TCS 4562
pp: 1--23 (col.fig.: Nil)
PROD. TYPE: COM
ED: SG
PAGN: Vidya – SCAN: Jane
ARTICLE IN PRESS

ELSEVIER
Theoretical Computer Science ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮
Theoretical Computer Science
www.elsevier.com/locate/tcs

Complexity in the case against accuracy estimation

Richard Nock*

GRIMAAG-Département Scientifique Interfacultaire, Université des Antilles-Guyane, Campus de Schoelcher, BP 7209, 97233 Schoelcher, Martinique, France

Received 28 February 2001; received in revised form 4 February 2002; accepted 30 May 2002
Communicated by G. Ausiello

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 and comparing classifiers, 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 Receiver Operating Characteristic 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 criteria's demands. The third point is the equivalence in difficulty of all these different criteria. In contrast, the single accuracy's optimization appears to be tractable in this framework.

1. Introduction

An essential task of Machine Learning (ML) and Data Mining (DM) systems is related to classification. This basically consists in giving the most accurate answer

* Tel.: +33-596-72-73-64; fax: +33-596-72-73-62.
E-mail address: rnock@martinique.univ-ag.fr (R. Nock).
URL: http://www.martinique.univ-ag.fr/~rnock

0304-3975/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0304-3975(02)00573-X

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, i.e., the frequency (or probability) of correct predictions made [3,17]. 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. [17]), as well as to compare different classifier learning algorithms to decide which one should be used preferentially (see e.g. [22]). 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 deem our approach 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/completion.

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 [20]. This is obviously not true: skewed distributions are frequent in agronomy, or more generally in life or earth sciences. For example, no more than 6% of the human DNA represents coding genes [22]. Another example is the oil spill detection problem of [17], 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 [17]. 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 [17], 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. [17], 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 [17].

This last example shows two important and typical phenomena in real-world problems. First, the balanced distributions assumption is actually false. Second, the misclassification of some examples may be of heavy consequences, a cost which is not integrated in the accuracy. Fraud detection is another good example of such a cost-sensitive situation [22], 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 [3]. 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" [17].

Furthermore, the accuracy may be inadequate in some cases because other parameters are to be taken into account. Some works [16] report the need to add an information measure to the accuracy, to eliminate the influence of prior probabilities. Constraints on size parameters (see [18,19]) are sometimes to be used because we want to obtain small formulas, to ease their interpretability by the system end-user.

Finally, some works also report the experimental convenience that reducing the size of the data itself can have when simply optimizing the accuracy [24]. 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 their accuracy too much. Sometimes, it can even provide a way to improve their accuracy on hard problems. Whereas experiments show that feature reduction can be a good criterion to optimize in conjunction with the accuracy, one may wonder how these two constraints computationally interact.

To examine the possible influence of all these completion/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 [26], in 1998, almost 70 applications were using ILP formalism, 20 of which were science applications, which can be partitioned into biological (four) and drug design (16) applications. With the increasing popularity of ILP, in particular to address complex domains, this number has certainly increased since then. ILP–ML algorithms have been applied with some success in areas of biochemistry and molecular biology [26]. 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 [13], 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/completion. More precisely, here is the structuring of the argument.

First, 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 [17–19,24]. We do not integrate in our criteria the information measure of [16], 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. [20–22]).

Finally, we investigate the replacement of the error by a single replacement criterion. Two candidates we study are criteria proposed in [20], and used in [17].

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 Horn clauses sets of unbounded size (even exponential), 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 its tightness. Furthermore, the effects of the constraints on optimal

accuracies vanish as the number of predicates increases, since 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 the two appendices. The first presents a synthetic view of all proofs, the second 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 $\mathscr{C}$ and $\mathscr{H}$ two classes of concept 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 $\mathscr{C}$. In real-world domains, we do not know the target concept's class, that is why we have to make ad hoc choices for $\mathscr{H}$ with a powerful enough formalism, yet ensuring tractability. Even if some benchmark problems appear to be easily solvable [10], ML applications, and particularly ILP, face more difficult problems [26], for which the choice of $\mathscr{H}$ is crucial.

