Bregman Divergences and Surrogates for Learning

Richard Nock and Frank Nielsen, *Member*, *IEEE*

Abstract—Bartlett et al. (2006) recently proved that a ground condition for surrogates, classification calibration, ties up their consistent minimization to that of the classification risk, and left as an important problem the algorithmic questions about their minimization. In this paper, we address this problem for a wide set which lies at the intersection of classification calibrated surrogates and those of Murata et al. (2004). This set coincides with those satisfying three common assumptions about surrogates. Equivalent expressions for the members—sometimes well known—follow for convex and concave surrogates, frequently used in the induction of linear separators and decision trees. Most notably, they share remarkable algorithmic features: for each of these two types of classifiers, we give a minimization algorithm provably converging to the minimum of any such surrogate. While seemingly different, we show that these algorithms are offshoots of the same "master" algorithm. This provides a new and broad unified account of different popular algorithms, including additive regression with the squared loss, the logistic loss, and the top-down induction performed in CART, C4.5. Moreover, we show that the induction enjoys the most popular boosting features, regardless of the surrogate. Experiments are provided on 40 readily available domains.

Index Terms—Ensemble learning, boosting, Bregman divergences, linear separators, decision trees.

1 Introduction

Supervised learning is the problem that consists of finding a functional link between observations and classes that takes the form of a classifier, on the sole basis of a random set of examples that usually provides only few of these links. A very active supervised learning trend has been flourishing over the last decade: It studies functions known as *surrogates*—upper bounds of the empirical risk, generally with particular convexity properties—whose minimization remarkably impacts on empirical/true risks minimization [1], [2], [3]. Surrogates play fundamental roles in some of the most successful supervised learning algorithms, including AdaBoost, additive logistic regression, decision tree induction, Support Vector Machines [4], [5], [3], [6]. As their popularity has been rapidly spreading, authors have begun to stress the need to set in order the huge set of surrogates, and better understand their properties. Statistical consistency properties have been shown for a wide set containing most of the surrogates relevant to learning, *classification calibrated surrogates* [1]; other important properties, like the algorithmic questions about minimization, have been explicitly left as important problems to settle [1]. A relevant contribution on this side came earlier from Murata et al. [7], who proved mild convergence results on an algorithm inducing linear separators and working on a large class of convex surrogates, not necessarily classification calibrated; Murata et al. [7] also left as an important problem the necessity to fully solve this algorithmic question, such as by providing convergence rates.

In this paper, we address and solve this problem for all surrogates that satisfy three of the most common assumptions about surrogates in supervised learning: lower boundedness, symmetries in the cost matrix, and compliance with proper scoring rules [8]. We define such surrogates as *permissible*; the corresponding losses belong to a subset of Bregman divergences that we fully characterize. This set resembles classical surrogates (convex or concave), and provides a unified view of estimation and confidence-rated prediction [5] via a particular link with the exponential families of distributions. As we show, it is quite remarkable that all satisfy the pointwise form of Fisher consistency of classification calibrated surrogates [1] and the convexity property of $\mathcal{U}$-Boost surrogates [7] so that our results are finally relevant to both sets of surrogates.

Our algorithmic contribution consists of providing two provably universal minimization algorithms, for linear separators and decision trees, that provably converge to the optimum of any permissible surrogate. There is more, as they enjoy popular features of boosting algorithms, including guaranteed convergence rates under very weak assumptions. Apart from being one more advocacy for the computational supremacy of surrogates, these two algorithms and their analysis manage to unite popular members for the induction of linear separators and decision trees as instances of the same "master" algorithm, thus capturing the main features of both the surrogates and the classifiers.

Section 2 gives definitions and Section 3 presents permissible surrogates and their properties. Section 4

- *R. Nock is with the Université Antilles-Guyane, CEREGMIA—UFR Droit et Sciences Economiques, Campus de Schoelcher, B.P. 7209, 97275 Schoelcher Cedex, Martinique, France. E-mail: rnock@martinique.univ-ag.fr.*
- *F. Nielsen is with the Laboratoire d'Informatique (LIX), Ecole Polytechnique, 91128 Palaiseau Cedex, France. E-mail: nielsen@lix.polytechnique.fr.*

Manuscript received 16 Aug. 2007; revised 27 June 2008; accepted 5 Sept. 2008; published online 10 Sept. 2008.
Recommended for acceptance by A. Smola.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-08-0512.
Digital Object Identifier no. 10.1109/TPAMI.2008.225.

presents and proves the corresponding minimization algorithms. Section 5 provides and discusses experiments, and Section 6 concludes.

