Generalized Graph Colorability and Compressibility of Boolean Formulae

Richard Nock, Pascal Jappy, and Jean Sallantin

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

Abstract. In this paper, we study the possibility 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. 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 worst-case lower bound on the formulae's sizes found by learning systems proceeding by rule searching.

1 Introduction

The learnability of Disjunctive Normal Form formulae (disjunctions of conjunctions) is a central problem in machine learning [27]. 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*" [35]. 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 [19,27,2,1,4,5,8,15,20,23,28].

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 [10], 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 [29]. Many learning algorithms either build

K.-Y. Chwa and O. H. Ibarra (Eds.): ISAAC'98, LNCS 1533, pp. 237–246, 1998.

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, [7,10,9,12,21,24,25,26,31,30,29,32,33,36] and many others.

Practical and theoretical results often focus on an aspect of approximation they both share : given a set of examples (*e.g.* description of animals for which we know if they have or not some illness), can we devise an efficient algorithm which can find a small formula (*i.e.* a mean of classifying observations) consistent with all examples (making no errors)?
Theoretically speaking, this aspect is often related to the principle of Occam's razors [6]: an Occam's razor for a class (*i.e.* a set) of concept representations C (each of which is a function mapping observations to classes) 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 (it does not make errors) and has size satisfying $|h| \leq |LS|^a(n|t|)^b$ (with $0 \leq a < 1$ and $b \geq 0$). The principle of Occam's razors [18] states that in order to learn, a system should compress the information contained in the examples. This principle was originally stated by philosoph William of Occam (1285-1349), and led to theoretical results in the PAC-learning model of Valiant [34] : learning is in fact equivalent to finding Occam's razors [18].
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 [17].

The principal result on the inapproximability of DNF comes from [19]. They show that DNF is as hard to approximate as a proglem related to a generalization of Graph Colorability (which does not have commonpoints with ours). [13] prove that Graph Colorability is hard to approximate to within n^δ ($\forall 0 < \delta < 1$). Using the reduction of [19], we are able to show that the consistent DNF with minimal size is not approximable (in size) 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 [3].

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 [19] that this problem is as hard as the problem *Opt* (Independant-set cover) (this is the same as the graph-coloring problem [14]; 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 (proof omitted due to space limitations).

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.*

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*

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

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 (graphs instance of *Opt*(Independant-set cover)) 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

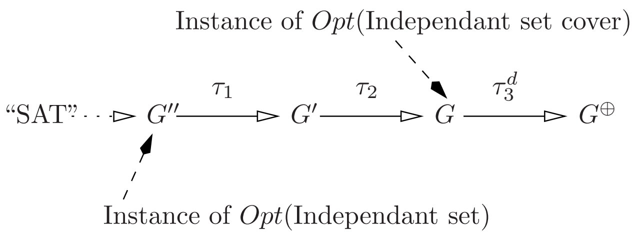

Fig. 1. Scheme of the reduction $\tau_1 \circ \tau_2 \circ \tau_3^d$.

need in this part is the following:

Theorem 2 *(From [16]) Unless $NP \subseteq ZPP$, $\forall 0 \leq \delta < 1$, 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 [13,16], 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.

Definition 2 τ_1 **[22]** *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 [22], Section 2.1, we have $\alpha(G'') = \alpha(G')$. Let Z_p be the field of integers modulo p.

Definition 3 τ_2 **[22]** *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*

$$\max\{\max_i |C_i|, r/\alpha(G'), \sqrt{r}\}$$

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

Proposition 2 *G satisfies:*

1. $n(G) = rp^5$ *([22], part 2.1)*
2. $\alpha(G) = p^2\alpha(G')$ *([22], corollary 2.2)*
3. *If G is built from a satisfiable instance of "SAT", then $\chi(G) = n(G)/\alpha(G)$ ([22], theorem 2.3)*
4. $p^2\chi(G) \leq n(G)$

(proof of [4] omitted due to space limitations). 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, [22]) 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

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

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

We now check the constant value for d satisfying inequation 4. [22] choose for p a prime at least as large as $\max\left\{\max_i |C_i|, \frac{r}{\alpha(G'')}, \sqrt{r}\right\}$, which is $n(G'')$ in our case. We also have