After the choice of $\mathscr{H}$, 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 [22,21]. We also consider a two-class 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 then use it 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 draws its roots in two fundamental bodies, the so-called Probably Approximately Correct (PAC) learning model of [27], and the Statistical Learning Theory, fathered by Vapnik [29]. The principle is that the examples are drawn from some

unknown, but fixed distribution D, and labeled according to an unknown $c \in \mathscr{C}$. If we suppose that the representation space is discretized, then we can denote the accuracy of some $h \in \mathscr{H}$ 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, 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. [29]). The objective of our 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 [12]), 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 [27]. Some of our results can be extended to negative results on the PAC-derived robust learning model of [9,11]. 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:

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

$q(\ldots)$ is called the head of the clause and the conjunction $a_1(\ldots) \wedge a_2(\ldots) \wedge \ldots \wedge a_n(\ldots)$ 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 $\mathscr{L}$ and a correct inference relation on $\mathscr{L}$, the problem can be formalized in a general way as follows [11]:

Definition 1. Given:

- A background knowledge $\mathscr{B}\mathscr{K}$ expressed in a language $\mathscr{L}\mathscr{B} \subseteq \mathscr{L}$,
- A set of examples $\mathscr{S}$ in a language $\mathscr{L}\mathscr{S} \subseteq \mathscr{L}$, consisting of positive examples, $\mathscr{S}^+$, and negative examples, $\mathscr{S}^-$, such that $\mathscr{B} \not\models \mathscr{S}^+$ ($\mathscr{B}$ does not entail the positive examples) and $\mathscr{B}, \mathscr{S} \not\models \square$ ($\mathscr{S}$ is consistent with $\mathscr{B}$).
- A hypothesis class $\mathscr{H}$ described over a language $\mathscr{L}\mathscr{H} \subseteq \mathscr{L}$,

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

$$\mathscr{B} \wedge h \models \mathscr{S}^+, \tag{1}$$

$$\mathscr{B} \wedge h \not\models \mathscr{S}^- \tag{2}$$

i.e., $\mathscr{B}$ 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 [13,4]. A usual restriction makes use of ground background knowledge, i.e., 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 [14], 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 however leads to a modification of the learning problem, as stated in the following lemma:

Lemma 3 (Kietz, [13]). *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 $e \in \mathscr{S}$ and $b \in \mathscr{BK}$.*

This lemma allows us to incorporate the background knowledge in the new examples (and is thus empty). Our results 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 [13,4]. 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.
- $\mathscr{BK}$ 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 [11]: find a hypothesis $h \in \mathscr{H}$ such that

$$\mathscr{B} \wedge h \models \mathscr{S}^+, \tag{3}$$

$$\mathscr{B} \wedge h \not\models \mathscr{S}^- \tag{4}$$

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

3. Replacement criteria and the hardness technique

3.1. Extending the accuracy

For any fixed positive rational v, we use the following adequate notion of distance [1] between two reals u, v: $d_v(u,v) = |u-v|/(u+v+v)$. We also use eight rates on the examples (definitions differ slightly from [22]):

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

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

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

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

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. [17], which ignores precision. Some authors, such as [16], 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 [20], or precision or recall measures [20,17], or size parameters [18,24]. Each of them is parameterized by a number ζ (between 0 and 1), and defines a subset of $\mathscr{H}$, which shall be parameterized by D if the distribution controls the subset through the constraint. The first three subsets of $\mathscr{H}$ contain hypotheses for which the FP and FN are not far from each other, or a one-side error is upper bounded:

$$\mathscr{H}_{D,1}(\zeta) = \{h \in H \,|\, d_v(FP, FN) \leqslant \zeta\}, \tag{5}$$

$$\mathscr{H}_{D,2}(\zeta) = \{h \in H \,|\, FN \leqslant \zeta\}, \tag{6}$$

$$\mathscr{H}_{D,3}(\zeta) = \left\{h \in H \,|\, FN \leqslant \frac{1}{\zeta} FP\right\}. \tag{7}$$

The two following subsets are parameterized by constraints equivalent to some frequently encountered in the information retrieval community [25], respectively (1 minus) the precision and (1 minus) the recall criteria.