2 Definitions

Unless otherwise stated, bold-faced variables like $\boldsymbol{w}$ denote vectors (components are w_i, $i = 1, 2, \ldots$), calligraphic upper cases like $\mathcal{S}$ denote sets, and blackboard faces like $\mathbb{O}$ denote subsets of $\mathbb{R}$, the set of real numbers. We let set $\mathcal{O}$ denote a *domain* ($\mathbb{R}^n$, $[0,1]^n$, etc., where n is the number of description variables), whose elements are *observations*. An *example* is an ordered pair $(\boldsymbol{o}, c) \in \mathcal{O} \times \{c^-, c^+\}$, where $\{c^-, c^+\}$ denotes the set of classes (or *labels*) and c^+ (respectively, c^-) is the *positive* class (respectively, *negative* class). Classes can take on any values whose semantic is related to the domain (e.g., good/bad, small/large, etc.). Rather than leaving this set as is, people usually carry out an abstraction of classes that fits to theoretical studies, by a bijective mapping to one of two other sets:

$$c \in \{c^-, c^+\} \rightleftharpoons y^* \in \{-1, +1\} \leftarrow y \in \{0, 1\}.$$

The convention is $c^+ \rightleftharpoons +1 \rightleftharpoons 1$ and $c^- \rightleftharpoons -1 \rightleftharpoons 0$. We thus have three distinct notations for an example: $(\boldsymbol{o}, c)$, $(\boldsymbol{o}, y^*)$, $(\boldsymbol{o}, y)$ that shall be used without ambiguity. We suppose given a set of m examples, $\mathcal{S} = \{(\boldsymbol{o}_i, c_i), i = 1, 2, \ldots, m\}$.

Our objective is to build a *classifier* H, which can either be a function $H : \mathcal{O} \to \mathbb{O} \subseteq \mathbb{R}$ (hereafter, $\mathbb{O}$ is assumed to be symmetric with respect to 0) or a function $H : \mathcal{O} \to [0,1]$. Following a convention of [9], we can compute to which extent the outputs of H and the labels in $\mathcal{S}$ disagree, $\varepsilon(\mathcal{S}, H)$, by summing over all examples a *loss* ℓ which quantifies pointwise disagreements:

$$\varepsilon(\mathcal{S}, H) \doteq \sum_i \ell(c_i, H(\boldsymbol{o}_i)). \tag{1}$$

The smaller $\varepsilon(\mathcal{S}, H)$, the better H. Wherever needed for a clear distinction of the output of H, we put in index to ℓ and ε an indication of its image ($\mathbb{R}$, meaning it is actually some $\mathbb{O} \subseteq \mathbb{R}$, or $[0,1]$). Sometimes, we also put in exponent an indication of the loss name. For example, we let $\ell^{0/1}(c, H)$ denote the *0/1 loss*, which may be defined in two ways depending on $\mathrm{im}(H)$:

$$\ell^{0/1}_{\mathbb{R}}(y^*, H) \doteq 1_{y^* \neq \sigma \circ H}, \quad \text{if } \mathrm{im}(H) = \mathbb{O},$$
$$\ell^{0/1}_{[0,1]}(y, H) \doteq 1_{y \neq \tau \circ H}, \quad \text{if } \mathrm{im}(H) = [0,1].$$

Here, 1_π is the indicator variable that takes value 1 iff predicate π is **true**, and 0 otherwise. Furthermore, $\sigma : \mathbb{R} \to \{-1, +1\}$ is $+1$ iff $x \geq 0$, and -1 otherwise. Finally, $\tau : [0,1] \to \{0,1\}$ is 1 iff $x \geq 1/2$ and 0 otherwise.

Both losses $\ell_{\mathbb{R}}$ and $\ell_{[0,1]}$ are defined simultaneously via popular *matching* transforms on H, such as the *logit* transform [5]:

$$\mathrm{logit}(p) \doteq \log \frac{p}{1-p}, \forall p \in [0,1]. \tag{2}$$

We have indeed

$$\ell^{0/1}_{[0,1]}(y, H) = \ell^{0/1}_{\mathbb{R}}(y^*, \mathrm{logit}(H)),$$
$$\ell^{0/1}_{\mathbb{R}}(y^*, H) = \ell^{0/1}_{[0,1]}(y, \mathrm{logit}^{-1}(H)).$$

We have implicitly closed the domain of the logit, adding two symbols $\pm\infty$ to ensure that the eventual infinite values for H can be scaled back to $[0,1]$.

The 0/1 loss plays a fundamental role in supervised learning. The objective is to carry out the minimization of its expectation in *generalization*, the so-called *true risk*. Very often, however, this task can be relaxed to the minimization of the *empirical risk* of H, which is just (1) with the 0/1 loss [9]:

$$\varepsilon^{0/1}(\mathcal{S}, H) \doteq \sum_i \ell^{0/1}(c_i, H(\boldsymbol{o}_i)). \tag{3}$$

Over the last decade, researchers have found that (3) can be computationally efficiently minimized if we, rather, focus on the minimization of a *surrogate risk* (surrogate for short), i.e., a function $\varepsilon(\mathcal{S}, H) \doteq \sum_i \ell(c_i, H(\boldsymbol{o}_i))$ with [1]:

$$\varepsilon^{0/1}(\mathcal{S}, H) \leq \varepsilon(\mathcal{S}, H).$$

There are numerous examples of surrogates, four of which are particularly important in supervised learning. The corresponding losses are

$$\ell^{\exp}_{\mathbb{R}}(y^*, H) \doteq \exp(-y^* H), \tag{4}$$

$$\ell^{\log}_{\mathbb{R}}(y^*, H) \doteq \log(1 + \exp(-y^* H)), \tag{5}$$

$$\ell^{\mathrm{sqr}}_{\mathbb{R}}(y^*, H) \doteq (1 - y^* H)^2, \tag{6}$$

$$\ell^{\mathrm{hinge}}_{\mathbb{R}}(y^*, H) \doteq \max\{0, 1 - y^* H\}. \tag{7}$$

Equation (4) is the exponential loss, (5) is the logistic loss, (6) is the squared loss, and (7) is hinge loss. These losses play fundamental roles in some of the most popular supervised learning algorithms such as AdaBoost, [10], Additive logistic regression [5], Support Vector Machines [6].

