



# PAC Meditation on Boolean Formulas

Bruno Apolloni<sup>1</sup>, Fabio Baraghini<sup>1</sup>, and Giorgio Palmas<sup>2</sup>

<sup>1</sup> Dip. di Scienze dell'Informazione, Università degli Studi di Milano

<sup>2</sup> ST Microelectronics s.r.l. Agrate Brianza (Mi) - Italy

**Abstract.** We present a Probably Approximate Correct (PAC) learning paradigm for boolean formulas, which we call PAC meditation, where the class of formulas to be learnt are not known in advance. On the contrary we split the building of the hypothesis in various levels of increasing description complexity according to additional constraints received at run time. In particular, starting from atomic forms constituted by clauses and monomials learned from the examples at the 0-level, we provide a procedure for computing hypotheses in the various layers of a polynomial hierarchy including  $k$ -term-DNF formulas at the second level. Assessment of the sample complexity is based on the notion of sentry functions, introduced in a previous paper, which extends naturally to the various levels of the learning procedure. We make a distinction between meditations which waste some sample information and those which exploit all information at each description level, and propose a procedure that is free from information waste. The procedure takes only a polynomial time if we restrict us to learn an inner and outer boundary to the target formula in the polynomial hierarchy, while an access to an NP-oracle is needed if we want to fix the hypothesis in a proper representation.

## 1 Introduction

PAC learning 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 [1]. 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. Rather we receive requisites of the class a little at a time in subsequent steps of the learning process. Thus we must have at runtime a twofold care of:

1. correctly updating current knowledge on the basis of new requisites, so that the approximation of the hypotheses on the final concept is not compromised, and
2. suitably reinterpreting examples in the light of the current knowledge, so that only their essential features are focused on, without neither missing necessary data nor recording unuseful details.

We hit these targets in learning boolean formulas through a multi-level procedure that we call *PAC-meditation*:

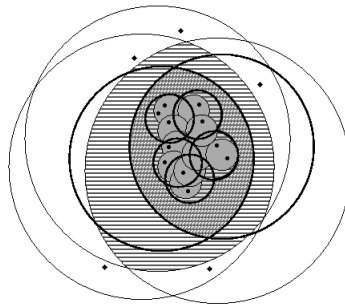
- at the first level we have two sets of positive and negative examples. From subsets of equally labelled examples we compute partial consistent hypotheses. Namely each hypothesis is consistent with a part of the positive examples and all negative

---

\* Corresponding author: e-mail [apolloni@dsi.unimi.it](mailto:apolloni@dsi.unimi.it)

examples, or vice-versa. The criterion is that the union of the hypotheses coming from positive subsets and the intersection of the other ones form two nested regions delimiting the gap where the contours of suitable consistent hypotheses are found. In Fig. 1 the gap is represented by the dashed area. We distinguish a gray region embedded in a white dashed one. Let us focus for a moment on the widest area (contoured by thin curves), which we call 0-level gap. It is delimited on the inside by a (non dashed) region we call *inner border* and, analogously, by the *outer border*.

- at further abstraction levels the partial consistent hypotheses of the immediately 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 borders. A new pair of borders are constructed running the same procedure on the so represented examples (and are contoured by bold lines in Fig. 1). Not far from what happens in the human mind, an actual benefit comes from these level jumps in case of suitable definition of the classes of formulas in the new borders. These classes induce new links between the formulas, with the twofold effect of reducing both the degrees of freedom of the final class of hypotheses, thus lowering the sample complexity of the learning problem, and narrowing the interstice between the borders, thus simplifying the search for a final hypothesis.



**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; rhombuses: negative examples.

The consistency constraint binds the whole learning process, which coincides in this respect with an efficient watching on the part of training examples sentinel that borders do not trespass forbidden points. This functionality represents the points' information content which will be managed optimally, i.e. without *information waste*. Passing from one symbolic level to another, properties about these points become points in a new

functional space (call them hyperpoints within a higher abstraction level), that are useful, in own turn, for building new properties, i.e. metaproperties on the original example space.

