

Learning rule representations from boolean data ^{*}

B. Apolloni¹, A. Brega¹, D. Malchiodi¹, G. Palmas², and A. M. Zanaboni¹

¹ Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
Via Comelico 39/41, 20135 Milano, Italy

{apolloni,malchiodi,zanaboni}@dsi.unimi.it, andrea@laren.dsi.unimi.it

² ST Microelectronics s.r.l., Agrate Brianza (Milano)

giorgio.palmas@st.com

Abstract. We discuss a Probably Approximate Correct (PAC) learning paradigm for Boolean formulas, which we call PAC meditation, where the class of formulas to be learnt is not known in advance. We split the building of the hypothesis in various levels of increasing description complexity according to additional inductive biases received at run time. In order to give semantic value to the learnt formulas, the key operational aspect represented is the understandability of formulas, which requires their simplification at any level of description. We deepen this aspect in light of two alternative simplification methods, which we compare through a case study.

1 Introduction

PAC learning [1] is a very efficient approach for selecting a function within a class of Boolean functions (call them concepts) on the basis of a set of examples of how this function computes. In this paper we will consider an extension of this approach to the case that the class of concepts is not known at the beginning of the learning procedure. Rather we receive requisites of the class a little at a time in subsequent steps of the learning process. Thus at runtime we must be sure of both correctly updating current knowledge on the basis of new requisites and suitably reinterpreting examples in the light of this knowledge, so that the approximation of the hypothesis on the final concept and its readability are not compromised. We learn Boolean formulas through a multi-level procedure that we call *PAC-meditation*:

- at the ground level we have two sets of positive (label 1) and negative (label 0) examples. From subsets of examples with a same label we compute partial consistent hypotheses. Namely each hypothesis is consistent with (i.e. computes the correct labels of) a part of the positive examples and all negative examples, or *vice versa*. The union of the former constitutes an *inner border*

^{*} Work partially funded by E.C. contract No. IST-2000-26091, “ORESTEIA: mOdular hyBRid artEfactS wiTh adaptivE functIonAlity”.

and the intersection of the latter an *outer border*. The two nested regions delimit a gap where we look for consistent hypotheses. In Fig. 1 the gap is represented by the dashed area.

- at further abstraction levels the partial consistent hypotheses of the preceding level play the role of labelled examples, where the sampled data are substituted by formulas and the positive and negative labels are substituted by a flag which denotes whether these formulas belong to the inner or outer border. We construct a new pair of borders running the same procedure on the so represented examples (these borders are contoured by bold lines in Fig. 1), with the double benefit of both reducing the gap between the border and increasing their understandability through the introduction of new symbols.

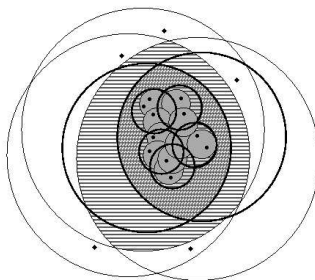


Fig. 1. Inner and outer borders, in the sample space, of a concept at two abstraction levels. Inner borders are delimited by the union of formulas bounded by positive examples (gray circles with thin contour at ground level), outer borders by the intersection of formulas bounded by negative examples (white circles with thin contour at ground level). Bold lines describe higher level formulas. Bullets: positive examples; diamonds: negative examples.

The idea of fitting the formula within minimum and maximum hypotheses is well known in the literature, and acquainted from various perspectives [2] [3]. The formula simplification is performed according to an optimality criterion. We propose one entropic method and an alternative one based on fuzzy relaxation; numerical case studies show that the latter one proves faster achieving almost the same accuracy as the former. The paper is organized as follows. In section 2 we describe the learning procedure, in section 3 its performance is discussed, and the last section is devoted to concluding remarks.