3 Permissible Surrogates

To question the existence of these surrogates and more precisely the loss ℓ, let us build it upon three assumptions that underlie a majority of works in supervised learning. These assumptions, stated for $\mathrm{im}(H) \subseteq [0,1]$ without loss of generality, are

A1. *The loss is lower bounded* by 0. We have

$$\ell(.,.) \geq 0.$$

A2. *The loss is a proper scoring rule*. Consider a singleton domain $\mathcal{O} = \{\boldsymbol{o}\}$. Then, the best (constant) prediction is

$$\arg\min_{x \in [0,1]} \varepsilon_{[0,1]}(\mathcal{S}, x) = p \doteq \hat{\mathbf{P}}\mathbf{r}[c = c^+ | \boldsymbol{o}] \in [0,1],$$

where p is the relative proportion of positive examples with observation o.

A3. *The loss is symmetric* in the following sense:

$$\ell(y, H) = \ell(1 - y, 1 - H), \forall y \in \{0, 1\}, \forall H \in [0, 1].$$

Lower boundedness in **A1** is standard. For **A2**, we can equivalently write:

$$\varepsilon_{[0,1]}(\mathcal{S}, x) = p\ell_{[0,1]}(1, x) + (1 - p)\ell_{[0,1]}(0, x),$$

which is just the expected loss of zero-sum games used in [8, (8)] with Nature states reduced to the class labels. The fact is that the minimum is achieved at $x = p$ makes the loss a proper scoring rule. p also defines *Bayes classifier*, i.e., the one which minimizes the $0/1$ loss [11]. **A3** implies $\ell_{[0,1]}(1, 1) = \ell_{[0,1]}(0, 0)$, which is virtually assumed for any domain; otherwise, it scales to $H \in [0, 1]$, a well-known symmetry in the *cost matrix* that holds for domains without class-dependent misclassification costs. This 2×2 matrix, L, gives $l_{ij} \doteq \ell(i - 1, j - 1)$ for any values $(i, j) \in \{1, 2\}^2$. Usually, it is admitted that $\ell(1, 1) = \ell(0, 0)$, i.e., right classification incurs the same loss regardless of the class. Generally, this loss is zero. Problems *without* class-dependent misclassification costs, on which focus the vast majority of theoretical studies, also make the assumption that $\ell(1, 0) = \ell(0, 1)$. Assumption **A3** scales theses two properties to $H \in [0, 1]$.

To state our first result, we need few more definitions. First, for any strictly convex function $\phi : \mathbf{X} \to \mathbb{R}$ defined over an interval $\mathbf{X}$ of $\mathbb{R}$, differentiable over the opened interval, the *Bregman Loss Function* (BLF, [12]) D_ϕ with generator ϕ is

$$D_\phi(x\|x') \doteq \phi(x) - \phi(x') - (x - x')\nabla_\phi(x'), \tag{8}$$

where ∇_ϕ denotes the first derivative of ϕ.

Second, we extend a terminology due to [3], and define a function $\phi : [0, 1] \to \mathbb{R}_+$ to be *permissible* iff $-\phi$ is differentiable on $(0, 1)$, strictly concave, symmetric about $x = \frac{1}{2}$, and with $-\phi(0) = -\phi(1) \doteq a_\phi \geq 0$ ($a_\phi = 0$ for all popular permissible ϕ [3]). We let $b_\phi \doteq -\phi(1/2) - a_\phi > 0$. Permissible functions are a subset of the generalized entropies studied, e.g., in [8]. Finally, we say that loss $\ell_{[0,1]}$ is *properly defined* iff $\mathrm{dom}(\ell) = [0, 1]^2$, and it is twice differentiable on $(0, 1)^2$. This last definition is only a technical convenience: even the $0/1$ loss coincides on $\{0, 1\}$ with properly defined losses. In addition, the differentiability condition would be satisfied by many popular surrogates. Hinge loss (7) is a notable exception, yet it plays a key role in the properties of *permissible surrogates*, for which the following lemma is central:

Lemma 1. *Any loss $\ell(.,.)$ is properly defined and satisfies assumptions* **A1**, **A2**, *and* **A3** *if and only if $\ell(y, H) = D_\phi(y\|H)$ for some permissible function ϕ.*

Proof. ($\Leftarrow$) Assumption **A3** follows from the strict concavity and symmetry of $-\phi$. Assumptions **A1** and **A2** follow from usual properties of BLFs [12]. ($\Rightarrow$) Without assumption **A3**, $\ell(y, H)$ is a BLF [12], $D_\phi(y\|H)$ for some strictly convex function ϕ, differentiable on $(0, 1)$. Modulo rearrangements in assumption **A3**, we obtain