$$\mathscr{H}_{D,4}(\zeta) = \left\{h \in H \,|\, \frac{FP}{TP+FP} \leqslant \zeta\right\}, \tag{8}$$

$$\mathscr{H}_{D,5}(\zeta) = \left\{ h \in H \mid \frac{FN}{TP + FN} \leqslant \zeta \right\}. \quad (9)$$

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) [4] has the following form:

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

Here, q, $a_1, a_2, \ldots, a_n$ 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 $\mathscr{H}$ are parameterized by formulas, respectively, having a sufficiently small fraction of the available predicates, or having a sufficiently small overall size:

$$\mathscr{H}_6(\zeta) = \left\{ h \in H \mid \frac{\#\texttt{Predicates}(h)}{\#\texttt{Total\_predicates}} \leqslant \zeta \right\}, \quad (10)$$

$$\mathscr{H}_7(\zeta) = \left\{ h \in H \mid \frac{\#\texttt{Whole\_predicates}(h)}{\#\texttt{Total\_predicates}} \leqslant \zeta \right\}. \quad (11)$$

The division by the total number of different predicates in $\mathscr{H}_7(\zeta)$ 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 $i \in \{1, 2, \ldots, 7\}$, can we find an algorithm returning a set of Horn clauses from $\mathscr{H}_{(D,)i}(\zeta)$ 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 [5]. It has been used in machine learning recently [20–22] 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. Fig. 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 [22], or a real-valued confidence such as in AdaBoost [23], 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 $y = x$. One important thing to note is that the ROC representation gives the

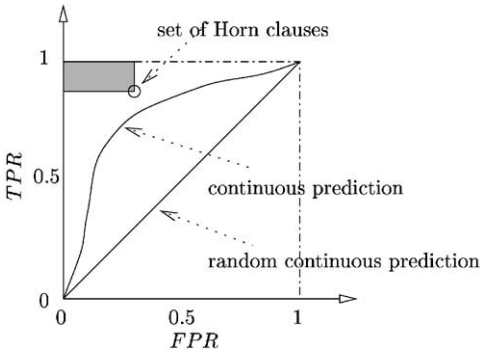

Fig. 1. The ROC analysis of a learning algorithm, with three general classifications: discrete, continuous, and default.

behavior of an algorithm without regarding the class distribution or the error cost [20]. 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:

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

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 Fig. 1). From that, the second problem we address is the following:

Problem 2. *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 [20]. Some researchers [20] propose the use of the following criterion: $(1 - FPR) \times TPR$. A geometric interpretation of the criterion is the following [20]: 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 $(1 - FPR) \times TPR \geqslant \gamma$, if such a hypothesis exists?*

In [17], 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 [17].

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 $\mathscr{H}_{(D,)a}(\zeta)$ by one still in $\mathscr{H}_{(D,)a}(\zeta)$. However, the best concept in $\mathscr{H}_{(D,)a}(\zeta)$ generally does not have an error equal to the optimal one over $\mathscr{H}$ given D, $opt_{\mathscr{H}_D}(c)$. In fact, it has an error that we can denote

$$opt_{\mathscr{H}_{(D,)a}(\zeta)}(c) = \min_{h' \in H_{(D,)a}(\zeta)} \sum_{h(x) \neq c(x)} D(x).$$

$$\geqslant opt_{\mathscr{H}_D}(c)$$

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

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

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

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

$$n^*_{\mathscr{H}_{(D,)a}(\zeta)} \leqslant (n_e)^3. \tag{15}$$

Note that the constraint makes generally

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

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

$$d_v(opt_{\mathscr{H}_D}(c), opt_{\mathscr{H}_{(D,)a}(\zeta)}(c)) = \mathrm{o}(1) \tag{17}$$

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 (#`Total_predicates` $\rightarrow \infty$). 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 a y 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*$(\mathscr{H}, \gamma_{FPR}, \gamma_{TPR})$). • *Name*: Approx-Constrained-ROC$(\mathscr{H}, \gamma_{FPR}, \gamma_{TPR})$.

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

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

Definition 6 (*Approx-Constrained-Single*$(\mathscr{H}, \Gamma, \gamma)$). • *Name*: Approx-Constrained-Single$(\mathscr{H}, \Gamma, \gamma)$.

