights are as follows for negative examples (we do not use n'_2):

$$\forall j \in \{1, ..., |X|\}, w(n_j) = w^- = \frac{1}{|X|+k}$$

$$w(n'_1) = 1 - |E|w^+ - |X|w^- - w(p_1)$$

The choice of γ_{max} comes from the necessity of a tight calculation of the weights, in order to keep them in correct limits. In order to illustrate this, we proceed through the proof of the correct values for the weights. The positive values of all weights (except from n'_1, whose correctness stems from the study of all the other weights) is easily checked. However, we need to prove that they all take values which do not give a negative weight value to n'_1.

Fix $k = \alpha|X|$, where α takes the adequate value $\alpha = \frac{5}{12}$ (other ones are possible, also valid according to theorem 11, but we concentrate on this one).

Remark that $|X|w^- = \frac{1}{1+\alpha} = \frac{12}{17} < 1$. Now, we study $|E|w^+$. We have

$$w^+ = \frac{\gamma(|X|+k)}{(|X|-k)\left(\binom{k}{2} + \frac{(|X|+1)^2 - \left(k - \frac{|X|+1}{3}\right)^2 - 3|X|}{6}\right)}$$

Note that $\frac{\gamma(|X|+k)}{(|X|-k)} = \frac{17\gamma}{7}$. Suppose we choose $\gamma \leq \frac{1}{17} \times \frac{7 \times 25}{17 \times 144}$ (details of γ are given for the proofs). Then, with such values, we have for the denominator of

26

w^+:

$$\left(\binom{k}{2} + \frac{(|X|+1)^2 - \left(k - \frac{|X|+1}{3}\right)^2 - 3|X|}{6}\right) \leq \frac{25}{144 \times 2}|X|^2$$

which leads to an upperbound for $|E|w^+$ which is (taking into account that $|E| < \frac{|X|^2}{2}$:

$$|E|w^+ < \gamma \times \frac{17}{7} \times \frac{144 \times 2}{25 \times 2} = \frac{1}{17}$$

An upperbound of $\gamma \times \frac{17}{7} \times \frac{144\times 2}{25 \times |X|^2}$ is also available for w^+, which leads to (for $w(p_1)$):

$$w(p_1) < \gamma \times \frac{17}{7} \times \frac{144 \times 2}{25 \times |X|^2} \times \frac{|X|^2}{6} < \frac{1}{17}$$

This shows that the weight of n'_1 is positive, as we claimed.

Now, we explain more in depth the proof scheme by describing a polynomial of order 3, $F(k')$ which upperbounds $TPR \times TNR$, and of course has the desirable property of having its maximum for $k' = k$, with value γ, and with no other equal or greater values on the interval $[0, |X|]$. Similarly to the other proofs, the value γ can only be reached when $k' = k$ represents k "holes" among predicates $\{a_j(.)\}$, and this induces a size-k clique in the graph.

Fix $k_{\max} = 1 + \max_{2 \leq k'' \leq |X|; |E| - \binom{k}{2} \geq 0} k''$. From the weights, $\mathrm{lcm}\,(\cup_{x_i \in \mathcal{S}^+ \cup \mathcal{S}^-} d_i) = \mathcal{O}(|X|^8)$, which is polynomial. Define the function:

$$\begin{aligned}
\forall k' \in \{0;1\}, F(k') &= w(p_1) \times (|X| - k')w^- \\
\forall 2 \leq k' \leq k_{\max}, F(k') &= \left(\binom{k'}{2} w^+ + w(p_1)\right) \times (|X| - k')w^- \\
\forall k_{\max} < k' \leq |X|, F(k') &= \left(|E|w^+ + w(p_1)\right) \times (|X| - k')w^-
\end{aligned}$$

With our choice of weights, and inside the values of k' for which we described k (clearly, in the second curve), F describes a polynomial of degree 3, with a second-order derivative taking its zero for $k' = k'' = \frac{|X|+1}{3}$. Its first-order derivative takes its zeroes respectively for $k' = k'_0 \in [0, k'']$ and $k' = k'_1 = k > k''$ (note that the choice of α respects this latter inequality). Outside $[k'_0, k]$, F is decreasing, and increasing inside. Since the choice of weights was also made

27

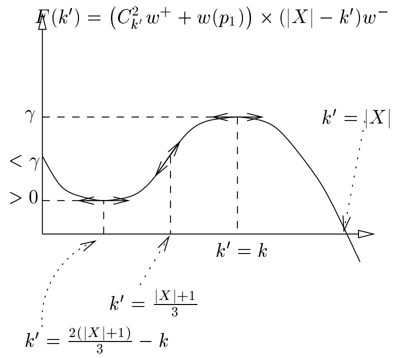

Fig. 2. Scheme of $F(k')$.

so as to have $F(0) < \gamma$, and $F(k) = \gamma$, this is sufficient to prove that there is only one point for $k' = k$ where F takes a value of γ, with lower values elsewhere (Figure 2 shows a simplified view of the function, for the sake of clarity). As we pointed out before, F upperbounds the product of TPR and TNR of any set of Horn clauses, which leads to a single favorable case: the "holes" inside the set of Horn clauses describe a clique of size k in the graph.

