Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 321 (2004) 371 – 382
Theoretical Computer Science
www.elsevier.com/locate/tcs

On domain-partitioning induction criteria: worst-case bounds for the worst-case based

Richard Nock[a,*], Frank Nielsen[b]

[a] *Grimaag-Département Scientifique Interfacultaire, Université des Antilles-Guyane, Campus de Schoelcher, BP 7209, 97275 Schoelcher, Martinique, France*
[b] *SONY CS Labs Inc., 3-14-13 Higashi Gotanda, Shinagawa-Ku, Tokyo 141-0022, Japan*

Received 26 June 2003; received in revised form 23 March 2004; accepted 13 May 2004
Communicated by O. Watanabe

Abstract

One of the most popular induction scheme for supervised learning is also one of the oldest. It builds a classifier in a top-down fashion, following the minimization of a so-called index criterion. While numerous papers have reported experiments on this scheme, little has been known on its theoretical aspect until recent works on decision trees and branching programs using a powerful classification tool: boosting.

In this paper, we look at this problem from a worst-case *computational* (rather than informational) standpoint. Our conclusions for the ranking of these indexes' minimization follow almost exactly that of boosting (with matching upper and lowerbounds), and provide extensions to more classes of Boolean formulas such as decision lists, multilinear polynomials and symmetric functions. Our results also exhibit a strong worst-case for the induction scheme, as we build particularly hard samples for which the replacement of most index criteria, or the class of concept representation, even when producing the same ranking as boosting does for the indexes, makes no difference at all for the concept induced. This is clearly not a limit of previous analyses, but a consequence of the induction scheme.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Classifier induction; Computational complexity; Index criteria

1. Introduction

Classifier induction refers to a process which builds, from labeled examples, a formula whose inputs are observations, and whose outputs are classes [15]. The set

* Corresponding author. Tel.: +33-596-72-74-24; fax: +33-596-72-73-62.
E-mail addresses: richard.nock@martinique.univ-ag.fr, rnock@martinique.univ-ag.fr (R. Nock), frank.nielsen@acm.org (F. Nielsen).

of labeled examples is called a learning sample. It is drawn from a set called a *domain*, whose size is typically huge compared to the learning sample's. The goodness of fit of a formula is evaluated by the discrepancies on the classes between the concept it induces over the domain, and an unknown target concept which governs the labeling of the domain. The problem is obviously to minimize these discrepancies. Conventional approaches to this problem proceed by minimizing the *empirical risk*, i.e. the error on the learning sample. In this paper, we are concerned with the strategies adopted by some popular induction algorithms to succeed in this task.

It is hard to exaggerate in Machine Learning and Classification the influence of an induction scheme among the most popular, which builds the final classifier from scratch, in a greedy, top-down fashion. Algorithms integrating as first stage such a mechanism are numerous, and some date back to the early eighties. The most popular induce decision trees (DT) [1,13], and others induce branching programs [2], decision lists [11], multilinear polynomials [10], symmetric functions [12]. All these algorithms share another property: the criterion they minimize through their local choices is *not* the empirical error, but an upperbound known as an *index criterion*.

Apart from the choice of the concepts they induce, a prominent difference between all these algorithms is the choice of the index criterion they optimize. Since recent works on a powerful new classification tool known as boosting [4], very little was known about the true influence of the choice of index on the minimization of the error and size. In fact, recent works have shown that, from the classification standpoint, it is crucial. Kearns and Mansour [7] have proven that the efficiency of any index criterion on DT induction depends on its concavity, and a careful choice (Matsushita error) may even bring optimality from an informational standpoint. Most importantly, their results are basically sufficient conditions for the control of the error/size by the index, and as they argue, their results do not yield necessary conditions (matching bounds). Better than comparing different indexes for a single concept representation class, Mansour and McAllester [8] have compared, for a single index, the interest in switching the class. They have shown that a class with larger expressive power may lead to huge gaps in size for the same guarantees on the decrease of the empirical error. Again, these results are sufficient conditions for the gap, as they do not provide matching bounds for its *systematic* existence.

In this paper, we compare again index criteria, classes of concept representations and induction algorithms, but our tool is much different: complexity theory. We obtain a ranking according to the minimization's hardness for each index which, interestingly, follows that of boosting [7], unless some widely believed complexity assumption is false. We also observe that the index criterion which yields optimality from the informational standpoint of Kearns and Mansour [7] also yields optimality from our computational standpoint. Finally, we somewhat extend in our setting their results on DT to other classes of concept representations. This similar behavior which appears from two different theoretical standpoints is, we think, an advocacy for the theoretical efficiency of boosting and these index criteria-based induction methods.