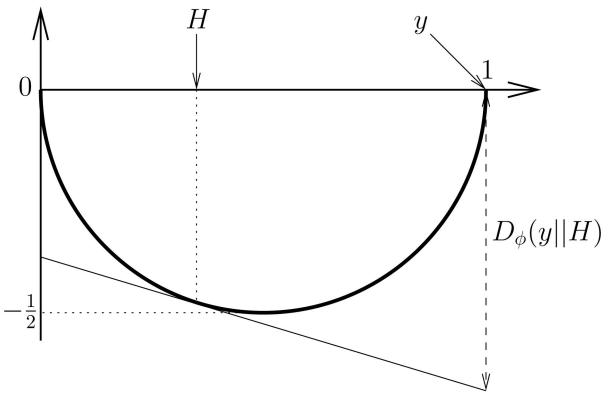

Fig. 1. Plot of $\ell(y, H) = D_\phi(y\|H)$ (Lemma 1) for $\phi = \phi_{\mathrm{M}}(x)$ in (21).

$$\nabla_{\tilde{\phi}}(H) = (\tilde{\phi}(H) - \tilde{\phi}(y))/(H - y), \forall y, H \in [0, 1],$$

with $\tilde{\phi}(x) = -\phi(1 - x) + \phi(x)$. We now have $\tilde{\phi}(x) = ax + b$ for some $a, b \in \mathbb{R}$. Since $\tilde{\phi}(1 - x) = -\tilde{\phi}(x)$, we easily obtain $a = b = 0$, i.e., $\phi(x) = \phi(1 - x)$. Ultimately, since a BLF $D_\phi(y\|H)$ does not change by adding a constant term to ϕ, we can suppose without loss of generality that $\phi(0) = \phi(1) = -a_\phi \leq 0$, which makes that ϕ is permissible. □

ϕ is thus the "signature" of the loss. Notice that we could have replaced **A1** by a simple lower boundedness condition without reference to zero, in which case from Lemma 1, the loss would be a BLF plus a constant factor, without impact on the structural or minimization properties that are to come. Using Lemma 1, Fig. 1 depicts an example of $\ell(y, H)$ for ϕ as in (21).

For any strictly convex function $\phi : \mathbf{X} \to \mathbb{R}$ defined over an interval $\mathbf{X}$ of $\mathbb{R}$, differentiable over the opened interval, the *Legendre conjugate* $\phi^\star$ of ϕ is defined as

$$\phi^\star(x) \doteq \sup_{x' \in \mathrm{int}(\mathbf{X})} \{xx' - \phi(x')\}. \tag{9}$$

Because of the strict convexity of ϕ, the analytic expression of the Legendre conjugate becomes:

$$\phi^\star(x) = x\nabla_\phi^{-1}(x) - \phi\big(\nabla_\phi^{-1}(x)\big).$$

$\phi^\star$ is also strictly convex and differentiable. Hereafter, unless otherwise stated, ϕ always denote a permissible function. The following lemma is simple to prove, yet its identities shall be central for the remaining of the paper. It is stated for $\mathrm{im}(H) = \mathbb{O}$, without loss of generality.

Lemma 2. *We have*

$$D_\phi\big(y\|\nabla_\phi^{-1}(H)\big) = \phi^\star(-y^*H) - a_\phi, \tag{10}$$

$$= D_\phi\big(0\|\nabla_\phi^{-1}(-y^*H)\big), \tag{11}$$

$$= D_\phi\big(1\|\nabla_\phi^{-1}(y^*H)\big). \tag{12}$$

Proof. Direct derivations using **A3**, the possible values for y, y^*, and the fact that $\nabla_\phi^{-1}(-y) = 1 - \nabla_\phi^{-1}(y)$. □

There are three main consequences to Lemma 2 that are reviewed afterward. The first is related to the definition of permissible surrogates; the second is a link with exponential families of distributions; the third is related to margins.

3.1 Permissible Surrogates

Throughout Lemma 1, Lemma 2 makes the link between any loss that meets **A1**, **A2**, **A3**, and the Legendre conjugate of its Bregman generator. $\phi^\star(x)$ is strictly convex and satisfies the following three relationships that are easy to check:

$$\phi^\star(x) = \phi^\star(-x) + x, \tag{13}$$

$$\phi^\star(0) = -\phi(1/2), \tag{14}$$

$$\lim_{x \to \mathrm{infim}(\nabla_\phi)} \phi^\star(x) = a_\phi. \tag{15}$$

Let

$$F_\phi(x) \doteq (\phi^\star(-x) - a_\phi)/b_\phi. \tag{16}$$

It follows that $\lim_{x \to \mathrm{supim}(\nabla_\phi)} F_\phi(x) = 0$ from (15) and $\lim_{x \to \mathrm{infim}(\nabla_\phi)} F_\phi(x) = -x/b_\phi$ from (13). We get that the asymptotes of all F_ϕ can be summarized as

$$\tilde{\ell}(x) = x(\sigma(x) - 1)/(2b_\phi). \tag{17}$$

When $b_\phi = 1$, (17) is the linear hinge loss [13], a generalization of (7) for which $x \doteq y^* H - 1$. Thus, while hinge loss is not properly defined, it defines the limit behavior of all F_ϕ. The reason why F_ϕ is important follows. F_ϕ is strictly convex and $F_\phi(0) = 1$ from (14). We easily get

$$\ell^{0/1}_{\mathbb{R}}(y^*, H) \leq F_\phi(y^* H), \tag{18}$$

and we immediately get the statement of the following lemma:

Lemma 3. $\varepsilon^{0/1}(\mathcal{S}, H) \leq \varepsilon^\phi_{\mathbb{R}}(\mathcal{S}, H) \doteq \sum_i F_\phi(y_i^* H(\boldsymbol{o}_i))$.

It turns out that F_ϕ spans a wide subclass of the *convex surrogates* that are common to the induction of linear separators (LS) [2]. In this case, $H(\boldsymbol{o}) \doteq \sum_t \alpha_t h_t(\boldsymbol{o})$ for features h_t with $\mathrm{im}(h_t) \subseteq \mathbb{R}$ and leveraging coefficients $\alpha_t \in \mathbb{R}$.

Let us refer to any F_ϕ in (16) as a *Permissible Convex Loss* (PCL) and $\varepsilon^\phi_{\mathbb{R}}(\mathcal{S}, H)$ as the corresponding *Permissible Convex Surrogate* (PCS). Remark that Adaboost's exponential loss (4) is not a PCL due to (13). In fact, it is well known that it is an approximation to the logistic loss (5), and it turns out that this loss *is* a PCL.

PCS has interesting relationships with respect to two prominent previous approaches that seek general properties of surrogates [1], [7]. First, let $F(x)$ denote a general loss and introduce the conditional F-risk:

$$\varepsilon^\eta_{\mathbb{R}}(x) \doteq \eta F(x) + (1 - \eta) F(-x), \forall \eta \in [0, 1].$$

To understand the meaning of this risk, consider the setting of assumption **A2**, in which all examples have the same observation. Suppose that the two classes are in proportion η, $1 - \eta$. Then, $\varepsilon^\eta_{\mathbb{R}}(x)$ is just the surrogate risk associated to $\mathcal{S}$ if we plug in $x = H$. Classification calibration requires that, for any $\eta \neq 1/2$, the minimal risk is smaller than the minimal risk in which we require x to be of a different sign than $2\eta - 1$. More precisely, F is classification calibrated iff

$$\begin{aligned} &\varepsilon^+_{\mathbb{R}}(\eta) < \varepsilon^-_{\mathbb{R}}(\eta), \forall \eta \neq 1/2, \\ &\varepsilon^+_{\mathbb{R}}(\eta) \doteq \inf_{x \in \mathbb{R}} \varepsilon^\eta_{\mathbb{R}}(x), \\ &\varepsilon^-_{\mathbb{R}}(\eta) \doteq \inf_{x \in \mathbb{R}: x(2\eta - 1) \leq 0} \varepsilon^\eta_{\mathbb{R}}(x). \end{aligned} \tag{19}$$

In our setting, quantity $2\eta - 1 \in [-1, 1]$ is just another matching transform, like (2). Furthermore, if we make assumption **A2**, then $\sigma(2\eta - 1)$ is the best possible real-valued prediction for the classes, in the same way as $\sigma(\mathrm{logit}(\eta))$ would be, for example. Thus, condition (19) states that from the efficient minimization of the surrogate risk necessarily follows the most accurate prediction of the classes, for every observation. Failing to meet this weak condition would severely undermine the usefulness of the surrogate for classification purposes. It follows from [1, Theorem 4], that any PCS is classification calibrated. PCS also belongs to the $\mathcal{U}$-Boost surrogates, which contains every convex surrogate. This latter class is mainly built on convenient assumptions for minimization purposes. It does not have a classification rationale, and it is interesting to notice that while not all classification calibrated surrogates are convex, $\mathcal{U}$-Boost surrogates are not all classification calibrated so that PCS lies at the intersection of both, sharing the classification rationale and convenient technical properties.

Below are examples of permissible functions ϕ that have been arranged from the bottommost to the topmost function (when scaled so that $\phi(1/2) = -1$).

$$\phi_\mu(x) \doteq -(\mu + (1 - \mu)\sqrt{x(1 - x)}), \forall \mu \in (0, 1), \tag{20}$$