2 The inference process

Let us denote by $X_n \equiv \{0, 1\}^n$ the space of boolean vectors \mathbf{x} of size n that can be assigned to the set of propositional variables from $V_n = \{v_1, \dots, v_n\}$, and

by g^* the target of our learning procedure. The block diagram of the inference process is shown in Fig. 2. Given a set of positive and negative examples the procedure core consists of the iterated activation of an *abstraction module* made up of two steps: i) a *Symbols' jump*, where we introduce new symbols to describe (Boolean) properties on the points by the search for an inner and an outer border; and ii) a *Reduction* step, which is devoted to broadening or narrowing these hypotheses with: i) the constraint of not violating consistency; and ii) the aim of narrowing the gap between borders through a simplified version of them. At each abstraction level we may restart the two steps after assuming the minimal hypotheses as positive *metapoints* at the previous level, maximal hypotheses as negative *metapoints*, and searching for new *metasymbols* to describe properties on these points. To avoid tautologies, the new abstraction level must be enriched by pieces of symbolic knowledge that are now available about the properties we want to discover and that translate into additional constraints in rebuilding the borders. Once we are satisfied with the achieved abstraction level (or simply do not plan attaining new formal knowledge), the *Level test* in Fig. 2 addresses us to the *Synthesis step*. Here we collapse the two borders into a single definite formula lying between them which we assume as representative of the properties on the random population we observed.

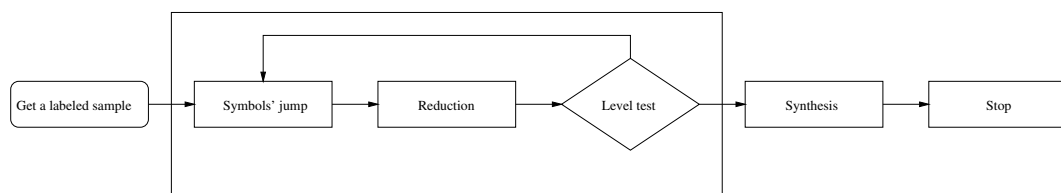


Fig. 2. Block diagram of PAC-meditation.

2.1 Symbol's jump

Ground level. Starting from \mathbf{X}_n , we construct the atomic components of 0-level borders, which we call *canonical monomials* and *clauses*. They are respectively product and sum of propositional variables from V_n . Denoting $\text{set}(\mathbf{g})$ the set of literals used by the formula \mathbf{g} , we describe the canonical formulas as follows.

Definition 1. *i) given \mathbf{X}_n and a set E^+ of positive examples, a monotone monomial \mathbf{m} with arguments in V_n is a canonical monomial if there exists $\mathbf{x} \in E^+$ such that for each $i \in \{1, \dots, n\}$, $v_i \in \text{set}(\mathbf{m})$ if $x_i = 1$, and $v_i \notin \text{set}(\mathbf{m})$ otherwise. ii) given \mathbf{X}_n and a set E^- of negative examples, a monotone clause \mathbf{c} with arguments in V_n is a canonical clause if there exists $\mathbf{x} \in E^-$ such that for each $i \in \{1, \dots, n\}$, $v_i \in \text{set}(\mathbf{c})$ if $x_i = 0$ and $v_i \notin \text{set}(\mathbf{c})$ otherwise.*

These formulas do not constrain the final expression of the target g^* of the learning procedure (that we assume to be a monotone formula). Infact, any Boolean formula on the binary hypercube can be represented either through the union of monomials (DNF) or through the intersection of clauses (CNF). They just represent a set of points that necessarily must belong to g^* given a positive example, or cannot belong to it given a negative one.

First abstraction level. The jump to the first abstraction level is done as follows. Applying the distributive property to the union of two monomials

$$v_i v_j v_k \vee v_l v_m v_n v_r = (v_i \vee v_l v_m) \wedge (v_i \vee v_n v_r) \wedge (v_j v_k \vee v_l v_m) \wedge (v_j v_k \vee v_n v_r) \quad (1)$$

we obtain a new monomial where literals are constituted by clauses and, in turn, literals in the clauses are substituted by monomials. We generalize the operation, keeping groups of at most k_2 monomials and splitting them in such a way that at most k_1 metaclauses arise. As a consequence each metaclause is the union of k_2 monomials. Bounds on k 's stand for requisites of conciseness on the formula description, i.e. for a compression of our knowledge that constitutes the true *inductive bias* [4] of our procedure. We do the same with metaclauses.