Proposition 3 [11] *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

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

with

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

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

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

implies whenever $c \geq 1$

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

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 proposition 1, from the fact that $|LS| \leq (\max_{1\leq i\leq n}\{|E_i| + |V_i|\})^d$, $n \leq d \max_{1\leq i\leq n}\{|V_i|\}$, 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 reasonable 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. [27] cite the complexity hypothesis $NP \not\subset \mathrm{DTIME}(poly(2^{n^\epsilon}))$ (for some $\epsilon > 0$). With such hypotheses, adaptated to handle zero-error, probabilistic algorithms, it would be interesting to see how close to the Occam requirements the non-approximability ratio would come.

References

1. 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 Learning Theory*, pages 71–76, 1992. 237
2. H. Aizenstein and L. Pitt. On the learnability of Disjunctive Normal Form formulas. *Machine Learning*, 19:183–208, 1995. 237
3. J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity I.* Springer Verlag, 1988. 238
4. U. Berggren. Linear time deterministic learning of k-term-DNF. In *Proc. of the 6 th International Conference on Computational Learning Theory*, pages 37–40, 1993. 237
5. 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 Learning Theory*, pages 110–117, 1994. 237
6. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, pages 377–380, 1987. 238
7. C. Brunk and M. Pazzani. Noise-tolerant relational concept learning. In *Proc. of the 8 th International Conference on Machine Learning*, 1991. 238
8. 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 Learning Theory*, pages 198–205, 1995. 237
9. W. W. Cohen. PAC-learning a restricted class of recursive logic programs. In *Proc. of AAAI-93*, pages 86–92, 1993. 238
10. W. W. Cohen. Fast effective rule induction. In *Proc. of the 12 th International Conference on Machine Learning*, pages 115–123, 1995. 237, 238
11. C. de la Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning*, pages 1–14, 1997. 243
12. L. de Raedt. Iterative concept learning and construction by analogy. *Machine Learning*, pages 107–150, 1992. 238
13. U. Feige and J. Kilian. Zero knowledge and the chromatic number. draft, 1996. 238, 241
14. M.R. Garey and D.S. Johnson. *Computers and Intractability, a guide to the theory of NP-Completeness*. Bell Telephone Laboratories, 1979. 239
15. 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 Learning Theory*, pages 77–84, 1992. 237
16. J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. In *FOCS'96*, pages 627–636, 1996. 241

17. R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, pages 63–91, 1993. 238
18. M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. M.I.T. Press, 1994. 238
19. 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. 237, 238, 239
20. R. Khardon. On using the fourier transform to learn disjoint DNF. *Information Processing Letters*, pages 219–222, 1994. 237
21. N. Lavrac, S. Dzeroski, and M. Grobelnik. Learning non-recursive definitions of relations with linus. In *European Working Session in Learning*, 1991. 238
22. 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. 241, 242, 243
23. 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 Learning Theory*, pages 53–61, 1992. 237
24. S. Muggleton and C. Feng. Efficient induction of logic programs. In *Inductive Logic Programming*, 1994. 238
25. R. Nock and O. Gascuel. On learning decision committees. In *Proc. of the 12 th International Conference on Machine Learning*, pages 413–420, 1995. 238
26. J. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 1990. 238
27. 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. 237, 245
28. L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *J. ACM*, pages 965–984, 1988. 237
29. J. R. Quinlan. Learning logical definition from relations. *Machine Learning*, pages 239–270, 1990. 237, 238
30. J. R. Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann, 1994. 238
31. J. R. Quinlan. MDL and categorical theories (continued). In *Proc. of the 12 th International Conference on Machine Learning*, pages 464–470, 1995. 238
32. C. Rouveirol. ITOU: induction of first-order theories. *Inductive Logic Programming*, 1992. 238
33. 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. 238
34. L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984. 238
35. L. G. Valiant. Learning disjunctions of conjunctions. In *Proc. of the 9 th IJCAI*, pages 560–566, 1985. 237
36. J Wnek and R. Michalski. Hypothesis-driven constructive induction in AQ17. In *Proc. of the 12 th IJCAI*, 1991. 238