However, a closer look at our results emphasizes original phenomena with respect to the conclusions of Mansour and McAllester [8], Kearns and Mansour [7]. Our technique relies on building a particular family of learning samples, and for each of them, for any applicable class of formulas, and any index criterion (with one notable exception), there is no impact at all in terms of size and/or error for the classifier induced. This is neither a paradox, nor a limit of boosting-type analyzes [7,8]. Rather, it is a consequence of the relationship between the index criteria and the error. This makes that one has certainly to care about the index criterion used in a DT induction algorithm from the theoretical standpoint, but she/he cannot prevent the existence of datasets for which the theoretical difference between criteria and/or formulas shall not be borne out from the induction's results.

This paper is formatted to present our results at first on DT, following Kearns and Mansour [7]. In Section 2, we present the basis of the comparison criteria on the indexes. Afterwards, Section 3 presents our family of learning samples and some characteristics which yields simplifying hypotheses on the DTs induced. Section 4 presents our results on the comparison of the index criteria, that are extended in Section 5, in particular to other classes of concept representation. Section 6 discusses our results and concludes.

2. The empirical error and its minimization with DT

Our notations closely follow Mansour and McAllester [8]. We consider a domain X of observations, and a function c from X to $\{0,1\}$, where 0 and 1 are classes: generally, 0 is called the negative class, and 1 the positive class. We suppose that each observation is given by the assignment of n Boolean description variables. We are given a set S of pairs $(x, c(x))$, also called examples, in which x is referred to as an observation. S is called a learning sample. Our objective is to find some $f : X \to \{0,1\}$ such that its empirical error on S with respect to c, $\hat{\varepsilon}(f) = |\{(x, c(x)) \in S : c(x) \neq f(x)\}|/|S|$, is minimal.

Let us first suppose that f is an ordinary DT. A DT is a recursive, domain-partitioning concept which can be seen as a tree with a root node, leaf nodes and possibly internal nodes. Except for the leaves for which it is 0, the out-degree of each node is exactly 2. Each internal node is labeled by a Boolean test $h \in H$, and each outgoing arc is labeled by one of the two truth values. Observations to be classified traverse the tree from its root through its internal nodes, following a path whose arcs correspond to the test they satisfy, until they reach a leaf whose label $\in \{0,1\}$ gives their predicted class. The most popular DT induction algorithms pick for H the set of Boolean description variables. Thus, the arcs are labeled with their projections [1,7,13]. These algorithms also share the property to be stagewise modeling procedures, fitting the tree to the data by the repetitive optimization of a so-called index criterion $\boldsymbol{I}$ whose shape is common to all. Suppose we are given a DT f which contains K internal nodes, thus, $K+1$ leaves. We suppose that for each applicable depth, the nodes of f are numbered in $\{0,1,\ldots\}$ without ambiguity in such a way that $S_{i,j}$ may define the subset of S reaching the node numbered j at depth i. In the sequel, we shall sometimes assimilate each node of f with the unique couple (i,j) to which it corresponds. We adopt the convention $S_{0,0} = S$, and for each applicable (i,j) and $b \in \{0,1\}$, $S^{b}_{i,j} = \{(x,b) \in S_{i,j}\}$, $\hat{p}_{i,j} = |S_{i,j}|/|S|$, and $\hat{q}_{i,j} = |S^{1}_{i,j}|/|S_{i,j}|$. For each internal node (i,j) and $b \in \{0,1\}$, we let $h_{i,j}$ define the Boolean test labeling the node, and $\hat{p}^b_{i,j} = |\{(x,c(x)) \in S_{i,j} : h_{i,j}(x) = b\}|/|S_{i,j}|$. Fix a function I, continuous over $[0,1]$, symmetric around $\frac{1}{2}$, concave, and such that $I(\frac{1}{2}) = 1, I(0) = I(1) = 0$. Such a function is called *permissible* in Kearns and Mansour [7]. Index criterion $\boldsymbol{I}$ is chosen as follows:

$$\boldsymbol{I}(f) = \sum_{(i,j) \text{ leaf in } f} \hat{p}_{i,j} I(\hat{q}_{i,j}). \tag{1}$$

Some possible choices for the index function I in Eq. (1) include

$$I(z) = 4z(1-z), \tag{2}$$

$$I(z) = -z\log z - (1-z)\log(1-z), \tag{3}$$

$$I(z) = 2\sqrt{z(1-z)}, \tag{4}$$

$$I(z) = 2\,\min\{z, 1-z\}. \tag{5}$$

Eq. (2) is known as Gini index [1,7], Eq. (3) codes for the binary entropy [7,13] and Eq. (5) relates to twice the local error. Eq. (4) is proportional to a geometrical average; this criterion has recently received a growing attention with the introduction of a new powerful methodology in learning/classification known as boosting [4]. However, its use dates back to the fifties, and Eq. (1) with I as in Eq. (4) is known as Matsushita error on S (provided we replace the description of each example in S by the Boolean tests of the path it follows in f) [9]. We denote as $\boldsymbol{I}_G(f)$, $\boldsymbol{I}_H(f)$, $\boldsymbol{I}_M(f)$, $\boldsymbol{I}_E(f)$ as the four expressions of $\boldsymbol{I}(f)$ in Eqs. (1), using, respectively, Eqs. (2)–(5) for I. Most importantly, since $\hat{\varepsilon}(f) \leqslant \boldsymbol{I}(f)/2$, it comes that any f with small index $\boldsymbol{I}(f)$ is guaranteed to have small empirical error.