A.4 [Nock and Sebban, 2000b]

Sharper Bounds for the Hardness of Prototype and Feature Selection

Richard Nock[1] and Marc Sebban[2]

[1] Université des Antilles-Guyane
Dépt Scientifique Interfacultaire, Campus de Schoelcher
97233 Schoelcher, France
Richard.Nock@martinique.univ-ag.fr
[2] Université des Antilles-Guyane
Dépt de Sciences Juridiques, Campus de Fouillole
97159 Pointe-à-Pitre, France
Marc.Sebban@univ-ag.fr

Abstract. As pointed out by Blum [Blu94], "nearly all results in Machine Learning [...] deal with problems of separating relevant from irrelevant information in some way". This paper is concerned with structural complexity issues regarding the selection of relevant Prototypes or Features. We give the first results proving that both problems can be much harder than expected in the literature for various notions of relevance. In particular, the worst-case bounds achievable by any efficient algorithm are proven to be very large, most of the time not so far from trivial bounds. We think these results give a theoretical justification for the numerous heuristic approaches found in the literature to cope with these problems.

1 Introduction

With the development and the popularization of new data acquisition technologies such as the World Wide Web (WWW), computer scientists have to analyze potentially huge data sets. The available technology to analyze data has been developed over the last decades, and covers a broad spectrum of techniques and algorithms. The overwhelming quantities of such easy data represent however a noisy material for learning systems, and filtering it to reveal its most informative content has become an important issue in the fields of Machine Learning (ML) and Data Mining.

In this paper, we are interested in two important aspects of this issue: the problem of selecting the most relevant examples (named prototypes), a problem to which we refer as "Prototype selection" (PS), and the problem of selecting the most relevant variables, a problem to which we refer as "Feature selection" (FS). Numerous works have addressed empirical results about efficient algorithms for PS and FS [Koh94,KS95,KS96,SN00a,SN00b,Ska94,WM97] and many others. However, in comparison, very few results have addressed the theoretical issues of both PS and FS, and more particularly have given insight into the hardness of FS and PS. This is an important problem because almost all efficient algorithms presented so far for PS or FS are heuristics, and no theoretical results are given for the guarantees they give on the selection process. The question of their behavior in the worst case is therefore of particular importance. Structural complexity theory can be helpful to prove lowerbounds valid for any time-efficient algorithm, and negative results for approximating optimization problems are important in that they may indicate we can stop looking for better algorithms [Bel96]. On some problems [KKLP97], they have even ruled out the existence of efficient approximation algorithms in the worst case.

In this paper, we are interested in PS and FS as optimization problems. So far, one theoretical result exists [BL97], which links the hardness of approximating FS and the hardness of approximating the MIN-SET-COVER problem. We are going to prove in that paper that PS and FS are very hard problems for various notions of what is "relevance", and our results go far beyond the negative results of [BL97]. The main difficulty in our approach is to capture the essential notions of relevance for PS and FS. As underlined in [BL97], there are many definitions for relevance, principally motivated by the question "relevant to what?", and addressing them separately would require large room space. However, these notions can be clustered according to different criteria, two of which seem to be of particular interest. Roughly speaking, relevance is generally to be understood with respect to a *distribution*, or with respect to a *concept*. While the former encompasses *information* measures, the latter can be concerned with the *target* concept (governing the labeling of the examples) or the *hypothesis* concept built by a further induction algorithm. In this work, we have chosen to address two notions of relevance, each representative of one cluster, for each of the PS and FS problems.

We prove for each of the four problems, that any time-efficient algorithm shall obtain very bad results in the worst case, much closer than expected to the "performances" of approaches consisting in *not* (or randomly) filtering the data ! From a practical point of view, we think our results give a theoretical justification to heuristic approaches of FS and PS. While these hardness results have the advantage of covering the basic notions of relevance found throughout the literature (of course by investigating four *particular* definitions of relevance), they have two technical commonpoints. First, the results are obtained by reduction from the same problem (MIN-SET-COVER), *but* they do *not* stem from a simple coding of the instance of MIN-SET-COVER. Second, the proofs are standardized: they all use the same reduction tool but in a different way. From a technical point of view, the reduction technique makes use of *blow-up reductions*, a class of reductions between optimization problems previously sparsely used in Computational Learning Theory [HJLT94,NJ98a,NJS98]. Informally, blow-up reductions (also related to *self-improving reductions*, [Aro94]) are reductions which can be made from a problem onto itself: the transformation is such that it depends on an integer d which is used to tune the hardness result: the higher d, the larger the inapproximability ratio obtained. Of course, there is a price to pay : the reduction time is also an increasing function of d; however, sometimes, it is possible to show that the inapproximability ratio can be blown-up *e.g.* up to *exponent* d, whereas the reduction time increases reasonably as a function of d, *e.g. linearly* [NJS98].

The remaining of this paper is organized as follows. After a short preliminary, the two remaining parts of the paper address separately PS and FS. Since all our results use reductions from the same problem, we detail one proof to explain the nature of *self-improving reductions*, and give proof sketches for the remaining results.