- *Instance*: A set of negative examples $\mathscr{S}^-$, a set of positive examples $\mathscr{S}^+$, a rational weight $0 < w(x_i) = \frac{n_i}{d_i} < 1$ for each example x_i. We assume that $\sum_{x \in \mathscr{S}^+ \cup \mathscr{S}^-} w(x_i) = 1$.
- *Question*: Does there exist a hypothesis $h \in \mathscr{H}$ satisfying $\Gamma(h) \leqslant \gamma$?

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 5. *We have*:

(i) [1] $\forall 0<\zeta<1$, *Approx-Constrained*$(FfHC,(1,\zeta))$ *is Hard, when* $\nu<(1-\zeta)/\zeta$
(ii) [2] $\forall 0<\zeta<\frac{1}{2}$, *Approx-Constrained*$(FfHC,(2,\zeta))$ *is Hard.*
(iii) [3] $\forall 0<\zeta<1$, *Approx-Constrained*$(FfHC,(3,\zeta))$ *is Hard.*
(iv) [4] $\forall 0<\zeta<1$, *Approx-Constrained*$(FfHC,(4,\zeta))$ *is Hard.*
(v) [5] $\forall 0<\zeta<1$, *Approx-Constrained*$(FfHC,(5,\zeta))$ *is Hard.*
(vi) [6] $\forall 0<\zeta<1$, *Approx-Constrained*$(FfHC,(6,\zeta))$ *is Hard.*
(vii) [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 [8] ($NP \not\subset ZPP$), which involves randomized complexity classes. We refer the reader to the paper of [8] 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 B; the other results presented strictly use the same type of reduction, and are eventually sketched [6,7]. Also, in Appendix A, 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.

Beyond the range of constraints that our negative results cover, note that any other additional constraint aside from the accuracy is a natural candidate to test the existence of negative results, 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. Therefore, another incidence of our results is that in between the two extreme situations (no/over constrained requirement), 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 4–6 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.

4.2. Replacing the accuracy: the ROC analysis

In this section, we show that the classical ROC components as described by Provost et al. [22] and Provost and Fawcett [21] 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 6. $\forall 0 < \gamma_{FPR}, \gamma_{TPR} < 1$, *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 Section 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 7. $\exists \gamma_{\max} > 0$ *such that* $\forall 0 < \gamma < \gamma_{\max}$, *Approx-Constrained-Single*$(FfHC, (1 - FPR) \times TPR, \gamma)$ *is Hard.*

(Proof included in Appendix B). As far as we know, $\gamma_{\max} \geqslant \frac{175}{41{,}616}$ (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 [2], which is an accurate composition of precision and recall (see Section 3.1 for their definition), useful for text categorization problems [2]. 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 [12] that negative results on such problems can sometimes be extended to negative results for PAC-type learning models [27]. 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 arising in exactly the same setting, but in which we replace the set of examples by a so-called oracle [12], drawing examples on demand, following a probability distribution D unknown, but fixed. Suppose that the requirement on the constraint defining $\mathscr{H}_{(D,)a}(\zeta)$ 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 $(< opt_{\mathscr{H}_{(D,)a}(\zeta)}(c) + \varepsilon)$, for some parameters $\varepsilon, \delta > 0$. If we require that the computational time be a polynomial in $1/\varepsilon, 1/\delta$, 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 [9,11], to which add the requirement that the outputs satisfy a constraint (among our seven first constraints). In that case, following a standardized approach [9,11,12], it is easy to show that a negative result regarding 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. One example is the subclass of DNF (disjunctive normal form formulas [28]) 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 [27], studied early by Valiant himself [28], and still raising some of the most important problems in computational learning theory [15], 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 ML and DM, a problem recently addressed 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 initially brought in [22], is therefore more complicated than usually presented in ML or DM papers. This justifies the title of our paper, which can be read in two ways, either presenting new aspects of the difficulty (complexity) of the task to find new criteria to replace the accuracy, or presenting (structural) complexity issues about the possible replacements/completions. One important thing about our results is that the complexity results go beyond the usual intractability results related to ML (or DM). 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, arguing against the use of the accuracy as the optimization criterion for the induction of classifiers [23]. This approach, called boosting, has been plebiscited as one of the best currently available in classification [6]. However, it raises conjectures about the tractability of the optimization of these new criteria [23] in some cases. This shall certainly be the subject of future studies.