For any non-leaf DT f, pick some internal node (i,j) labeled with some Boolean test h. Name $f\backslash h$ as the DT obtained by pruning the subtree rooted at h in f, thus replacing internal node (i,j) by a leaf. Define

$$\rho_{h,i,j}(I) = (\boldsymbol{I}(f\backslash h) - \boldsymbol{I}(f))/\boldsymbol{I}(f\backslash h). \tag{6}$$

This criterion is the cornerstone of the analysis of Kearns and Mansour [7]. It quantifies some sort of relative "potential decrease" between $f\backslash h$ and f. For any DT f, consider a sequence of distinct internal nodes $(i_1,j_1),(i_2,j_2),\ldots,(i_k,j_k)$ with $k \leqslant K$ (recall that K is the number of internal nodes of f), such that $f\backslash(i_k,j_k)$ is a leaf DT, and $\forall 1 \leqslant l < l' \leqslant k$, $(i_{l'},j_{l'})$ is an internal node in $f\backslash(i_l,j_l)$. Informally, such a sequence brings a way to prune f by removing successive internal nodes, and we refer to it as a "valid sequence" of internal nodes. For any valid sequence, we have the key equality

$$\boldsymbol{I}(f) = I(\hat{q}_{0,0}) \prod_{l=1}^{k} (1 - \rho_{h,i_l,j_l}). \tag{7}$$

In this equation, the Boolean test h in ρ_{h,i_l,j_l} refers without ambiguity to the one labeling the internal node (i_l,j_l). Since $I(\hat{q}_{0,0})$ is independent of the algorithm used to induce f, efficient induction algorithms for DT should impact on the fast maximization of the ρ's over some valid sequence of internal nodes. This is precisely what the most popular induction algorithms for DT do, with the same strategy: the top-down (greedy) induction of a large tree following the same routine **TD**:

grow a DT f from a single-leaf DT (the root), repeatedly replacing leaves by internal nodes (each with two new leaves).

Each leaf label is chosen as the majority class among the examples reaching the leaf. The algorithms typically stop whenever some condition on the empirical error of f, or on its size, is met. Afterwards, a second stage prunes the DT f with the objective to statistically limit its *generalization* error [1,3], but this is out of the scope of this paper. Suppose that the current f was built from the replacement of some leaf (i,j) in $f\backslash h$ by an internal node labeled with some test $h \in H$. h is chosen in **TD** as follows. Define the two new children leaves of (i,j) as $(i+1, j_0)$ (for those examples $\in S_{i,j}$ for which $h(x)=0$) and $(i+1, j_1)$ (for those examples $\in S_{i,j}$ for which $h(x)=1$). We have

$$\boldsymbol{I}(f\backslash h) - \boldsymbol{I}(f) = \hat{p}_{i,j}[I(\hat{q}_{i,j}) - \hat{p}^0_{i,j} I(\hat{q}_{i+1,j_0}) - \hat{p}^1_{i,j} I(\hat{q}_{i+1,j_1})]. \tag{8}$$

If we define the *local* decrease of $\boldsymbol{I}$ as

$$\Delta(S_{i,j}, h) = I(\hat{q}_{i,j}) - \hat{p}^0_{i,j} I(\hat{q}_{i+1,j_0}) - \hat{p}^1_{i,j} I(\hat{q}_{i+1,j_1}) \tag{9}$$

then the strategy to choose h in **TD** is [7]

$$h = \arg\max_{h'} \Delta(S_{i,j}, h'). \tag{10}$$

Since $f\backslash h$ is fixed, we see that $\arg\max_{h'} \Delta(S_{i,j}, h') = \arg\max_{h'} \boldsymbol{I}(f\backslash h') - \boldsymbol{I}(f) = \arg\max_{h'} (\boldsymbol{I}(f\backslash h') - \boldsymbol{I}(f))/\boldsymbol{I}(f\backslash h') = \arg\max_{h'} \rho_{h',i,j}(I)$, and **TD** is a stepwise maximization of the ρ's to minimize $\boldsymbol{I}$ in Eq. (7), whose valid sequence of internal nodes is the list of internal nodes from the last one created to the first one (replacing the root leaf).