2 Preliminary

Let LS be some learning sample. Each element of LS is an example consisting of an observation and a class. We suppose that the observations are described using a set V of n Boolean (0/1) variables, and there are only two classes, named "positive" (1) and "negative" (0) respectively. The basis for all our reductions is the minimization problem MIN-SET-COVER:

> NAME: MIN-SET-COVER.
> INSTANCE: a collection $C = \{c_1, c_2, ..., c_{|C|}\}$ of subsets of a finite set $S = \{s_1, s_2, ..., s_{|S|}\}$ ($|.|$ denotes the cardinality).
> SOLUTION: a set cover for S, *i.e.* a subset $C' \subseteq C$ such that every element of S belongs to at least one member of C.
> MEASURE: cardinality of the set cover, *i.e.* $|C'|$.

The central theorem which we use in all our results is the following one.

Theorem 1. *[ACG+99,CK00] Unless* $NP \subset DTIME[n^{\log\log n}]$*, the problem* MIN-SET-COVER *is not approximable to within* $(1-\epsilon)\log|S|$ *for any* $\epsilon > 0$.

By means of words, theorem 1 says that any (time) efficient algorithm shall not be able to break the logarithmic barrier $\log|S|$, that is, shall not beat significantly in the worst case the well-known *greedy set cover* approximation algorithm of [Joh74]. This algorithm guarantees to find a solution to any instance of MIN-SET-COVER whose cost, $|C'|$, is not larger than

$$\mathcal{O}(\log|S|) \times \mathrm{opt}_{\text{MIN-SET-COVER}},$$

where $\mathrm{opt}_{\text{MIN-SET-COVER}}$ is the minimal cost for this instance.
In order to state our results, we shall need particular complexity classes based on particular time requirement functions. We say that a function is *polylog*(n) if it is $\mathcal{O}(\log^c n)$ for some constant c, and quasi-polynomial, $QP(n)$, if it is $\mathcal{O}(n^{polylog(n)})$.

3 The Hardness of Approximating Prototype Selection

A simple and formal objective to prototype selection can be thought of as an information preserving problem as underlined in [BL97]. Fix some function $f : [0,1] \rightarrow [0,1]$ satisfying the following properties:

1. f is symmetric about $1/2$,
2. $f(1/2) = 1$ and $f(0) = f(1) = 0$,
3. f is concave.

Such functions are called *permissible* in [KM96]. Clearly, the binary entropy

$$H(x) = -x\log(x) - (1-x)\log(1-x),$$

the Gini criterion

$$G(x) = 4x(1-x)$$

[KM96] and the criterion

$$A(x) = 2\sqrt{x(1-x)}$$

used in [KM96,SS98] are all permissible. Define $p_1(LS)$ as the fraction of positive examples in LS, and $p_0(LS)$ as the fraction of negative examples in LS. Define $LS_{v=a}$ to be for some variable v the subset of LS in which all examples have value a ($\in \{0,1\}$) for v. Finally, define the quantity $I_f(v, LS)$ defined as

$$I_f(v, LS) = f(p_1(LS)) - \left(\frac{|LS_{v=1}|}{|LS|}f(p_1(LS_{v=1})) + \frac{|LS_{v=0}|}{|LS|}f(p_1(LS_{v=0}))\right)$$

This quantity, with f replaced by the functions $H(x), G(x)$ or $A(x)$, represents the common information measure to split the internal nodes of decision trees in all state-of-the-art decision tree learning algorithms (see for example [BFOS84,KM96,Mit97,Qui94,SS98]).

One objective in prototype selection can be to reduce the number of examples in LS while ensuring that any informative variable before will remain informative after the removal. The corresponding optimization problem, which we call MIN-PS$_f$ (for any f belonging to the category fixed above), is the following one:

> NAME: MIN-PS$_f$
> INSTANCE: a learning sample LS of examples described over a set of n variables $V = \{v_1, v_2, ..., v_n\}$.
> SOLUTION: a subset LS' of LS such that $\forall 1 \leq i \leq n, I_f(v_i, LS) > 0 \Rightarrow I_f(v_i, LS') > 0$.
> MEASURE: $|LS'|$.

There are two components in the self-improving reduction. The first one is to prove a basic inapproximability theorem. The second one, an *amplification lemma*, "blows-up" the result of the theorem. Then, we give some consequences illustrating the power of the amplification lemma.

Theorem 2. *Unless $NP \subset DTIME[n^{\log \log n}]$,* MIN-PS$_f$ *is not approximable to within $(1 - \epsilon) \log n$ for any $\epsilon > 0$.*

Proof. We show that MIN-PS$_f$ is as hard to approximate as MIN-SET-COVER: any solution to MIN-SET-COVER can be polynomially translated to a solution to MIN-PS$_f$ of the same cost, and reciprocally. Given an instance of MIN-SET-COVER, we build a set LS of $|C|$ positive examples and 1 negative example, each described over $|S|$ variables. We define a set $\{v_1, v_2, ..., v_{|S|}\}$ of Boolean variables, in one-to-one correspondence with the elements of S. The negative example is the all-0 example. Each positive example is denoted $e_1, e_2, ..., e_{|C|}$. We construct each positive example e_j so that it encodes the content of the corresponding set c_j of C. Namely, $e_j[k]$ is 1 iff $s_k \in c_j$, and 0 otherwise. Here we suppose obviously that each element of S is element of at least one element of C, which means that $\forall 1 \leq i \leq n, I_f(v_i, LS) > 0$. Suppose there exists a solution to MIN-SET-COVER of cost c. Then, we put in LS' the negative example, and all positive examples corresponding to the solution to MIN-SET-COVER. We see that for any variable v_j, there exists some positive example of LS' having 1 in its j^{th} component, since otherwise the solution to MIN-SET-COVER would not cover the elements of S. It is straightforward to check that $\forall 1 \leq i \leq n, I_f(v_i, LS') > 0$, which means that LS' is a solution to MIN-PS$_f$ having cost $c + 1$.