Climbing abstraction levels. Denoting $\cup^t = \cap^{t-1} = \cap$ if t is odd and $\neq 0$, \cup otherwise, for $t \geq 0$, at the L^{th} abstraction level we obtain formulas belonging to the families of meta- L -monomials $\mathbf{G}_{n;k_0,k_1,\dots,k_{\nu-1}}$ and meta- L -clauses $\mathcal{G}_{n;k_0,k_1,\dots,k_{\nu-1}}$ whose elements g can be written respectively as follows, for $\nu = 2L$, $k'_i \leq k_i$ for each $i < \nu$, $k'_\nu \in \mathbb{N}$ and suitable q

$$g = \begin{cases} \bigcup_{j_0=1}^{k'_0} \bigcup_{j_1=1}^{k'_1} \dots \bigcup_{j_\nu=1}^{k'_\nu} v_{q(j_0,j_1,\dots,j_\nu)} & \text{for the former and} \\ \bigcup_{j_0=1}^{k'_0} \bigcup_{j_1=1}^{k'_1} \dots \bigcup_{j_\nu=1}^{k'_\nu} v_{q(j_0,j_1,\dots,j_\nu)} & \text{for the latter} \end{cases} \quad (2)$$

In short, we pass from one level to the next adding a pair of operations of the “ \cap ” kind for metamonomials and “ \cup ” for metaclauses.

2.2 Reduction step

At a given abstraction level, the *Symbol's Jump* block leaves us with a pair of borders within which any function is a candidate hypothesis for the goal concept g^* . We adopted the general strategy of obtaining the new pair through incremental changes in the original ones that do not induce a trespassing of the inner border beside the outer one, and *vice versa*. Selection of the best simplifications is managed in terms of an optimization task. Two possible approaches (mutual information maximization and fuzzy relaxation) are described in the following.

Mutual information maximization (Fitness Maximization, FM). Given the set of (meta)monomials constituting the inner border, let us focus on monomial \mathbf{a} and on any other of the remaining monomials that we denote by \mathbf{b}_i . Thus

$a \cup_i b_i$ is the inner border that contains all n^+ positive points. Let us denote by A the event: a point belongs to a , and B the event: a point belongs to b_i , for a given i . We mean by H_A the entropy associated to the first partition and by $H_{A/B}$ the conditional entropy associated to the first partition conditioned to the second one.

The mutual information [5] $I_{A,B} = H_A - H_{A/B}$ can be estimated as in [6]: We numerically discovered [6] that in learning formulas it is suitable to go in the direction of maximizing the mutual information, which is a direction opposite to what is usually suggested (see for instance [7]). In line with [8], the rationale for our strategy lies in the fact that formulas with high mutual information discover a strong structure within data, a structure that we may expect to be preserved in the new unseen data.

Fuzzy relaxation (FR). As an alternative, we may decide to broaden the contours of the inner and outer borders balancing a desirable shortening of the formula f with the undesirable loss of its description power [9]. We account for this trade-off by minimizing a cost function $O(f, \lambda)$ that takes into account the formula length and the radius of its fuzzy border, as follows.

Definition 2. Given a monomial m , for an ordered sequence $\mathbf{d} = (d_1, \dots, d_\ell)$ of length ℓ of literals from $\text{set}(m)$, let us denote by \mathbf{d}^k its prefix of length k . Let $m_{\mathbf{d}^0} = m$, and $m_{\mathbf{d}^k}$ denote the monomial obtained by flipping from 1 to 0 the crisp membership value of literal d_k in $m_{\mathbf{d}^{k-1}}$. Let us denote $\sigma(\mathbf{d}^k)$ the cardinality of the E subset belonging to $m_{\mathbf{d}^k}$ - m . We define the (fuzzy) membership function $\mu_{m_{\mathbf{d}}}(d_k)$ of a literal d_k in respect to $m_{\mathbf{d}}$ as follows

$$\mu_{m_{\mathbf{d}}}(d_k) = 1 - \frac{\sigma(\mathbf{d}^k)}{\sigma(\mathbf{d})} \quad (3)$$

An analogous definition can be given for clauses. In a local interpretation of the membership function we can consider $\mu_{m_{\mathbf{d}}}(d_k)$ as a probability estimate of finding points that belong to the fuzzy frontier outside the enlargement induced by d_k , and we can define the radius of the frontier as the mean value of the distances of points belonging to each enlargement slice from the crisp monomial m as follows.

Definition 3. Given a monomial m_i and an ordered sequence $\mathbf{d}_i = (d_1, \dots, d_\ell)$ of length ℓ of literals from $\text{set}(m_i)$, we call $m_{\mathbf{d}} - m_i$ the fuzzy frontier of m_i , and

$$\rho_i = \sum_{k=1}^{\ell} \mu_{m_{\mathbf{d}}}(d_k) \quad (4)$$

its radius.