More generally, because index criteria are direct upperbounds for the error and due to Eq. (7), the study of lowerbounds on $\rho_{h,i,j}(I)$ is of significant importance; in the sequel, we study the minimal guarantee on $\rho_{h,i,j}(I)$ that may bring some (possibly randomized) algorithm $\mathbf{A} \in \mathcal{A}$, where $\mathcal{A}$ is the set of all efficient (polynomial-time) induction algorithms for DT. To this extent, for any f induced by some $\mathbf{A}$, we let $\rho^*_f(I) = \min_{i,j} \rho_{h,i,j}(I)$ ($\mathbf{A}$ is absent from the notation but it should be clear from context), and we let $\rho^*_{\mathbf{A}}(I)$ denote the minimal value of $\rho^*_f(I)$ for some f which is output by $\mathbf{A}$ on some S, where S belongs to a particular family of samples which we now define.

3. Hard samples, simple trees

We create particular learning samples from "Set-Cover" instances [3]. The "Set-Cover" instance contains a set E of elements, and a collection $C = \{C_1, C_2, \ldots, C_{|C|}\}$ of subsets of E. The objective is to find a cover of E, i.e. a subset of C whose union of elements is E, with the least number of elements from C. It is well known that this problem is hard to solve or even approximate, as finding coverings whose size is no more than the optimum times $(1-\delta)\ln|E|$, for any constant $0<\delta<1$, is intractable unless NP has slightly superpolynomial time algorithms, that is, unless $NP \subseteq DTIME[N^{\log\log N}]$ [3]. We build a learning sample S which contains $|E|+1$ examples, with only one negative example

- there are $n=|C|$ description variables, in one-to-one correspondence with the elements of C;
- the negative example, $(x^-, 0)$, contains only assignments to 0 of its n description variables;
- the positive examples are in one-to-one correspondence with the elements of E. Positive example $(x_j^+, 1)$ has assignments to 1 of its description variables corresponding to those elements of C containing the jth element of E. The assignments of all other description variables are 0.

This reduction is well-known in learning theory [5,10]. The next Lemmata show two properties on the proof of Feige [3], which relies on a reduction from instances of a 3SAT variant, 3SAT5 (each variable appears exactly in five clauses).

Lemma 1. *We can assume without loss of generality that* $\forall 1 \leqslant i \leqslant |C|$, $|C_i| = \Theta(|E|/(k'Q))$, *where* k', Q *are reduction-dependent parameters.*

Proof. We follow the proof of Feige [3]. The proof proceeds by building R partition systems. Informally, each partition system is built on a separate set of m elements, onto which a collection of L distinct partitions is built at random: for each of the m elements, we decide at random where to put this element into one of the subset of each of the L partitions. Each partition contains k' subsets of the m elements. The whole number of elements is $|E|=mR$, and the final, expected size of some C_i is $mR/(k'Q)$ (each C_i is the reunion of R/Q different subsets of different partition systems). Since it does not rely on similarities in the sizes of the elements of C, the proof [3] can be modified without significant complexity penalty, to enforce that the size of each C_i be not too far from its average. This ends the proof of Lemma 1. □

We denote c^* as the minimum "Set Cover" solution.

Lemma 2 (Feige [3]). *Whenever the* 3*SAT*5 *instance is satisfiable*, $c^* = k'Q$ *and the sets in the optimal solution define a partition of E.*

The family of samples S we build makes that only samples $S_{i,j}$ on the all-0 path from the root of f may contain both positive examples and the negative example. All the other subsets of S are pure in that they only contain positive examples, so they do not participate to $\boldsymbol{I}(f)$ and do not change the concept with respect to S. We thus simplify the notations of Section 2, and consider in f only those nodes from the all-0 path, with notations detailed in Fig. 1. We also suppose that the internal nodes of the

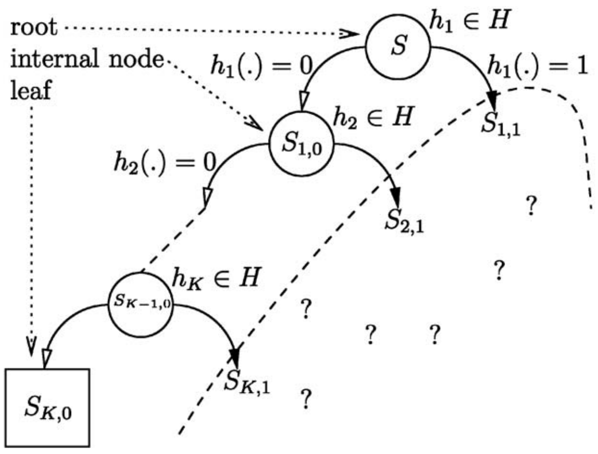

Fig. 1. The nodes of f that interest us and the notations we adopt for the subsets of S throughout the DT. Black (resp. empty) arrows corresponds to arcs labeled with the true (1) value for h (resp. false (0)).