$$\phi_{\mathrm{M}}(x) \doteq -\sqrt{x(1 - x)}, \tag{21}$$

$$\phi_{\mathrm{Q}}(x) \doteq x \log x + (1 - x)\log(1 - x), \tag{22}$$

$$\phi_{\mathrm{B}}(x) \doteq -x(1 - x). \tag{23}$$

When scaled so that $\phi(1/2) = -1$, most confound with the opposite of popular choices: Gini index for (23) [18], Bit-entropy for (22) [19], and Matsushita's error for (21) [3], [20]. Table 1 gives the expressions of F_ϕ, $\hat{\mathbf{Pr}}_\phi[c = c^+ | H; \boldsymbol{o}]$ along with the right $\mathrm{im}(H) = \mathbb{O} \subseteq \mathbb{R}$ for the permissible functions in (20)-(23). Notice that the logistic loss (5) and the squared loss (6) are PCLs and the logit is actually the matching real prediction (26) for (22). It is well known that the direct fitting of a LS with the squared loss is not a good idea [5]; the row for (23) in Table 1 corroborates this, as the PCL regime restrains the output of H to be in $[-1, 1]$. Fig. 2a provides examples of plots for ∇_ϕ for the permissible functions in (20)-(23). The sigmoid curve on the right, indexed by variable ζ, is for a permissible ϕ_ζ as follows ($\forall \zeta \in \mathbb{R}_{-,*}$):

$$\phi_\zeta(x) \doteq -\frac{2}{\zeta} \log \cosh \left(\frac{\zeta}{2} \left(x - \frac{1}{2} \right) \right). \tag{24}$$

When properly scaled, this permissible function is located in between $2\phi_{\mathrm{B}}$ in (23) and $-\min\{x, 1 - x\}$ (and strictly in

TABLE 1
Correspondence between Permissible Functions, the Corresponding PCLs, and the Matching $[0,1]$ Predictions

$\phi(x)$	a_ϕ	$\mathrm{im}(\nabla_\phi)$ $\supseteq \mathrm{im}(H)$	$F_\phi(y^*H)$ $= (\phi^\star(-y^*H) - a_\phi)/b_\phi$	$\hat{\mathbf{Pr}}_\phi[c = c^+ \mid H; \boldsymbol{o}]$ $= \nabla_\phi^{-1}(H)$
(20)	μ	$\mathbb{R}$	$\frac{1}{1-\mu}\left(-y^*H + \sqrt{(1-\mu)^2 + (y^*H)^2}\right)$	$\frac{1}{2}\left(1 + H/\sqrt{(1-\mu)^2 + H^2}\right)$
(21)	0	$\mathbb{R}$	$-y^*H + \sqrt{1 + (y^*H)^2}$	$\frac{1}{2}\left(1 + H/\sqrt{1 + H^2}\right)$
(22)	0	$\mathbb{R}$	$\log(1 + \exp(-y^*H))$	$\exp(H)/(1 + \exp(H))$
(23)	0	$[-1, 1]$	$(1 - y^*H)^2$	$\frac{1}{2}(1 + H)$

between for $x \neq 0, 1/2, 1$). Tuning $\zeta \in \mathbb{R}_{-,*}$ makes the function span all the available area. It was chosen to show that there can be different concave/convex regimes for ∇_ϕ. Since $\mathrm{dom}(F_\phi) = \mathrm{im}(\nabla_\phi)$, there are also much different domains for the PCLs.

3.2 Links with the Exponential Families of Distributions

We have the remarkable property that the BLF for the generator ϕ equals that of the Legendre conjugate on *swapped* gradient arguments:

$$D_\phi(y\|p) = D_{\phi^\star}(\nabla_\phi(p)\|\nabla_\phi(y)). \tag{25}$$

This equality is important because it unifies $[0,1]$ predictions (left) and $\mathbb{R}$eal (confidence-rated) predictions (right), and the right-hand side spans a whole subclass of *matching losses* [14] that are well known in online learning. Equation (10), and, so, any PCL, makes the same tight connection between the two predictions. Let this connection be more formal: The *matching* $[0,1]$ prediction for some H with $\mathrm{im}(H) = \mathbb{O}$ is

$$\hat{\mathbf{Pr}}_\phi[c = c^+ \mid H; \boldsymbol{o}] \doteq \nabla_\phi^{-1}(H(\boldsymbol{o})). \tag{26}$$

This quantity is $< 1/2$ iff $H < 0$, for any permissible ϕ. In the same way as we did for the logit, we implicitly close $\mathrm{im}(\nabla_\phi)$ for (26), adding two symbols $\pm\infty$ so that (26) properly scales back to $[0,1]$. With the definition in (26), illustrated in Table 1 (rightmost column), we can explicit the true nature of the minimization of any PCS with real-valued hypotheses like linear separators. Using the general bijection between BLFs and the exponential families of distributions [15], [8], there exists through (10) a bijection between PCL and a subset of these exponential families whose members' pdfs may be written:

$$\mathbf{Pr}_\phi[y|\theta] = \exp\big(-D_\phi(y\|\nabla_\phi^{-1}(\theta))\big) + \phi(y) - \nu(y)),$$

where $\theta \in \mathbb{R}$ denotes the natural parameter of the pdf, and $\nu(.)$ is used for normalization. Plugging $\theta = H(\boldsymbol{o})$, using (10) and (26), we obtain that any PCS can be rewritten as

$$\varepsilon^\phi_{\mathbb{R}}(\mathcal{S}, H) = u + \sum_i -\log \hat{\mathbf{Pr}}_\phi[y_i|H(\boldsymbol{o}_i)],$$

where u does not play a role in the minimization of the PCS with H. We obtain the following lemma, in which we suppose again that $\mathrm{im}(H) = \mathbb{O}$.