Now, suppose that there exists a feasible solution to MIN-PS$_f$, of size c. There must be the negative example inside LS' since otherwise we would have $\forall 1 \leq i \leq n, I_f(v_i, LS') = 0$. Consider all elements of C corresponding to the $c - 1$ positive examples of LS'. If some element s_i of S were not covered, the variable v_i would be assigned to zero over all examples of LS', be they positive or negative. In other words, we would have $I_f(v_i, LS') = 0$, which is impossible. In other words, we have build a solution of MIN-SET-COVER of cost $c - 1$.

If we denote $\text{opt}_{\text{MIN-SET-COVER}}$ and $\text{opt}_{\text{MIN-PS}}$ the optimal costs of the problems, we have immediately $\text{opt}_{\text{MIN-PS}} = \text{opt}_{\text{MIN-SET-COVER}} + 1$. A possible interpretation of theorem 1 is the following one [Aro94]: there exists some $\mathcal{O}(n^{\log \log n})$-time reduction from some NP-hard problem, say "SAT" for example, to MIN-SET-COVER, such that

- to any satisfiable instance of "SAT" corresponds a solution to MIN-SET-COVER whose cost is α,
- unsatisfiable instance of "SAT" are such that any feasible solution to MIN-SET-COVER will be of cost $> \alpha(1 - \epsilon) \log |S|$ for any $\epsilon > 0$.

This property is also called a *hard gap* in [Bel96].
If we consider the reduction from MIN-SET-COVER to MIN-PS$_f$, we see that the ratio between unsatisfiable and satisfiable instances of "SAT" is now

$$\rho = \frac{\alpha(1 - \epsilon) \log n + 1}{\alpha + 1}$$

For any $\epsilon' > 0$, if we choose $0 < \epsilon < \epsilon'$ (this is authorized by theorem 1), we have $\rho > (1 - \epsilon') \log n$ for MIN-PS$_f$, at least for sufficiently large instances of

"SAT". This concludes the proof of the theorem. □

The amplification lemma is based on the following self-improving reduction. Fix some integer value $d > 1$. Suppose we take again the instance of MIN-SET-COVER, but we create $|S|^d$ variables instead of the initial $|S|$. Each variable represents now a d-tuple of examples. Suppose we number the variables $v_{i_1,i_2,...,i_d}$ with $i_1, i_2, ..., i_d \in \{1, 2, ..., |S|\}$, to represent the corresponding examples. The $|C| + 1$ old examples are replaced by $|C|^d + 1$ examples described over these variables, as follows:

- for any possible d-tuple $(c_{j_1}, c_{j_2}, ..., c_{j_d})$ of elements of C, we create a positive example $e_{j_1,j_2,...,j_d}$, having ones in variable $v_{i_1,i_2,...,i_d}$ iff

$$\forall k \in \{1, 2, ..., d\}, s_{i_k} \in c_{j_k},$$

  and zeroes everywhere else. Thus, the Hamming weight of the example's description is exactly $\prod_{k=1}^{d} |c_{j_k}|$. By this procedure, we create $|C|^d$ positive examples,
- we add the all-zero example, having negative class.

We call LS_d this new set of examples. Note that the time made for the reduction is no more than $\mathcal{O}(|S|^d|C|^d)$. The following lemma exhibits that the inapproximability ratio for MIN-PS$_f$ actually grows as a particular function of d provided d is confined to reasonable values, in order to keep an overall reduction time not greater than $\mathcal{O}(n^{\log \log n})$. Informally, this assumption allows to use the inapproximability ratio of theorem 1 for our reduction. For the sake of simplicity in stating the lemma, we say that the reduction is *feasible* to state that this assumption holds.

Lemma 1. *Unless $NP \subset DTIME[n^{\log \log n}]$, provided the reduction is feasible, then* MIN-PS$_f$ *is not approximable to within*

$$\left(\frac{(1-\epsilon)\log n}{d}\right)^d$$

for any $\epsilon > 0$.