The cost function we want to minimize with respect to the formula f is:

$$O(f, \lambda) = \lambda_1 \sum_{i=1}^m L_i + \lambda_2 \sum_{i=1}^m \rho_i + \lambda_3 \nu_0 \quad (5)$$

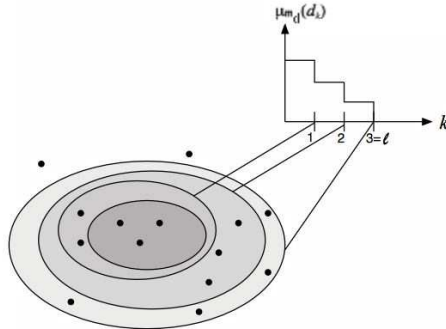


Fig. 3. The fuzzy border of a monomial. Dark region \rightarrow m ; lessening gray regions \rightarrow progressive enlargements after removal of d_1 , d_2 , and d_3 from set(m); $\mu_{m_d}(d_k)$ as in Definition 2.

where: $\lambda = (\lambda_1, \lambda_2, \lambda_3)$; f is an inner or outer border; $\sum_i L_i$ is the length of the formula, being L_i the number of literals in the i^{th} atomic formula; ν_0 is the percentage of positive examples left out of the support of f (a short way for accounting the dummy monomial cost).

3 Numerical results

We discuss effectiveness of our procedure on an artificial instance consisting in recovering formulas in the polynomial hierarchy. The experiments are aimed at learning DNFs on 12 propositional variables drawn randomly in order to have a number of terms ranging between 2 and 7. We considered 100 such formulas. For each of them we generate two kinds of training sets containing 100 examples equally partitioned in positive and negative ones. In the unbiased training set assignments of values 0 or 1 to all the propositional variables are made with same probability 0.5. In the biased training set, instead, value 1 is assigned with probability 0.7 to 4 randomly selected propositional variables, and with probability 0.5 for the others. The test set is constituted by the whole set of 4096 different 12 long Boolean vectors labeled according to the DNF to be recovered. Table 1 reports comparative performances of: a) *PACmeditation* using *FM*, b) *PACmeditation* using *FR*, c) *C4.5* [10], the well spread concept learning algorithm, where a decision tree in terms of IF-THEN-ELSE rules is drawn directly by iterated partitioning of the sampled data on the basis of mutually exclusive tests on their range. Performances are evaluated in terms of: i) residual indeterminacy, measured by the number of test set points falling in the gap between inner and outer borders (column *Gap*); ii) compression rate between original and rediscovered formulas (column ρ); and iii) test set classification accuracy.

Concerning accuracy, in methods *FM* and *FR* if we consider correctly classified the points falling in the gap, we get a lower bound to the error percentage (column *FFN*); if we consider them incorrectly classified, we get an upper bound

Table 1. Comparing the performances of *PACmeditation*, in the *FM* and *FR* release, and C4.5. μ and σ are mean and standard deviation over 100 trials. *Gap* between inner and outer border is measured in number of points falling inside it; ρ is the ratio between the found and original rule lengths; *FP*, *FFP*, *FN*, *FFN* are percentages of False Positives, Fuzzy False Positives, False Negatives, and Fuzzy False Negatives, respectively. Level 0 and Level 1 are the abstraction levels of PAC meditation; C4.5 results are conventionally reported in the Level 1

		Level 0						Level 1							
		Gap	ρ	FP	FFP	FN	FFN	Gap	ρ	FP	FFP	FN	FFN		
<i>FM</i>	PACmed	unbiased	μ	23.42	1.22	6.25	5.53	2.61	2.55	11.77	0.63	6.25	5.89	2.58	2.55
		σ	38.34	0.49	4.5	4.11	2.95	2.93	23.86	0.23	4.5	4.4	2.95	2.93	
	biased	μ	29.67	1.3	6.85	5.99	3.57	3.35	13.21	0.67	6.85	6.45	3.49	3.35	
		σ	37.6	0.6	4.76	4.46	3.47	3.49	21.59	0.28	4.76	4.61	3.54	3.49	
<i>FR</i>	PACmed	unbiased	μ	149.09	0.93	10.23	5.59	2.53	1.73	124.9	0.78	10.23	6.29	2.17	1.73
		σ	114.1	0.32	6.13	4.87	2.16	2.44	103.9	0.36	6.13	5.05	2.61	2.44	
	biased	μ	131.1	0.94	9.9	5.89	2.69	1.73	112.4	0.8	9.9	6.48	2.52	1.73	
		σ	109.7	0.35	6.46	5.03	2.9	2.11	104.3	0.31	6.46	5.19	2.7	2.11	
C4.5	unbiased	μ							1.13	8.45		5.58			
		σ							0.48	6.95		4.72			
	biased	μ							1.2	7.25		6.09			
		σ							0.51	6.51		4.93			