Lemma 4. *Minimizing any PCS with classifier H yields a maximum likelihood estimation, for each observation $\boldsymbol{o}$, of the natural parameter $\theta = H(\boldsymbol{o})$ of an exponential family defined by signature ϕ.*

When minimizing any PCS, real-valued hypotheses like linear separators may thus be viewed as estimating the natural parameters; by duality, classifiers that are naturally able to fit $[0,1]$ values, such as decision trees, would rather be considered estimating the expectation parameters of the corresponding exponential families, i.e., $\nabla_\phi^{-1}(\theta)$ (Section 4.2).

To end up, only one exponential family is in fact concerned in our setting. Assuming $y \in \{0,1\}$, the pdf simplifies and we end up with $\mathbf{Pr}_\phi[y|\theta] = 1/(1 + \exp(-\theta))$, the logistic prediction for a Bernoulli prior. To summarize, minimizing any surrogate whose loss meets **A1**, **A2**, and **A3**

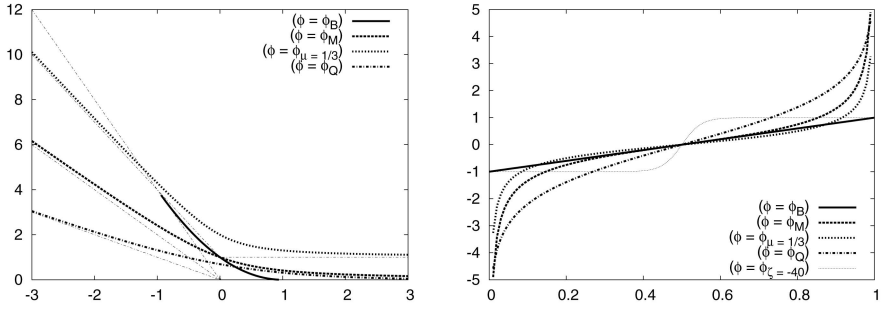

Fig. 2. (a) Bold curves depict plots of $\phi^\star(-x)$ for the ϕ in (20)-(23); thin dotted half-lines display parts of its asymptotes. (b) Plots of ∇_ϕ for the same ϕ, plus an additional one that displays a particular regime (see the text for details).

(i.e., any PCS) amounts to the same ultimate goal. The crux of the choice of the PCS mainly relies on algorithmic and data-dependent considerations for its efficient minimization.

3.3 Margin Maximization

Researchers in machine learning have soon remarked that the output of classifiers returning real values is useful beyond its thresholding via functions σ or τ (Section 2). In fact, we can also retrieve a measure of its "confidence" [10]. For example, when $\mathrm{im}(H) = \mathbb{O}$, it can be its absolute value [10]. Intuitively, learning should aim at providing classifiers that decide right classes with large confidences. Integrating both notions of class and confidence in criteria to optimize was done via margins [16], [17]. Informally, the (normalized) margin of H on example $(\boldsymbol{o}, y^*)$, $\mu_H((\boldsymbol{o}, y^*))$, takes value in $[-1, 1]$; it is positive only when the class assigned by H is the right one and its absolute value quantifies the confidence in the classification. Different definitions of margins coexist, each of which tailored to a particular kind of classifier, with a particular kind of outputs: for example, in the case of linear separators, we may have [10], [17]:

$$\mu_H((\boldsymbol{o}, y^*)) \doteq \frac{y^* \sum_t \alpha_t h_t(\boldsymbol{o})}{\sum_t \alpha_t}.$$

Lemma 2 suggests a general and simple margin definition that we state for $\mathrm{im}(H) = \mathbb{O}$. Fix

$$\mu_H((\boldsymbol{o}, y^*)) \doteq 2\nabla_\phi^{-1}(y^* H(\boldsymbol{o})) - 1. \tag{27}$$

When ϕ is chosen as in (22), (27) simplifies to the margin adopted in [16] for linear separators. The fact that (27) satisfies the classical properties of margins stated above would not justify its use without a strong link to loss minimization. Lemma 2 gives this link, as (12) states that the minimization of any loss that meet **A1**, **A2**, **A3** is strictly equivalent to margin maximization. This justification to margins is new. Since it does not depend on the type of classifier induced, it represents a valuable companion to the statistical rationale of margins provided for LS in [17]. Finally, since ϕ is permissible, (26) yields

$$\mu_H((\boldsymbol{o}, y^*)) = y^*(\hat{\mathbf{Pr}}_\phi[c = c^+ | H; \boldsymbol{o}] - \hat{\mathbf{Pr}}_\phi[c = c^- | H; \boldsymbol{o}]) \in [-1, 1],$$

a quantity that does not depend on ϕ outside the class membership probability estimators, confined in $[0, 1]$. This is convenient for experiments as we can make fair comparisons between margins for different ϕ.