Proof. Again, we suppose obviously that each element of S is element of at least one element of C, which means that each variable $v_{i_1,i_2,...,i_d}$ has

$$I_f(v_{i_1,i_2,...,i_d}, LS_d) > 0$$

Note that any feasible solution to MIN-PS$_f$ contains the negative example (same reason as for theorem 2). Also, in any solution $C' = \{c'_1, c'_2, ..., c'_{|C'|}\}$ to MIN-SET-COVER, the following property **P** is satisfied without loss of generality: any element of C belonging to it has at least one element (of S) which is present in no other element of C', since otherwise the solution could be transformed in polynomial time into a solution of lower cost (simply remove arbitrarily elements in C' to satisfy **P** while keeping a cover of S). As **P** is satisfied, we call any subset of cardinality $|C'|$ of S containing one such distinguished element for each element of C' a *distinguished* subset of S. Finally, remark that MIN-PS$_f$ is equivalent to the problem of covering the set S^d using elements of C^d, and the minimal number of positive examples in LS_d is exactly the minimal cost c' of the instance of this generalization of MIN-SET-COVER. But, since **P** holds, covering C^d requires to cover any d-tuple of distinguished subsets of S and because property **P** holds, c' is at least c^d where c is the optimal cost of the instance of MIN-SET-COVER. Also, if we take all d-tuples of elements of C' feasible solution to MIN-SET-COVER, we get a feasible solution to the generalization of MIN-SET-COVER, which leads to the equality $c' = c^d$.

If we denote $\text{opt}_{\text{MIN-PS}}$ the optimal cost of MIN-PS$_f$ on the new set of examples LS_d, we obtain that

$$\text{opt}_{\text{MIN-PS}} = (\text{opt}_{\text{MIN-SET-COVER}})^d + 1$$

Given that $n = |S|^d$, and using the same ideas as for theorem 2, we obtain the statement of the lemma. $\square$

What can we hope to gain by using lemma 1, which was not already proven by theorem 2 ? It is easy to show that the largest inapproximability ratio authorized by the same complexity assumption is

$$\rho = \log^{\log\left(\frac{\log n^{1-\epsilon}}{\log\log n}\right)} n \tag{1}$$

(by taking $d = \mathcal{O}(\log\log n)$), which implies the simpler one:

Theorem 3. *Unless $NP \subset DTIME[n^{\log\log n}]$, MIN-PS$_f$ is not approximable to within*

$$\log^{(1-\epsilon)\log\log n} n$$

for any $\epsilon > 0$.

Another widely encountered complexity hypothesis, stronger than the one of theorem 3, is that $NP \not\subset QP$ [CK00]. In that case, the result of theorem 3 becomes stronger:

Theorem 4. *Unless $NP \subset QP$, $\exists\delta > 0$ such that MIN-PS$_f$ is not approximable to within n^δ.*

Proof. We prove the result for $\delta < 1/e$, and take $d = (1-\delta)\log n$. A good choice of ϵ in theorem 2 proves the result. $\square$

The preceeding model takes into account the information of the variables to select relevant prototypes. We now give a model for prototype selection based on the notion of relevance with respect to a concept. For any set of examples LS, denote as $\mathcal{C}_{opt}(LS)$ the set of concept representations having minimal size, and consistent with LS. The notion of size can be *e.g.* the overall number of variables of the concept (if a variable appears i times, it is counted i times). The nature of the concepts is not really important: these could be decision trees, decision lists, disjunctive normal form formulas, linear separators, as well as simple clauses. Our negative results will force the concepts of $\mathcal{C}_{opt}(LS)$ to belong to a particularly simple subclass, expressible in each class. This notion of relevance is closely related to a particular kind of ML algorithms in which we seek consistent formulas with limited size: Occam's razors [KV94,NJS98]. Formulated as an optimization problem, the MIN-PS problem is the following one:

NAME: MIN-PS.
INSTANCE: a learning sample LS of examples described over a set of variables $\{v_1, v_2, ..., v_n\}$.
SOLUTION: a subset LS' of LS such that $\mathcal{C}_{opt}(LS') \subseteq \mathcal{C}_{opt}(LS)$.
MEASURE: $|LS'|$.

By means of words, PS is a problem of reducing the number of examples while ensuring that concepts consistent and minimal with respect to the subset of prototypes will also be valid for the whole set of examples. Our first result on the inapproximability of this new version of MIN-PS is the following one.

Theorem 5. *Unless $NP \subset DTIME[n^{\log\log n}]$, MIN-PS is not approximable to within $(1-\epsilon)\log n$ for any $\epsilon > 0$.*

Proof. (*sketch*) The proof resembles the one of theorem 2. Given an instance of MIN-SET-COVER, we build a set LS of $|S|$ positive examples and 1 negative example, each described over $|C|$ variables. We define a set $\{v_1, v_2, ..., v_{|C|}\}$ of Boolean variables, in one-to-one correspondence with the elements of C. The negative example is the all-0 example. Each positive example is denoted $e_1, e_2, ..., e_{|S|}$. We construct each positive example e_j so that it encodes the membership of s_j into each element of C. Namely, $e_j[k]$ is 1 iff $s_j \in c_k$, and 0 otherwise. Similarly to theorem 2, the least number of examples which can be kept is exactly the cost of the optimal solution to MIN-SET-COVER, plus one.