all-0 path are not pure, since otherwise we could prune them and get the same concept with respect to S. Remark also that there is no need to compute $\rho_{h,i,j}$ for $j>0$, so we simplify its notation with $\rho_{h,i}$ in Eq. (6); suppose we replace $(K,0)$ by an internal node labeled with some $h_{K+1} \in H$ in formula f of Fig. 1. Then we have

$$\rho_{h_{K+1},K}(I) = (\boldsymbol{I}(f) - \boldsymbol{I}(f \cup h_{K+1}))/\boldsymbol{I}(f) \tag{11}$$

$$= (\hat{p}_{K,0}\Delta(S_{K,0}, h_{K+1}))/(\hat{p}_{K,0}I(\hat{q}_{K,0})) \tag{12}$$

$$= \Delta(S_{K,0}, h_{K+1})/I(\hat{q}_{K,0}). \tag{13}$$

4. Main results

We now show a universal complexity-theoretic upperbound on $\rho^*_.(.)$.

Theorem 3. *Unless $NP \subseteq DTIME[N^{\log\log N}]$, for any index function I, any $\mathbf{A} \in \mathcal{A}$, $\rho^*_{\mathbf{A}}(I) = O(1/c^*)$.*

Proof. Suppose that $\rho^*_f(I) = \Omega(1/c^*)$ for some f output by $\mathcal{A}$. For any internal node $(k,0)$ of f, we denote $f(k)$ as the DT obtained when pruning $(k,0)$ and making it replaced by a leaf labeled by its majority class. We show that, up to a constant factor, an optimal approximation of "Set Cover" is obtained. We have $\forall 0 \leqslant k \leqslant K-1$, $\boldsymbol{I}(f(k+1)) = \hat{p}_{k+1,0}I(\hat{q}_{k+1,0}) = \hat{p}_{k,0}\hat{p}^0_{k,0}I(\hat{q}_{k+1,0}) = \hat{p}_{k,0}(1 - \rho_{h_{k+1},k}(I))\boldsymbol{I}(f(k)) \leqslant (1 - \rho_{h_{k+1},k}(I))\boldsymbol{I}(f(k))$. We get $\boldsymbol{I}(f(k)) \leqslant (1 - \rho^*_f(I))^k \boldsymbol{I}(f(0)) \leqslant (1 - \rho^*_f(I))^k \leqslant \exp(-k\rho^*_f(I))$. For any node for which $|S_{k+1,0}| \leqslant 2$ (exactly one positive and one negative example belong to $S_{k+1,0}$), we have $\boldsymbol{I}(f(k+2)) = 0$, since otherwise we would have $\rho^*_f(I)=0\neq\Omega(1/c^*)$. In that case, **A** shall have selected k hypotheses from H to label those internal nodes, each of which corresponds to an element of C whose union brings a cover of E. Furthermore, if $\hat{p}_{k+1,0}<2/(|E|+1)$ and $I(\hat{q}_{k+1,0})<1/(|E|+1)$, then $\hat{\varepsilon}(f(k+1))=0$. We get therefore a cover of size k as soon as $\boldsymbol{I}(f(k))<2/(|E|+1)^2$, a sufficient condition for which is obtained when $k=\Omega((1/\rho^*_f(I))\ln|E|)$. $\rho^*_f(I)=\Omega(1/c^*)$ yields an approximation to "Set Cover" up to ratio O(ln $|E|$) which is, up to the constant hidden in the "O" notation, optimal [3]. It is thus the best guarantee on $\rho^*_f(I)$ for any index function I and any $\mathbf{A}\in\mathcal{A}$. □

The following theorem shows that the worst-case bound of Theorem 3 is achieved by algorithm $\mathbf{TD}\in\mathcal{A}$ with Matsushita error.

Theorem 4. $\rho^*_{\mathbf{TD}}(I_M)=\Omega(1/c^*)$.