Appendix A. the global reduction

Reductions are achieved from the *NP*-Complete problem "Clique" [7]:

Definition 8 (Clique). • *Name*: Clique.
- *Instance*: A graph $G=(X,E)$, an integer k.
- *Question*: Does there exists a clique of size $\geqslant k$ 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 9. (i) *We can suppose that* $\binom{k}{2} \leqslant |E|$, *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* $k = |X|^{\alpha}$.

Proof. (i) is immediate; (ii) follows from [8]: 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 $l < |X|^{1-\beta}$. The idea is then to make k fall somewhere in between l and $l \times |X|^{\beta}$. For $k = |X|^{\alpha}$ ($\alpha \in]0,1[$), this is immediate (if $\alpha > \frac{1}{2}$, we pick $1 > \beta > \alpha$ and if $\alpha \leqslant 1/2$, we pick $1 > \beta > 1 - \alpha$); for $k = \alpha|X|$, whenever the graph is large enough and satisfies $l \times |X|^{\beta} < \alpha|X|$, then we simply add u new vertices, each linked to all other vertices. Picking $u = (\alpha/(1-\alpha))|X| - l(1 + |X|^{\beta})/(2(1-\alpha))$ is enough to make k fall in the interval $]l + u, l|X|^{\beta} + u[$. This ends the proof of the theorem. □

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

- Define a set of $|X|$ unary literals $a_1(.), \ldots, a_{|X|}(.)$, 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:

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

Examples are described in the following way:

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

$$\forall (i,j) \in E,\ p_{i,j} = q(l_{i,j}) \leftarrow \wedge_{k \in \{1,\ldots,|X|\} \setminus \{i,j\}} a_k(l_{i,j}) \wedge t(l_{i,j}), \tag{A.1}$$

$$p_1 = q(l_1) \leftarrow \wedge_{k \in \{1,\ldots,|X|\}} a_k(l_1) \wedge t(l_1), \tag{A.2}$$

$$p_2 = q(l_2) \leftarrow a_1(l_2). \tag{A.3}$$

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

$$\forall i \in \{1, \ldots, |X|\}, n_i = q(m_i) \leftarrow \wedge_{k \in \{1,\ldots,|X|\} \setminus \{i\}} a_k(m_i) \wedge t(m_i) \tag{A.4}$$

$$n'_1 = q(l_3) \leftarrow \wedge_{k \in \{1,\ldots,|X|\}} a_k(l_3) \wedge s(l_3) \wedge t(l_3) \tag{A.5}$$

$$n'_2 = q(l_4) \leftarrow \wedge_{k \in \{1,\ldots,|X|\}} a_k(l_4) \wedge s(l_4). \tag{A.6}$$

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

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 **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 $\mathscr{H}_6(\zeta)$ (resp. $\mathscr{H}_7(\zeta)$) 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 $h = \{h_1, \ldots, h_l\}$ a set of Horn clauses, we define the set

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

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

- $E(k')$ is strictly increasing, $\sum_{(x \in \mathscr{S}^+ \wedge h(x)=0) \vee (x \in \mathscr{S}^- \wedge h(x)=1)} w(x) \geqslant E(k')$ and $E(k) = \gamma$.
- $F_a(k')$ is strictly decreasing, it is a lowerbound of the function inside $\mathscr{H}_{(D,)a}(\zeta)$, 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 $h \in \mathscr{H}_{(D,)a}(\zeta)$ satisfying $\sum_{(x \in \mathscr{S}^+ \wedge h(x)=0) \vee (x \in \mathscr{S}^- \wedge h(x)=1)} w(x) \leqslant \gamma$, its error rate implies $k' \leqslant k$ and constraint implies $k' \geqslant k$. So $|\mathscr{I}| = k' = k$. 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 $\{n_i\}_{i \in \{1, \ldots, |X|\}}$. It comes that the $\binom{k}{2}$ correspond to the $\binom{k}{2}$ edges linking the $|\mathscr{I}| = k$ vertices corresponding to negative examples badly classified. We therefore dispose of a clique of size $\geqslant k$.