The proof is similar to that of theorem 2, with the following remark on the minimal concepts. It can be shown that minimal concepts belonging to each of the classes cited before (trees, lists, etc.) will contain a number of variables equal to the minimal solution to MIN-SET-COVER, and each will be present only once. The reduction is indeed very generic and similar results were previously obtained by e.g. [NG95] (for linear separators and even multilinear polynomials), [NJ98b] (for decision lists), [HR76,HJLT94] (for decision trees), [Noc98] (for Disjunctive Normal Form formulas and simple clauses). From that, all minimal concepts will be equivalent to a simple clause whose variables correspond to C'. Property **P** in lemma 1 can still be used. □

The amplification lemma follows from a particular self-improving reduction. Again, fix some integer value $d > 1$. Suppose we take again the instance of MIN-SET-COVER, but we create $d|C|$ variables instead of the initial $|C|$. Each variable is written $v_{i,j}$ to denote the j^{th} copy of initial variable i, with $i = 1, 2, ..., |C|$ and $j = 1, 2, ..., d$. The $|S| + 1$ old examples are replaced by $|S|^d + 1$ examples described over these variables, as follows:

- for any possible d-tuple $(s_{j_1}, s_{j_2}, ..., s_{j_d})$ of elements of S, we create a positive example $e_{j_1,j_2,...,j_d}$, having ones in variable $v_{k,l}$ iff $s_{j_l} \in c_k$, and zeroes everywhere else. By this procedure, we create $|C|^d$ positive examples,
- we add the all-zero example, having negative class.

We call LS_d this new set of examples. Note that the time made for the reduction is no more than $\mathcal{O}(|S|^d|C|^d)$. The following lemma is again stated under the hypothesis that the reduction is *feasible*, that is, takes no more time than $\mathcal{O}(n^{\log\log n})$, to keep the same complexity assumption as in theorem 1 (proof omitted).

Lemma 2. *Unless $NP \subset DTIME[n^{\log\log n}]$, provided the reduction is feasible, then* MIN-PS *is not approximable to within*

$$\left((1-\epsilon)\log\left[\frac{n}{d}\right]\right)^d$$

for any $\epsilon > 0$.

What can we hope to gain by using lemma 2, which was not already proven by theorem 5 ? It is easy to show that the largest inapproximability ratio authorized by the same complexity assumption is now

$$\rho = \log^{\log\log\left(\frac{n}{\log\log n}\right)^{1-\epsilon}} n \quad (2)$$

which in turn implies the following one (greater than eq. 1):

Theorem 6. *Unless $NP \subset DTIME[n^{\log\log n}]$,* MIN-PS *is not approximable to within*

$$\log^{\log\log(n^{1-\epsilon})} n$$

for any $\epsilon > 0$.

With a slightly stronger hypothesis (and using $d = \mathcal{O}(polylog(n))$), we obtain

Theorem 7. *Unless $NP \subset QP$, $\forall c > 0$,* MIN-PS *is not approximable to within* $n^{\log^c n \log\log\log n}$.

With respect to 1, lemma 2 brings results much more negative provided stronger complexity assumptions are made. [PR94] make the very strong complexity assumption $NP \not\subset DTIME(2^{n^{\Omega(1)}})$. This is the strongest complexity assumption, since NP is definitely contained in $DTIME(2^{poly(n)})$. Using this hypothesis with $d = n^{\Omega(1)}$, we obtain the following, very strong result:

Theorem 8. *Unless $NP \subset DTIME(2^{n^{\Omega(1)}})$, $\exists\gamma > 0$ such that* MIN-PS *is not approximable to within*

$$2^{n^{\gamma}\log\log n}$$

What theorem 8 says is that approximating prototype selection up to exponential ratios

$$2^{n^{\gamma+o(1)}}$$

will be hard. Note that storing the examples would require 2^n examples in the worst case. Up to what is precisely hidden in the γ notation, approximating MIN-PS might not be efficient at all with respect to the storing of all examples.

4 The Hardness of Approximating Feature Selection

The first model of feature selection is related to the distribution of the examples in LS. Let V_i be the set of all variables except v_i, *i.e.*

$$V_i = \{v_1, v_2, ..., v_{i-1}, v_{i+1}, ..., v_n\}$$

Denote by $v_{\backslash i}$ a value assignment to all variables in V_i.

Definition 1. *[JKP94] A variable v_i is* ***strongly*** *relevant iff there exists some v, y and $v_{\backslash i}$ for which $\mathbf{Pr}(v_i = v, V_i = v_{\backslash i}) > 0$ such that*

$$\mathbf{Pr}(Y = y | v_i = v, V_i = v_{\backslash i}) \neq \mathbf{Pr}(Y = y | V_i = v_{\backslash i})$$

Definition 2. *[JKP94] A variable v_i is* ***weakly*** *relevant iff it is not strongly relevant, and there exists a subset of features V_i' of V_i for which there exists some v, y and $v'_{\backslash i}$ with $\mathbf{Pr}(v_i = v, V_i' = v'_{\backslash i}) > 0$ such that*

$$\mathbf{Pr}(Y = y | v_i = v, V_i' = v'_{\backslash i}) \neq \mathbf{Pr}(Y = y | V_i' = v'_{\backslash i})$$

In other words, a feature is weakly relevant if it becomes strongly relevant after having deleted some subset of features. We now show that under these two definitions are hidden algorithmic problems of very different complexities. We formulate the selection of relevant features as an optimization problem by focusing on the class conditional probabilities, following the definition of *coherency* which we give below:

Definition 3. *Given a whole set V of features with which LS is described, a subset V' of V is said to be* **coherent** *iff for any class y and any observation s described with V whose restriction to V' is noted s', we have*

$$\mathbf{Pr}(Y = y | V = s) = \mathbf{Pr}(Y = y | V' = s')$$

By means of words, coherency aims at keeping the class conditional probabilities between the whole set of variables and the selected subset. Formulated as an optimization problem, the MIN-S-FS problem is the following one:

- NAME: MIN-S-FS.
- INSTANCE: a learning sample LS of examples described over a set of variables $V = \{v_1, v_2, ..., v_n\}$.