Proof. Consider f output by **TD**. Fix $0\leqslant k\leqslant K-1$. We have $\forall h_{k+1}\in H$

$$\begin{aligned}\Delta_M(S_{k,0},h_{k+1}) &= I_M(\hat{q}_{k,0})-\hat{p}^0_{k,0}I_M(\hat{q}_{k+1,0})-\hat{p}^1_{k,0}I_M(1)\\ &= \frac{2\sqrt{|S^1_{k,0}|}}{|S^1_{k,0}|+1}-\frac{|S^1_{k+1,0}|+1}{|S^1_{k,0}|+1}\times\frac{2\sqrt{|S^1_{k+1,0}|}}{|S^1_{k+1,0}|+1}\\ &= \frac{2\left(\sqrt{|S^1_{k,0}|}-\sqrt{|S^1_{k+1,0}|}\right)}{|S^1_{k,0}|+1}.\end{aligned}$$

Therefore, $\rho_{h_{k+1},k}(I_M)=\Delta_M(S_{k,0},h_{k+1})/I_M(\hat{q}_{k,0})=1-\sqrt{|S^1_{k+1,0}|/|S^1_{k,0}|}$. Now note that **TD** picks $h_{k+1}=\arg\max_{h\in H}\Delta_M(S_{k,0},h)$, and thus guarantees

$$|S^1_{k+1,0}|\leqslant|S^1_{k,0}|(1-(1/c^*)), \tag{14}$$

since otherwise there could not be a cover of size c^* of E. We get $\Delta_M(S_{k,0},h_{k+1})/I_M(\hat{q}_{k,0})\geqslant 1-\sqrt{1-(1/c^*)}$, and since $\sqrt{1-(1/c^*)}\leqslant 1-(1/(2c^*))$, we finally obtain $\rho_{h_{k+1},k}(I_M)\geqslant(1/(2c^*))$. Since inequality (14) holds regardless of the depth, we have $\rho^*_f(I_M)\geqslant(1/(2c^*))$ and thus $\rho^*(I_M)=\Omega(1/c^*)$. □

Now, we skip to entropy-index based induction algorithms $\in\mathcal{A}$. We show that they bring minimal guarantees over the maximization of Eq. (6) that do not match those for Matsushita index. This is due to a concavity property of the index criterion, a crucial property in the results of Kearns and Mansour [7]. We also show that algorithm **TD** brings the minimal guarantee.

Theorem 5. $\rho^*_{\mathbf{TD}}(I_H)=\Omega(1/(c^*\log|E|))$.

Proof. Consider some f output by **TD** and any depth $0 \leqslant k \leqslant K-1$. Basic arithmetics yield with $g(x) = \log(1+x) + x\log(1+(1/x))$:

$$\rho_{h_{k+1},k}(I_H) = 1 - \frac{(1+|S^1_{k+1,0}|)\log(1+|S^1_{k+1,0}|) - |S^1_{k+1,0}|\log|S^1_{k+1,0}|}{(1+|S^1_{k,0}|)\log(1+|S^1_{k,0}|) - |S^1_{k,0}|\log|S^1_{k,0}|}$$

$$= 1 - \frac{g(|S^1_{k+1,0}|)}{g(|S^1_{k,0}|)}.$$

Since g is strictly increasing over $\mathbb{R}^{+,*}$, concave ($g''(x) = -1/(x(1+x))$) and $|S^1_{k+1,0}| < |S^1_{k,0}|$, we have

$$g(|S^1_{k+1,0}|) < (|S^1_{k+1,0}| - |S^1_{k,0}|)g'(|S^1_{k,0}|) + g(|S^1_{k,0}|). \tag{15}$$

The proof is now the same as that of Theorem 4, i.e. we look at the reduction index led by picking $h^*_{k+1} = \arg\max_{h\in H} \Delta_H(S_{k,0}, h)$ in **TD**. Using inequalities (14) and (15), we easily get $\rho_{h_{k+1},k}(I_H) \geqslant (|S^1_{k,0}| - |S^1_{k+1,0}|)g'(|S^1_{k,0}|)/g(|S^1_{k,0}|) \geqslant |S^1_{k,0}|g'(|S^1_{k,0}|)/(c^* g(|S^1_{k,0}|))$, and thus

$$\rho_{h_{k+1},k}(I_H) \geqslant \frac{1}{c^*}\left[1 - \frac{\log(1+|S^1_{k,0}|)}{g(|S^1_{k,0}|)}\right].$$

We have $|S^1_{k,0}| \geqslant 1$, since otherwise $|S^1_{k,0}| = 0$, thus it would label a leaf, and $S^1_{k+1,0}$ would not exist. Since $\log(1+(1/x)) \geqslant 1/x - (1/(2x^2))(x>0)$, we get $g(x) \geqslant \log(1+x) + 1 - (1/2x)$, and therefore $-\log(1+x)/g(x) \geqslant -2x\log(1+x)/(2x\log(1+x) + 2x - 1)$. Furthermore, $(2x-1)/(2x\log(1+x) + 2x - 1) \geqslant 1/(4\log(1+x))$ $(x \geqslant 1)$. Putting these altogether, we obtain

$$\rho_{h_{k+1},k}(I_H) \geqslant \frac{1}{4c^*\log(1+|S^1_{k,0}|)}$$

$$= \Omega(1/(c^*\log|E|)).$$

We get $\rho^*_f(I_H) = \Omega(1/(c^*\log|E|))$. This yields the statement of Theorem 5. □