Conversely, $\forall a \in \{1, 2, 3, 4, 5, 6, 7\}$, given some clique of size k whose set of vertices is denoted $\mathscr{I}$, we show that the singleton

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

is an element from $\mathscr{H}_{(D,)a}(\zeta)$ satisfying $\sum_{(x \in \mathscr{S}^+ \wedge h(x)=0) \vee (x \in \mathscr{S}^- \wedge h(x)=1)} w(x) \leqslant \gamma$. In this case, $n_{\mathscr{H}_{(D,)a}(\zeta)}$ drops down to $\mathscr{O}(n_e)$.

All distributions used in Theorems 5 and 7 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 among 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 6 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 $a_j(.)$. □

Appendix B. proofs of negative results

B.1. Proof of point [1], Theorem 5

We fix the following weights for positive examples:

$$w(p_2) = \frac{1}{2(1-\zeta)}(\zeta v + |X|^2 w^-(1+\zeta)),$$

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

$$w(p_1) = \frac{1}{2}\left(1 - \frac{\zeta v}{1-\zeta}\right) - \frac{1}{2}\left(w^-\left[|X|^2\left(\frac{1+\zeta}{1-\zeta} + |X| - k\right)\right]\right) - \frac{1}{2}\left(w^+\left[\frac{1-\zeta}{1+\zeta}\left(|X| - \binom{k}{2}\right) + |X|\right]\right).$$

We fix the following weights for negative examples:

$$w(n_2') = \frac{1}{2}$$

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

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

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

is polynomial. Define the functions:

$$\forall k' \in \{0;1\}, E(k') = |E|w^+ + k'w^- + w(p_2) + w(n_1),$$

$$\forall 2 \leqslant k' \leqslant k_{\max}, E(k') = \left(|E| - \binom{k'}{2}\right) w^+ + k'w^- + w(p_2) + w(n_1),$$

$$\forall k_{\max} < k' \leqslant |X|, E(k') = k'w^- + w(p_2) + w(n_1)$$

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

$$\forall k' \in \{0;1\}, F_1(k') = \frac{||E|w^+ - k'w^- + w(p_2) - w(n_1)|}{v + |E|w^+ + k'w^- + w(p_2) + w(n_1)},$$

$$\forall 2 \leqslant k' \leqslant k_{\max}, F_1(k') = \frac{|\left(|E| - \binom{k'}{2}\right) w^+ - k'w^- + w(p_2) - w(n_1)|}{v + |E|w^+ + k'w^- + w(p_2) + w(n_1)},$$

$$\forall k_{\max} < k' \leqslant |X|, F_1(k') = \frac{|-k'w^- + w(p_2) - w(n_1)|}{v + |E|w^+ + k'w^- + w(p_2) + w(n_1)}$$

(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,

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

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

$$d_v\left(\sum_{h(x)\neq 1=c(x)} w(x); \sum_{h(x)\neq 0=c(x)} w(x)\right) \geqslant F_1(k').$$

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

Reciprocally, the Horn clause h constructed in Appendix A satisfies both relations $h \in \mathscr{H}_{\{w_i\},1}(\zeta)$, and $\sum_{h(x)\neq c(x)} w(x) \leqslant \gamma$. 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,

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

and $h \in \mathscr{H}_{\{w_i\},1}(\zeta)$. We also have $\sum_{h(x)\neq c(x)} w(x) = E(k) = \gamma$.