in the (column FN). The same happens with the negative points, for which FFP and FP percentages are defined analogously.

Concerning compression rate, the better behavior of the disjunctive representation is due to the fact that the original formula is a disjunctive form too.

The jump to abstraction level 1 improves all parameters with the sole exception of some accuracy indices. Actually we have a slight increase of the FFP percentages, since the inner border still expands, while FP and FFN remain unchanged such as the outer border.

A slight degradation of all performance indices is registered when we pass from unbiased to biased samples.

The greatest reason for employing *FR*, however, lies in the computing time. Its average in all these trials is 3.6 seconds (with almost 0 variance) on a Pentium IV as a reference architecture, which is one order less than the time (around 72 seconds) taken for *FM*. With these running times we may compete with the C4.5 algorithm ([7]). Table 1 shows that the description lengths of formulas obtained by C4.5 are meanly from 1.3 to 1.9 times greater than those provided by *FR*, with almost equal accuracy, obtained in an average time of 0.03 seconds running a program written in a language compiled into bytecode (while C4.5 is written in a language compiled into machine language).

The fact that standard deviations are almost equal to the means denotes the presence of some rare hard instances causing very high values of the indices, balancing a higher concentration of all the other cases which follow within at most 1 standard deviation from the mean.

4 Conclusions

Confining an unknown function f between one tight and one weak hypothesis is a usual way for the human brain to infer properties about the function and then take operational decisions. It is a functional extension of the confidence interval notion, thus based on statistics on the observed examples. In absence of any indication about f we commit to the example the sole role of watching for inconsistencies. Deciding to bind f through monotone formulas awards the watching points the connotation of delimiters of wide regions in the Boolean hypercube. We transfer to these regions the role of examples of f and make them more and more complex through the addition of some syntactical constraints. This is the main idea of our meditation process. By definition, under the monotonicity assumption, no negative point can be found inside the inner border and no positive point outside the outer one. Then to render the formulas understandable we accept the compromise of reducing the sharpness of the borders. The primary idea is to modify their shape maintaining a robust common structure. This gives rise to an algorithm for learning concepts that proves quite accurate when a monotone formula really underlies the data. Although it does not find the best fitting of the data, the structure it finds underlying them pays in terms of robustness versus the randomness of the training set.

References

1. Valiant L.: A Theory of the Learnable. Communications of the ACM, Vol. 27 (1984) 1134-1142
2. Mitchell, T. M.: Machine Learning. McGraw-Hill Series in Computer Science. The McGraw-Hill Companies, Inc., New York (1997).
3. Selman, B., Kautz, H.: Knowledge compilation and theory approximation. In: Journal of the ACM, Vol. 43. (1996) 193-224
4. Michalski, R. S.: A theory and methodology of inductive learning. In: Machine Learning: An Artificial Intelligence Approach. J. G. Carbonell and T. M. Mitchell (eds.) Tioga, Palo Alto (1983) 83-134
5. Cover, T., Thomas, J.: Elements of information theory, Wiley, New York (1991).
6. Apolloni, B. and Malchiodi, D. and Orovas, C. and Palmas, G.: From synapses to rules. In: Cognitive Systems Research, Vol. 3/2 (2002) 167-201
7. Quinlan, J. R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers. San Mateo, California (1993)
8. Orovas, C. and Austin, J.: A Cellular Neural Associative Array for Symbolic Vision. In: Wernter, S. and Sun, R. (eds.): Hybrid Neural Systems. Springer-Verlag, Berlin Heidelberg New York (2000) 372-386
9. Vapnik V.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin Heidelberg New York (1995)
10. Ross Quinlan Home Page. <http://www.cse.unsw.edu.au/~quinlan/>