Theorem 6. *For any index function I, any $\mathbf{A} \in \mathcal{A}$, any f output by $\mathcal{A}$, $\rho^*_f(I_H) = O(1/(c^*\log|E|))$.*

Proof. Consider any formula f output by $\mathcal{A}$. We use the notations of Theorem 5. Because g is strictly increasing and concave, we have $g(|S^1_{k+1,0}|) \geqslant (|S^1_{k+1,0}| - |S^1_{k,0}|)$ $g'(|S^1_{k+1,0}|) + g(|S^1_{k,0}|)$. This leads to

$$\rho_{h_{k+1},k}(I_H) \leqslant (|S^1_{k,0}| - |S^1_{k+1,0}|)\frac{g'(|S^1_{k+1,0}|)}{g(|S^1_{k,0}|)}.$$

We would like $\rho_{h_{k+1},k}(I_H) = O(1/(c^*\log|E|))$, and thus $g(|S^1_{k,0}|)/(|S^1_{k,0}| - |S^1_{k+1,0}|$ $(g'(|S^1_{k+1,0}|))) = \Omega(c^*\log|E|)$ for some suitable k. Since $g(|S^1_{k,0}|) \geqslant \log(1 + |S^1_{k,0}|)$

and $g'(|S^1_{k+1,0}|) \leqslant 1/|S^1_{k+1,0}|$, a sufficient condition is to have

$$|S^1_{k+1,0}|\log(1+|S^1_{k,0}|) = \Omega((|S^1_{k,0}| - |S^1_{k+1,0}|)c^* \log|E|). \tag{16}$$

Consider the first step of the algorithm **A**, when $k=0$, $S^1_{0,0}=E$, and fix $|S^1_{1,0}|=|E|-\gamma$. Eq. (16) is then

$$\log(1+|E|) = \Omega\left(\frac{\gamma c^*}{|E|-\gamma}\log|E|\right). \tag{17}$$

From Lemma 1, a sufficient condition for Eq. (17) to hold is $c^* = \mathrm{O}(k'Q)$, which is the case under Lemma 2. This ends the proof of Theorem 6. □

Now, we analyze the case of Gini-based induction algorithms $\in \mathcal{A}$. We show that they bring minimal guarantees over the maximization of Eq. (6) that do not match those for the Entropy index. Again, this appears to be due to the concavity of the criterion, and we show that algorithm **TD** realizes the minimal guarantee.

Theorem 7. $\rho^*_{\mathbf{TD}}(I_G) = \Omega(1/(c^*|E|))$.

Proof. The strategy is the same as those of Theorems 4 and 5, so we only sketch the principal steps. We have from Eqs. (2) and (9)

$$\rho_{h_{k+1},k}(I_G) = (|S^1_{k,0}| - |S^1_{k+1,0}|)/(|S^1_{k,0}|(1+|S^1_{k+1,0}|)). \tag{18}$$

Because of Eq. (14), we get $\rho_{h_{k+1},k}(I_G) \geqslant 1/(c^*(1+|S^1_{k+1,0}|)) = \Omega(1/(|E|c^*))$. This ends the proof of Theorem 7. □

Theorem 8. *For any index function I, any $\mathbf{A}\in\mathcal{A}$, any f output by* **A**, $\rho^*_f(I_G) = \mathrm{O}(1/(|E|c^*))$.

Proof. The strategy follows that of Theorem 6. Fix $k=0$, and $|S^1_{0,0}| - |S^1_{1,0}| = \gamma$. Eq. (18) becomes $\rho_{h_1,0}(I_G) \leqslant \gamma/(|E|(|E|-\gamma))$. This function is an increasing function of γ. If we pick $\gamma = \mathrm{O}(|E|/(k'Q))$ (Lemma 1), then a sufficient condition for Theorem 8 to hold is to have $1/(|E|(kQ-1)) = \mathrm{O}(1/(|E|c^*))$, thus $c^* = \mathrm{O}(k'Q)$, which is the case under Lemma 2. This ends the proof of Theorem 8. □

We end this section with our last replacement of I by Eq. (5), which yields $\boldsymbol{I}_E(f) = 2\hat{\varepsilon}(f)$. According to Kearns and Mansour [7], it is an “especially *poor* choice” for $\boldsymbol{I}$. In our setting, we show that it is in fact the poorest of all.

Theorem 9. *For any $\mathbf{A}\in\mathcal{A}$, any f output by* **A**, *and any internal node $(k,0)$ of f*, $\rho_{h_{k+1},k}(I_E) = 0$.

(The proof is straightforward and omitted here.)