- SOLUTION: a coherent subset V' of V containing strongly relevant features w.r.t. LS.
- MEASURE: $|V'|$.

The MIN-W-FS problem is the following one:

- NAME: MIN-W-FS.
- INSTANCE: a learning sample LS of examples described over a set of variables $V = \{v_1, v_2, ..., v_n\}$.
- SOLUTION: a coherent subset V' of V containing weakly relevant features w.r.t. LS.
- MEASURE: $|V'|$.

Since strong relevance for a variable is not influenced by its peers, we easily obtain the following theorem

Theorem 9. *Minimizing* MIN-S-FS *is polynomial.*

We now show that MIN-W-FS is much more difficult to approximate.

Theorem 10. *Unless* $NP \subset DTIME[n^{\log \log n}]$, MIN-W-FS *is not approximable to within* $(1 - \epsilon) \log n$ *for any* $\epsilon > 0$.

Proof. The reduction is the same as for theorem 5. □

The result of theorem 10 shows that MIN-W-FS is hard, but it does not rule out the possibility of efficient feature selection algorithms, since the ratio of inapproximability is quite far from critical bounds of order n^γ (given that the number of features is n). We now show that theorem 10 is also subject to be amplified so that we can effectively remove the possibility of efficient feature selection. Fix some integer value $d > 1$. Suppose we take again the instance of MIN-SET-COVER of theorem 5, but we create $|C|^d$ variables instead of the initial $|C|$. Each variable represents now a d-tuple of elements of C. Suppose we number the variables $v_{i_1,i_2,...,i_d}$ with $i_1, i_2, ..., i_d \in \{1, 2, ..., |C|\}$, to represent the corresponding elements of C. The $|S| + 1$ old examples are replaced by $|S|^d + 1$ examples described over these variables, as follows:

- for any possible d-tuple $(s_{j_1}, s_{j_2}, ..., s_{j_d})$ of elements of S, we create a positive example $e_{j_1,j_2,...,j_d}$, having ones in variable $v_{i_1,i_2,...,i_d}$ iff

$$\forall k \in \{1, 2, ..., d\}, s_{j_k} \in c_{j_k},$$

  and zeroes everywhere else. By this procedure, we create $|S|^d$ positive examples,
- we add the all-zero example, having negative class.

We call LS_d this new set of examples. The reduction time is no more than $\mathcal{O}(|S|^d|C|^d)$. The following lemma is stated under the same hypothesis as for lemma 2.

Lemma 3. *Unless $NP \subset DTIME[n^{\log \log n}]$, provided the reduction is feasible,* MIN-W-FS *is not approximable to within*

$$\left(\frac{(1-\epsilon)\log n}{d}\right)^d$$

for any $\epsilon > 0$.

An immediate consequence is the following.

Theorem 11. *Unless $NP \subset QP$, $\exists \delta > 0$ such that* MIN-W-FS *is not approximable to within n^{δ}.*

In other words, up to what is be the maximal δ, theorem 11 shows that any non trivial algorithm cannot achieve a significant worst-case approximation of the MIN-W-FS problem, with respect to the simple keeping of all variables.

Our second model for feature relevance defines it with respect to the target concept [BL97].

Definition 4. *[BL97] A variable v_i is said to be relevant to the target concept c iff there exists a pair of examples e_A and e_B in the instance space such that their observations differ only in their assignment to v_i and they have a different class.*

From this, [BL97] define the following complexity measure.

Definition 5. *[BL97] Given a sample LS and a set of concept $\mathcal{C}$, $r(LS,\mathcal{C})$ is the number of features relevant using definition 4 to a concept in C that, out of all those whose error over LS is least, has the fewest relevant features.*

We call $\mathcal{C}_{min}(LS)$ to be the set of concepts from $\mathcal{C}$ whose error on LS is least. It is straightforward to check that in definition 5, $r(LS,\mathcal{C})$ defines the optimum of the following minimization problem.

NAME: MIN-FS.
INSTANCE: a learning sample LS of examples described over a set of variables $V = \{v_1, v_2, ..., v_n\}$, a class of concept $\mathcal{C}$.
SOLUTION: a subset V' of V such that there exists a concept in $\mathcal{C}_{min}(LS)$ which is described over V'.
MEASURE: the cardinality of the subset of V' consisting of relevant features according to definition 4.