In this way our procedure puts a bridge between inductive and deductive learning. The atomic formulas at the first level are inductively learnt from examples [1], then they are managed through special deductive tools. This is a true different acception of agnostic learning. With the general understanding that it is very difficult to know a priori the class of the goal concept or even the set of involved variables [2], many authors infer its functional shape directly from the data within paradigms like boosting [3] or other kind of modular learning [4]. Their approaches share with the most elementary ones like decision trees [5] or Rulex [6] the idea that this shape comes uniquely from a best fitting of the training set data. Our approach aims at building a concept by improving elementary formulas using in a logical way pieces of symbolic knowledge coming from an already achieved experience. This allows for a more complex and well funded management of the trade-off between class complexity and error rate clearly synthetised by Vapnik in the problem of the structural risk minimization [7]

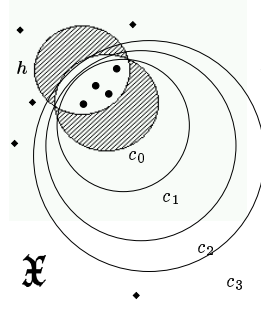
For lack of space, the exposition proceeds through a series of definitions and theorems whose proof is deferred elsewhere. In particular, in Sect. 2 we review the PAC learning theory within a new statistical framework, while Sect. 3 is devoted to introduce the conceptual framework of PACmeditation and the related theoretical results. A very short numerical section concludes the paper.

## 2 PAC Learning Theory Revisited

A very simple way we found for discussing of the statistical properties of a learning procedure is the following [8]. We have a labeled sample  $\mathbf{Z}_m = \{(X_i, b_i), i = 1, \dots, m\}$  where  $X$  takes values in  $\mathfrak{X}$ <sup>3</sup> and  $b_i$  are boolean variables. We assume that for every  $M$  and every  $\mathbf{Z}_M$  an  $f$  exists in a boolean class  $C$ , call it *concept*  $c$ , such that  $\mathbf{Z}_M = \{(X_i, c(X_i)), i = 1, \dots, M\}$ , and we are interested in the measure of the symmetric difference  $U_{c \div h}$  between another function computed from  $\mathbf{Z}_m$ , that we denote as *hypothesis*  $h$ , and any such  $c$  (i.e. the set of points where we will answer 0 using  $h(x)$  while the correct answer is  $c(x) = 1$  or vice versa, see Fig. 2).

Actually, for fixed sample  $\mathbf{Z}_m$  we can have different populations  $\mathbf{Z}_M$ , hence different  $c$ 's explaining them. These are related however to the explanation  $h$  we found for the sample, and we work precisely on this relation for computing the distribution law of the random variable  $U_{c \div h}$  as a function of the random suffix  $\mathbf{Z}_M$  of a given sample  $\mathbf{z}_m$ . This relation is very similar to the one between sample and population properties of a Bernoulli variable, as in both cases we work with 0/1 assignments. But here we need some sampled points – which we call (*outer*) *sentry points* [9] – to recognize that the probability measure of the error domain is less than a given  $\varepsilon$ . These points are assigned by a *sentinelling function*  $\mathbf{S}$  whose formal definition is given in [9], to each concept of a class in such a way that : i. they are external to the concept  $c$  to be sentinelled and

<sup>3</sup> By default capital letters (such as  $U, X$ ) will denote random variables and small letters ( $u, x$ ) their corresponding realizations; the sets the realizations belong to will be denoted by capital gothic letters ( $\mathfrak{U}, \mathfrak{X}$ ).



**Fig. 2.** A PAC learning framework.  $\mathfrak{X}$ : the set of points belonging to the cartesian plane;  $c$ : a concept from the concept class of circles;  $h$ : a hypothesis from the same concept class; bullets: 1-labeled (positive) sampled points; rhombuses: 0-labeled (negative) sampled points. Line filled region: symmetric difference.

internal to at least one other including it, ii, each concept  $c'$  including  $c$  has at least one of the sentry points of  $c$  either in the gap between  $c$  and  $c'$  or outside of  $c'$  and distinct from the sentry points of  $c'$ , and iii. they constituted a minimal set with these properties. An upper bound to the cardinality of these points is represented by the detail  $D_C$  of a concept class. For instance, the class  $C$  on  $\mathfrak{X} = \{x_1, x_2, x_3\}$  whose concepts are

$x_1$	$x_2$	$x_3$	$x_1$	$x_2$	$x_3$
$c_1 = \ominus$	$\ominus$	$-$	$c_1 = -$	$-$	$\ominus$
$c_2 = \ominus$	$+$	$+$	$c_2 = \ominus$	$+$	$+$
$c_3 = +$	$\ominus$	$+$	$c_3 = +$	$\ominus$	$+$
$c_4 = +$	$+$	$+$	$c_4 = +$	$+$	$+$