With this last criterion, everything is like if any induction algorithm $\mathbf{A}\in\mathcal{A}$ were making blind choices for the internal nodes’ labels. Notice that there is a connection with **TD** and Chvátal's greedy "Set-Cover" approximation algorithm, which manages an optimal approximation of the problem (up to low order terms). The criterion it minimizes to pick h_k is not the overall error $\boldsymbol{I}_E(f)$, but the local error, $I_E(\hat{q}_{k,0})$. It is interesting to notice that Matsushita error, which is a global criterion, leads to the same kind of optimal approximation for "Set Cover".

5. Extension of the results

There are two ways to extend the results of the preceding section. The first one concerns classes H bigger than the set of description variables and the second concerns classes of domain partitioning concepts different from DT. It is easy to see that one can replace H by larger classes of size polynomial in the number of description variables, such as conjunctions of description variables' assignments of bounded constant size, while keeping all results. The argument consists in mapping all formulas in H to a set of new description variables from which we make exactly the same reduction as above. We can also replace the class DT by many other classes, whose elements would represent on our family of samples S the same concepts as the simple DTs of Section 3. The first example is a generalization of DT: branching programs (BPs); BPs are direct acyclic graphs (DAGs) relaxing the constraint that internal nodes must have in-degree 1. Other examples include monomials (conjunction of description variables' assignments), disjunctive normal form formulas (DNF, disjunction of monomials), decision lists (DL, ordered sets of if–then rules) [11,14], multilinear polynomials [10], and symmetric functions (SF, formulas invariant upon permutation of the input bits) [6,12]. All these classes share the additional commonpoint to have **TD**-like induction algorithms.

6. Discussion and conclusion

The ranking of the four index criteria follows exactly that of Kearns and Mansour [7]. The best criterion is $\boldsymbol{I}_M$, which matches the complexity-theoretic bound, while the worst is $\boldsymbol{I}_E$; the other two, Gini and the entropy, are in between, with the entropy being the best of the two. The fact that our conclusions drawn from complexity theory follow those of Kearns and Mansour [7] drawn from information theory is one more advocacy for the strength of boosting, and advocates for the importance of choosing I as accurately as possible from a general theoretical standpoint. **TD** also appears to be the best induction algorithm among all polynomial-time approaches in our framework (either top down or not).

However, it is remarkable that **TD** would induce the *same* concept on every fixed "Set Cover" instance, whichever index criterion it uses (different from $\boldsymbol{I}_E$), and for any applicable class of concept representation, thus contrasting with [7,8]. This paradox is only apparent and comes from the fact that any index criterion minimization approach is worst-case based from a classification standpoint (recall that the index upperbounds the error, see Section 2). Thus, different index choices (or formulas from different classes) may not always yield the gaps predicted by theory on the respective formulas induced. This last remark tends to show that the search for matching lowerbounds [7] may be bound to failure if we want them to hold regardless of some properties on the sample S.

Acknowledgements

The authors would like to thank the reviewers for their insightful comments that helped to substantially improve the quality of this paper.

References

[1] L. Breiman, J.H. Freidman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, Belmonk, CA, 1984.
[2] T. Elomaa, M. Kääriäinen, On the practice of branching program boosting, in: Proc. 16th European Conf. on Machine Learning, 2001.
[3] U. Feige, A threshold of ln n for approximating set cover, J. ACM 45 (4) (1998) 634–652.
[4] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. System Sci. 55 (1997) 119–139.
[5] T. Hancock, T. Jiang, M. Li, J. Tromp, On approximating decision lists and trees, Inform. and Comput. 126 (1996) 114–122.
[6] M. Kearns, M. Li, Learning in the presence of malicious errors, SIAM J. Comput. 22 (1993) 807–837.
[7] M. Kearns, Y. Mansour, On the boosting ability of top-down decision tree learning algorithms, J. Comput. System Sci. 58 (1999) 109–128.
[8] Y. Mansour, D. McAllester, Boosting using branching programs, J. Comput. System Sci. 64 (2002) 103–112.
[9] K. Matsushita, Decision rule, based on distance, for the classification problem, Ann. Inst. Statist. Math. 8 (1956) 67–77.
[10] R. Nock, Inducing interpretable Voting classifiers without trading accuracy for simplicity: theoretical results, approximation algorithms and experiments, Artificial Intelligence Res. J. 17 (2002) 137–170.
[11] R. Nock, P. Jappy, On the power of decision lists, in: Proc. 15th Internat. Conf. on Machine Learning. Morgan Kaufmann, Los Altos, CA, 1998.
[12] R. Nock, P. Lefaucheur, A robust boosting algorithm, in: Proc. 17th European Conf. on Machine Learning, Vol. 2430, Lecture Notes in Artificial Intelligence, Springer, New York, 2002.
[13] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA, 1993.
[14] R. Rivest, Learning decision lists, Mach. Learning 2 (1987) 229–246.
[15] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.