A result stated in the paper of [BL97] says that MIN-FS is at least as hard to approximate as the MIN-SET-COVER problem (thus, we get the inapproximability ratio of theorem 1). On the other hand, the greedy set cover algorithm of [Joh74] can be used to approximate $r(LS,\mathcal{C})$ when $\mathcal{C}$ is chosen to be the set of monomials. If we follow [KV94] using a comment of [BL97], the number of variables chosen is no more than

$$r(LS, \text{monomials}) \times \log |LS|,$$

but $|LS|$ can theoretically be as large as 2^n. The question is therefore to what extent we can increase the inapproximability ratio to come as close as possible to the trivial barrier n (we keep all variables). Actually, it can easily be shown that the amplification result of lemma 1 still holds with the reduction allowing to prove the equivalence of MIN-SET-COVER and MIN-FS. Therefore, we get

Lemma 4. *Unless $NP \subset DTIME[n^{\log\log n}]$, provided the reduction is feasible, then* MIN-FS *is not approximable to within*

$$\left(\frac{(1-\epsilon)\log n}{d}\right)^d$$

for any $\epsilon > 0$.

Similarly to theorem 4, we also get as a consequence:

Theorem 12. *Unless $NP \subset QP$, $\exists\delta > 0$ such that* MIN-FS *is not approximable to within n^δ.*

References

[ACG+99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, Marchetti Spaccamela A., and Protasi M. *Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties.* Springer-Verlag, Berlin, 1999.

[Aro94] S. Arora. Probabilistic checking of proofs and hardness of approximation problems. Technical Report CS-TR-476-94, Princeton University, 1994.

[Bel96] M. Bellare. Proof checking and Approximation: towards tight results. *SIGACT news*, 1996.

[BFOS84] L. Breiman, J. H. Freidman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.

[BL97] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–272, 1997.

[Blu94] A. Blum. Relevant examples and relevant features: Thoughts from computational learning theory. In *AAAI Fall Symposium (survey paper)*, 1994.

[CK00] P. Crescenzi and V. Kann. *A Compendium of NP-Optimization problems.* WWW-Available at `http://www.nada.kth.se/~viggo/wwwcompendium/`, 2000.

[HJLT94] T. Hancock, T. Jiang, M. Li, and J. Tromp. Lower bounds on learning decision lists and trees. In *Proc. of the Symposium on Theoretical Aspects of Computer Science*, 1994.

[HR76] L. Hyafil and R. Rivest. Constructing optimal decision trees is np-complete. *Inform. Process. Letters*, pages 15–17, 1976.

[JKP94] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proc. of the 11 th International Conference on Machine Learning*, pages 121–129, 1994.

[Joh74] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sci.*, pages 256–278, 1974.

[KKLP97] V. Kann, S. Khanna, J. Lagergren, and A. Panconesi. On the hardness of approximating MAX k-CUT and its dual. *Chicago Journal of Theoretical Computer Science*, 2, 1997.

[KM96] M.J. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 459–468, 1996.

[Koh94] R. Kohavi. Feature subset selection as search with probabilistic estimates. In *AAAI Fall Symposium on Relevance*, 1994.

[KS95] R. Kohavi and D. Sommerfield. Feature subset selection using the wrapper model: overfitting and dynamic search space topology. In *First International Conference on Knowledge Discovery and Data Mining*, 1995.

[KS96] D. Koller and R. M. Sahami. Toward optimal feature selection. In *Proc. of the 13 th International Conference on Machine Learning*, 1996.

[KV94] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory.* M.I.T. Press, 1994.

[Mit97] T. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[NG95] R. Nock and O. Gascuel. On learning decision committees. In *Proc. of the 12 th International Conference on Machine Learning*, pages 413–420, 1995.

[NJ98a] R. Nock and P. Jappy. Function-free horn clauses are hard to approximate. In *Proc. of the Eighth International Conference on Inductive Logic Programming*, pages 195–204, 1998.

[NJ98b] R. Nock and P. Jappy. On the power of decision lists. In *Proc. of the 15 th International Conference on Machine Learning*, pages 413–420, 1998.

[NJS98] R. Nock, P. Jappy, and J. Sallantin. Generalized Graph Colorability and Compressibility of Boolean Formulae. In *Proc. of the 9 th International Symp. on Algorithms and Computation*, pages 237–246, 1998.

[Noc98] R. Nock. *Learning logical formulae having limited size : theoretical aspects, methods and results.* PhD thesis, Université Montpellier II, 1998. Also available as techreport RR-LIRMM-98014.

[PR94] K. Pillaipakkamnatt and V. Raghavan. On the limits of proper learnability of subclasses of DNF formulae. In *Proc. of the 7 th International Conference on Computational Learning Theory*, pages 118–129, 1994.

[Qui94] J. R. Quinlan. *C4.5 : programs for machine learning.* Morgan Kaufmann, 1994.

[Ska94] D. B. Skalak. Prototype and feature selection by sampling and random mutation hill-climbing algorithms. In *Eleventh International Conference on Machine Learning*, pages 293–301, 1994.

[SN00a] M. Sebban and R. Nock. Combining feature and prototype pruning by uncertainty minimization. In *Proc. of the 16 th International Conference on Uncertainty in Artificial Intelligence*, 2000. to appear.

[SN00b] M. Sebban and R. Nock. Prototype selection as an information-preserving problem. In *Proc. of the 17 th International Conference on Machine Learning*, 2000. to appear.

[SS98] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Annual ACM Conference on Computational Learning Theory*, pages 80–91, 1998.

[WM97] D. Wilson and T. Martinez. Instance pruning techniques. In *Proc. of the 14 th International Conference on Machine Learning*, pages 404–411, 1997.