where “+” denotes an element  $x_j$  belonging to  $c_i$ , “-” an element outside  $c_i$  and  $\ominus$  a sentry point, has  $D_C = 2$ . A worst case  $\mathbf{S}$  is:  $\mathbf{S}(c_1) = \{x_1, x_2\}$ ,  $\mathbf{S}(c_2) = \{x_1\}$ ,  $\mathbf{S}(c_3) = \{x_2\}$ ,  $\mathbf{S}(c_4) = \emptyset$ . However a cheaper one is  $\mathbf{S}(c_1) = \{x_3\}$ ,  $\mathbf{S}(c_2) = \{x_1\}$ ,  $\mathbf{S}(c_3) = \{x_2\}$ ,  $\mathbf{S}(c_4) = \emptyset$ . Further examples can be found in [9]. In particular here we will refer to classes of concepts  $C \div C$  made up of the symmetric differences  $c_i \div c_j$  between concepts belonging to a same class  $C$  and its detail  $D_{C,C}$ .

A learning algorithm is a procedure  $\mathcal{A}$  to generate a family of hypotheses  $h_m$  with their respective  $U_{c \div h_m}$  converging to 0 in probability with the sample size  $m$ .

**Lemma 1.** For a space  $\mathfrak{X}$  and an unknown probability measure  $P$  on it, assume we are given i) a concept class  $C$  on  $\mathfrak{X}$  with  $D_{C,C} = \mu$ , ii) a sample  $\mathbf{Z}_m$  drawn from the fixed space and labeled according to a  $c \in C$  labeling an infinite suffix  $\mathbf{Z}_M$  of it, and iii) a fairly strongly surjective function  $\mathcal{A} : \{\mathbf{Z}_m\} \mapsto C$  misclassifying at most  $t \in \mathbb{N}$  points of total probability not greater than  $\rho$ .

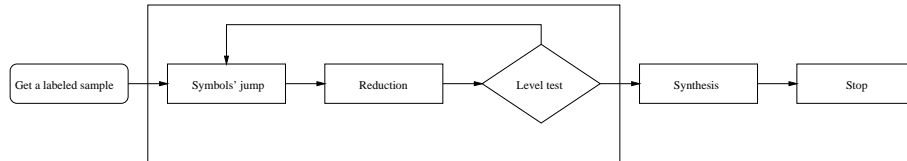
In case  $m \geq \max \left\{ \frac{2}{\varepsilon} \log \frac{1}{\delta}, \frac{5.5(\mu+t-1)}{\varepsilon} \right\}$   $\mathcal{A}$  is a learning algorithm for  $C$  such that  $P^{(M)} (U_{c \div \mathcal{A}(\mathbf{Z}_m)} \leq \max\{\rho, \varepsilon\}) \geq 1 - \delta$ .

In case  $m < \frac{1-\varepsilon}{\varepsilon} \log \frac{1}{\delta}$  no learning algorithm exists satisfying the above probabilistic inequality on the measure of the symmetric difference.  $\square$

The main lesson we draw from the above discussion is that, when we want to infer a function we must divide the available examples in two categories, the relevant ones and the mass. Like in a professor’s lecture, some, the former, straight fix the ideas, thus binding the difference between concept and hypothesis. The latter are redundant; but if we produce a lot of examples we are confident that a sufficient number of those belonging to the first category will have been exhibited.

### 3 PAC-meditation

If we do not know  $C$  in advance we propose a procedure to discover it progressively. Its block diagram is shown in Fig. 3. Given a set of positive and negative examples the procedure core consists in the iterated implementation 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; and ii. a *Reduction* step for refining these properties. Namely, we start considering a set of minimal hypotheses about the goal formula that are consistent with positive examples and maximal for the negatives ones. Thus a second step is devoted to broadening or narrowing these hypotheses with: i. the constraint of not violating the examples consistency and ii. the scope of narrowing the gap between the union of minimal hypotheses (the mentioned inner border) and the intersection of the maximal hypotheses (the mentioned outer border). This happens at zero level. To increase the abstraction level we may restart the two steps after assuming the minimal hypotheses as positive (hyper)points at 1-level, maximal hypotheses as negative hyperpoints, and searching for new hypersymbols to describe properties on these new 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 translate in additional constraints in rebuilding the borders. Once we are satisfied with the abstraction level reached (or simply do not plan on achieving new formal knowledge), the level test in Fig. 3 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. 3.** Block diagram of PAC-meditation.

In this paper we restrict ourselves to classes of monotone boolean formulas. With  $\mathfrak{X} = \mathbf{X}_n \equiv \{0, 1\}^n$  we construct the atomic components of 0-level borders which call canonical monomial and clauses described by the propositional variables  $V_n = \{v_1, \dots, v_n\}$  as follows.