The reduction is achieved. We end by a remark on $d_v(opt_{\mathscr{H}_{\{w_i\}}}(c), opt_{\mathscr{H}_{\{w_i\},1}(\zeta)}(c))$. We have

$$\begin{aligned} |opt_{\mathscr{H}_{\{w_i\}}}(c) - opt_{\mathscr{H}_{\{w_i\},1}(\zeta)}(c)| &\leqslant 1 - w(p_1) - w(p_2) - w(n'_1) - w(n'_2) \\ &\leqslant |E|w^+ + |X|w^- \end{aligned}$$

and

$$\begin{aligned} opt_{\mathscr{H}_{\{w_i\}}}(c) + opt_{\mathscr{H}_{\{w_i\},1}(\zeta)}(c) + v &\geqslant 2(w(p_2) + w(n'_1)) + v \\ &\geqslant \frac{\zeta v}{(1-\zeta)} + v. \end{aligned}$$

Therefore, we get

$$d_v(opt_{\mathscr{H}_{\{w_i\}}}(c), opt_{\mathscr{H}_{\{w_i\},1}(\zeta)}(c)) = o(1).$$

B.2. Sketch of proof of points [6] and [7], Theorem 5

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

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

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):

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

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

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

$$\forall k' \in \{0;1\}, E(k') = |E| w^+ + k' w^- + w(p_2),$$

$$\forall 2 \leqslant k' \leqslant k_{\max}, E(k') = \left(|E| - \binom{k'}{2}\right) w^+ + k' w^- + w(p_2),$$

$$\forall k_{\max} < k' \leqslant |X|, E(k') = k' w^- + w(p_2),$$

(We have $E(k) = \gamma$). From that, it comes that the predicates that 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

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

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.

B.3. Proof of Theorem 7

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

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

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

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

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

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

The choice of $\gamma_{\max}$ 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 9, 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

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

Note that $\gamma(|X|+k)/(|X|-k) = 17\gamma/7$. Suppose we choose $\gamma \leqslant 1/17 \times 7 \times 25/17 \times 144$ (details of γ are given for the sake of clarity in the proof). Then, with such values, we have for the denominator of w^+:

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

which leads to an upperbound for $|E|w^+$ which is (taking into account that $|E| < |X|^2/2$):

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

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

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

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

$$\forall k' \in \{0; 1\}, F(k') = w(p_1) \times (|X| - k')w^-,$$

$$\forall 2 \leqslant k' \leqslant k_{\max}, F(k') = \left(\binom{k'}{2} w^+ + w(p_1)\right) \times (|X| - k')w^-,$$

$$\forall k_{\max} < k' \leqslant |X|, F(k') = (|E|w^+ + w(p_1)) \times (|X| - k')w^-.$$

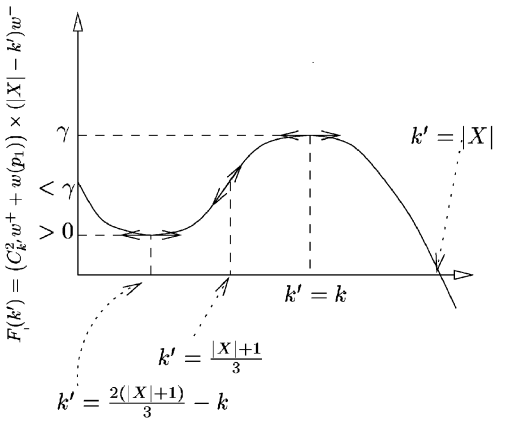

Fig. 2. Scheme of $F(k')$.

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 $k' = k'' = |X| + 1/3$. 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 so as to have $F(0) < \gamma$, and $F(k) = \gamma$, it is sufficient to prove that there is only one point for $k' = k$ where F takes a value of γ, with lower values elsewhere (Fig. 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.

References

[1] S. Boucheron, Théorie de l'apprentissage, de l'approche formelle aux enjeux cognitifs. Hermes, 1992.
[2] W.W. Cohen, Text categorization and relational learning, in: Proc. 12th Internat. Conf. on Machine Learning, 1995, pp. 124–132.
[3] P. Domingos, MetaCost: a general method for making classifiers cost-sensitive, in: Proc. 5th Internat. Conf. on Knowledge Discovery in Databases, 1999, pp. 155–164.
[4] S. Dzeroski, S. Muggleton, S. Russel, Pac-learning of determinate logic programs, in: Proc. 5th Internat. Conf. on Computational Learning Theory, 1992, pp. 128–137.
[5] J.P. Egan, Signal detection theory and ROC analyses, Academic Press, New York, 1975.
[6] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Statist. 28 (2000) 337–374.
[7] M. Garey, D. Johnson, Computers and Intractability, a guide to the theory of NP-Completeness, Bell Telephone Laboratories, 1979.
[8] J. Håstad, Clique is hard to approximate within $n^{1-\varepsilon}$, in: Proc. 37th IEEE Symp. on the Foundations of Computer Science, 1996, pp. 627–636.
[9] K.-U. Höffgen, H. Simon, Robust trainability of single neurons, in: Proc. 5th Internat. Conf. on Computational Learning Theory, 1992.