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

These formulas do not constrain the final expression of  $g^*$  in that 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 can not belong to it given a negative one. Moreover, let us consider a function  $\mathbf{s}$  that, in analogy to  $\mathbf{S}$ , assigns to a concept a set of inner sentry points sentinelling from inside the concept w.r.t. other concepts included in it. Thus they are a minimal set of points internal to the concept  $c$  to be sentinelled and external to at least one other included in it, with analogous features and functions. Our atomic formulas need only one example as inner or outer frontier.

According to the above canonical monomials are a richer representation of positive points and a more concise one as well in that, if one monomial contains another we can skip the latter from the set. Their union constitutes an inner border (the union of the thin contoured gray circles in Fig. 1) since represents a minimal hypothesis on  $g^*$ . Similar properties hold for the canonical clauses, whose intersection now represents the maximal hypothesis consistent with  $g^*$ , and then an outer border. These duties derive from the fact that they represent properties which we infer from the points after the monotonicity assumption. These properties pivot around the fact that positive examples are inner sentries for these monomials and negative examples for the clauses. Now, to render this prerogative proof against any other representation through monomials (clauses), i.e. any other consistent association of monomials to inner points, we must fix these examples as sentry points of the largest expansions of canonical monomials (narrowing of canonical clauses) which still prove consistent with negative (positive) points. This is the distinguishing feature of our abstraction process: we pass from a lower to higher level representation of partial hypotheses in such a way that new sentry points are a subset of older ones (with some points possibly becoming useless due to the expansion).

Applying the distributive property to the union of two monomials:  $v_i v_j v_k \vee v_l v_m v_n v_r = (v_i \vee v_l v_m) \wedge (v_i \vee v_n v_r) \wedge (v_j v_k \vee v_l v_m) \wedge (v_j v_k \vee v_n v_r)$  we obtain a new monomial where literals are constituted by clauses and, in turn, literals in the clauses are substituted by monomials. Let us generalize the operation, keeping groups of at most  $k_2$  monomials and splitting them in such a way that at most  $k_1$  hyperclauses arise. Bounds on  $k$ 's stand for requisites of conciseness on the formula description, i.e. for a compression of our knowledge. Then we examine the case of extending (enlarging) the single atomic formulas in a consistent way. We do the same with hyperclauses. A very easy procedure for doing these tasks is reported in [10].

Cycling along the block diagram in Fig.3 and denoting  $\cup^t = \cap^{t-1} = \cap$  if  $t$  is odd and  $\neq 0, \cup$  otherwise, for  $t \geq 0$ , at the  $L^{\text{th}}$  abstraction level we obtain formulas belonging to the families of hyper- $L$ -monomials  $\mathbf{G}_{n;k_0,k_1,\dots,k_{\nu-1}}$  and hyper- $L$ -clauses  $\mathfrak{G}_{n;k_0,k_1,\dots,k_{\nu-1}}$  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  $q$

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

In short, we pass from one level to the next adding a pair of operations of the “ $\cap$ ” kind for hypermonomials and “ $\cup$ ” for hyperclauses. When we are satisfied with the achieved abstraction level, we abandon the loop and try to synthesize the two borders in a unique formula in the *Synthesis* block, meeting possible further requirements. Obviously, not each formula may comply with the actual borders, since they come from a somehow biased gap reduction. We denote *mininside* and *maxinside* the classes  $\mathbf{C}_m(r; L)$  and  $\mathbf{C}_M(r; L)$  of formulas given by the disjunction of at most  $r$  hyper- $L$ -monomials and the conjunction of at most  $r$  hyper- $L$ -clauses respectively. These formulas are obtained from an exhaustive check on all the possible  $r$ -partitions of the hyperpoints constituting the inner or outer border. For  $r$  growing with  $n$  this constitutes a highly costly computational job, the escape from which is to learn an esier hypothesis. The complexity of a learning job indeed might strongly depend on the representation of the hypothesis. For instance, it is well known that learning  $k$ -term-DNF( $n$ ) formulas is NP-hard for every preassigned  $k \geq 3$  but is polynomial if werepresent them through  $k$ -CNF formulas (a less concise representation) [11]. We speak of *proper learning* when concept and hypothesis classes coincide. In our framework, we can decide either *proper learning* the final formulas by activating *Synthesis* or *non proper learning* it by precisely relying on the current inner or outer borders. Since getting the accuracy targets  $\varepsilon$  and  $\delta$  as in Lemma 1 requires in any case a polynomial number of examples we have:

**Lemma 2.** [10] *At every fixed abstraction level, PAC-meditation algorithm supplies in polynomial time inner and outer borders as non proper hypotheses for the target concept with accuracy parameters  $\varepsilon$  and  $\delta$ . Learning mininside or maxinside classes is an NP-easy problem.*

Under sentinels management perspective we start associating an atom (monomial or clause) to each example. Then we reduce the number of atoms checking inclusion relations either during the symbolic jump or after the reduction of the formulas. We still have monomials and clauses, each needing a unique sentinel. Iteration of the abstraction module leads similarly to further sentinels reductions. Namely, at the first level each monomial is a sentinel of the 1-level hyperformulas. But since a hypermonomial for instance comes from the union of more than one monomial it may need more than one hyperpoint for its sentinel. The point we stress is that our symbolic representation is such that the hyperformula will be sentineled by the same number of examples, whatever the abstraction level we use for representing the sentinels.

**Definition 2.** *Given a concept class  $\mathbf{G}_n$  on  $\mathbf{X}_n$ , we say that  $\mathbf{G}_n$  is learnable without information waste up to level  $L$  from its borders if there exists an algorithm that for any example set  $E = E^+ \cup E^-$  produces consistent hypotheses  $h \in \mathbf{G}_n$  whose borders at level  $L$  are sentineled by a same subset of  $E$ , whatever the level  $i \leq L$  of the abstraction at which they are represented.*

**Theorem 1.** [10] *PAC-meditation learns mininside  $\mathbf{C}_m(r; L)$  and maxinside  $\mathbf{C}_M(r; L)$  without information waste up to level  $L$  whenever *Synthesis* finds a solution.*

## 4 Numerical Results and Conclusions

We have applied the PAC-meditation algorithm in many case studies and real world problems. Concerning the formers a  $k$ -term-DNF formulas are learnt at the first abstraction level as they belong to  $\mathbf{G}_{n,a}$  for  $a(ny)$  number of literals per term. The complexity of the problem, NP-hard indeed, descends from the necessity of reducing to  $k$  the number of monomials originally raising from the positive examples.

We treated with our approach the problem of learning the symbolic representation of some emotional states starting from the speech of some people involved in a talk show. Reports on the solution can be found at the web site <http://www.image.ntua.gr/physta>. The table below is a typical report of a learning session allowing us to state logical necessary and sufficient conditions for characterizing the emotion sadness at level 1, where symbols  $a$  to  $f$  are linked to symbols  $v_1$  to  $v_{24}$  through relations such as  $a = v_{23}v_{24}$ ,  $b = v_{15} + v_{22}$ ,  $c = v_{15} + v_{13}$ .

Border	1-level formulas
Inner	$ab + cde + def$
Outer	$d * (c + b + f) * e$

Splitting the learning process in a sequence of two sided converging approximations appears an efficient approach typical of the human brain. The procedure we propose can be easily extended in two directions. More specific constraints can be stated for the set-union and set-intersection operations at the basis of the abstraction jumps, to embed other kinds of formal knowledge in addition to the bounds on the hyperterm complexity. In addition, we might consider a lot of relaxed meditation schemes, based for instance on neural network and fuzzy sets paradigms.

## References

- [1] Valiant, L.G.: A theory of the learnable. *Comm. of the ACM* **11** (1984) 1134–1142
- [2] Blum, A.: Learning boolean functions in an infinite attribute space. *Machine Learning* **9** (1992) 373–386
- [3] Schapire, R.E.: The strength of weak learnability. *Machine Learning* **2** (1990) 197–227
- [4] Linial, N., Mansour, Y., Rivest, R.L.: Results on learnability and the vapnik-chervonenkis dimension. *Information and Computation* **90** (1991) 33–49
- [5] Quinlan, J.R.: *C4.5: programs for machine learning*. Morgan Kaufmann Publishers, San Mateo, California (1993)
- [6] Andrews, R., Geva, S.: Inserting and extracting knowledge from constrained backpropagation network. In: *Proc. 6th Australian Conference on Neural Networks*, Sidney (1995) 29–32
- [7] Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, New York (1995)
- [8] Apolloni, B., Malchiodi, D., Orovas, C., Palmas, G.: From synapses to rules. *Cognitive Systems Research* (2002) in press.
- [9] Apolloni, B., Chiaravalli, S.: Pac learning of concept classes through the boundaries of their items. *Theoretical Computer Science* **172** (1997) 91–120
- [10] Apolloni, B., Baraghini, F., Palmas, G.: PAC meditation on boolean formulas. Technical report, Università degli Studi di Milano (2001)
- [11] Pitt, L., Valiant, L.: Computational limitations on learning from examples. *J. ACM* **35** (1988) 965–984