[10] R. Holte, Very simple classification rules perform well on most commonly used datasets, Mach. Learning 11 (1993) 63–91.
[11] P. Jappy, R. Nock, O. Gascuel, Negative robust learning results for horn clause programs, in: Proc. 13th Internat. Conf. on Machine Learning, Morgan Kauffman, Los Altos, CA, 1996, pp. 258–265.
[12] M. Kearns, M. Li, L. Pitt, L. Valiant, On the learnability of boolean formulae, in: Proc. 19th ACM Symp. on the Theory of Computing, 1987, pp. 285–295.
[13] J.-U. Kietz, Some lower bounds for the computational complexity of inductive logic programming, in: Proc. 8th European Conf. on Machine Learning, 1993, pp. 115–123.
[14] J. Kietz, S. Dzeroski, Inductive logic programming and learnability, Sigart Bull. 5 (1994) 22–32.
[15] A.R. Klivans, R. Servedio, Learning DNF in time $2^{\tilde{o}(n^{\frac{1}{3}})}$, in: Proc. 33th ACM Symp. on the Theory of Computing, 2001, pp. 258–265.
[16] M. Kukar, N. Besic, I. Kononenko, M. Auersperg, M. Robnik-Sikonja, Prognosing the survival time of the patients with anaplastic thyroid carcinoma using machine learning, in: N. Lavrac, E. Karavnou, B. Zupan (Eds.), Intelligent data analysis in medicine and pharmacology, Kluwer Academic Publishers, Dordrecht, 1997, pp. 116–129.
[17] M. Kubat, R.C. Holte, S. Matwin, Machine Learning for the detection of oil spills in satellite radar images, Mach. Learning 30 (1998) 195–215.
[18] R. Nock, O. Gascuel, On learning decision committees, in: Proc. 12th Internat. Conf. on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1995, pp. 413–420.
[19] R. Nock, P. Jappy, On the power of decision lists, in: Proc. 15th Internat. Conf. on Machine Learning, Morgan Kaufmann, Los Altos CA, 1998, pp. 413–420.
[20] F. Provost, T. Fawcett, Analysis and visualization of classifier performance : comparison under imprecise class and cost distributions, in: Proc. 3rd Internat. Conf. on Knowledge Discovery in Databases, 1997, pp. 43–48.
[21] F. Provost, T. Fawcett, Robust classification systems for imprecise environments, in: Proc. 15th National Conf. on Artificial Intelligence, 1998, pp. 706–713.
[22] F. Provost, T. Fawcett, R. Kohavi, The case against accuracy estimation for comparing induction algorithms, in: Proc. 15th Internat. Conf. on Machine Learning, 1998, pp. 445–453.
[23] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, in: Proc. 11th Internat. Conf. on Computational Learning Theory, 1998, pp. 80–91.
[24] M. Sebban, R. Nock, A hybrid filter/wrapper approach of feature selection using information theory, Pattern Recognition 35 (2002) 835–846.
[25] S. Slattery, M. Craven, Combining statistical and relational methods for learning in hypertext domains, in: Proc. 8th Internat. Conf. on Inductive Logic Programming, 1998, pp. 38–52.
[26] A. Srinivasan, Applications of ILP to problems in chemistry and biology, in: Invited talk at the 8th Internat. Conf. on Inductive Logic Programming, 1998.
[27] L.G. Valiant, A theory of the learnable, Commun. ACM 27 (1984) 1134–1142.
[28] L.G. Valiant, Learning disjunctions of conjunctions, in: Proc. 9th Internat. Joint Conf. on Artificial Intelligence, 1985, pp. 560–566.
[29] V. Vapnik, Statistical learning theory, John Wiley, New York, 1998.