4 Minimization Algorithms for Any Permissible Surrogate

4.1 Linear Separators

4.1.1 Definitions

Let $H \in$ LS, and suppose that the permissible function ϕ is such that $\mathrm{im}(\nabla_\phi) = \mathbb{R}$ (see Table 1). We begin with few more definitions. Because any BLF is strictly convex in its first argument, we can compute its Legendre conjugate as in (9). In fact, we shall essentially need the argument that realizes the supremum, for any permissible ϕ: For any $x \in \mathbb{R}$, for any $p \in [0, 1]$, we let

$$x \diamond p \doteq \arg\sup_{p' \in [0,1]} \{xp' - D_\phi(p' \| p)\}. \tag{28}$$

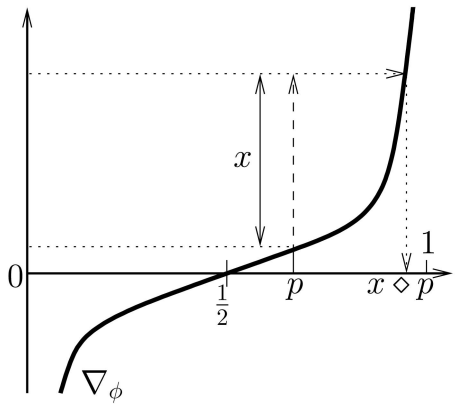

Fig. 3. Construction of the Legendre dual $x \diamond p$. ∇_ϕ is symmetric around point $(1/2, 0)$, for any permissible ϕ.

We do not make reference to ϕ in the $\diamond$ notation as it shall be clear from context. We name $x \diamond p$ the Legendre *dual* of the ordered pair (x, p), closely following a notation by [9]. The Legendre dual satisfies:

$$\nabla_\phi(x \diamond p) = x + \nabla_\phi(p), \tag{29}$$

$$x \diamond (x' \diamond p) = (x + x') \diamond p, \forall x, x' \in \mathbb{R}, \forall p \in [0, 1]. \tag{30}$$

Because ϕ is permissible, the Legendre dual is unique and it is always in $[0, 1]$: p' is not chosen in the interior of $[0, 1]$ in (28) to follow the same scaling issues as in (26). The formula in (28) is not simple at first glance, but its construction can be represented in a simple way, as explained in Fig. 3.

We follow the setting of [9] and suppose that we have T features h_t $(t = 1, 2, \ldots, T)$ known in advance, the problem thus reducing to the computation of the leveraging coefficients. We define $m \times T$ matrix M with

$$m_{it} \doteq -y_i^* h_t(\boldsymbol{o}_i). \tag{31}$$

Given leveraging coefficients vector $\boldsymbol{\alpha} \in \mathbb{R}^T$, we thus get

$$-y_i^* H(\boldsymbol{o}_i) = (M\boldsymbol{\alpha})_i. \tag{32}$$

We can easily specialize this setting to classical greedy induction frameworks for LS: In classical boosting, at step j, we would fit a single α_t [9]; in totally corrective boosting, we would rather fit $\{\alpha_t, 1 \leq t \leq j\}$ [21]. The question is thus: Given some PCS, how can we fit the leveraging coefficients for its minimization? If an algorithm exists that provably achieves the minimization of any given PCS, we say that this algorithm is a *Universal Minimization Algorithm*.

4.1.2 ULS and its Convergence Properties

Fig. 4 provides such an algorithm. In the algorithm, notations are vector based: the Legendre duals are computed componentwise. Hereafter, to save space, we also make use of a vector-based notation for BLFs and it shall mean a componentwise sum of BLFs, such as

Algorithm 1: Algorithm ULS(M, ϕ)

Input: $M \in \mathbb{R}^{m \times T}$, permissible function ϕ;

Let $\boldsymbol{\alpha}_1 \leftarrow \mathbf{0}$; Let $\boldsymbol{w}_0 \leftarrow (1/2)\mathbf{1}$;

for $j = 1, 2, ...J$ **do**

$$\boldsymbol{w}_j \quad \leftarrow \quad (M\boldsymbol{\alpha}_j) \diamond \boldsymbol{w}_0 \ ; \tag{34}$$

Let $\mathcal{T}_j \subseteq \{1, 2, ..., T\}$ chosen according to classical/totally corrective boosting;

Let $\boldsymbol{\delta}_j \leftarrow \mathbf{0}$;

$\forall t \in \mathcal{T}_j$, pick $\delta_{j,t}$ such that:

$$\sum_{i=1}^{m} m_{it}((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j)_i \quad = \quad 0 \ ; \tag{35}$$

Let $\boldsymbol{\alpha}_{j+1} \leftarrow \boldsymbol{\alpha}_j + \boldsymbol{\delta}_j$;

Output: $H(\boldsymbol{x}) \doteq \sum_{t=1}^{T} \alpha_{J+1,t} h_t(\boldsymbol{x}) \in \text{LS}$

Fig. 4. Algorithm ULS.

$$D_\phi(\boldsymbol{a} \| \boldsymbol{b}) = \sum_i D_\phi(a_i \| b_i). \tag{33}$$

The main component of the algorithm is a weight distribution $\boldsymbol{w}$ over the examples of $\mathcal{S}$, initialized to $1/2$ for each example, and updated in an iterative fashion. This distribution serves to compute the leveraging coefficients of the final linear separator.

Theorem 1. *ULS is a Universal Minimization Algorithm.*

Proof. In (34), (30) brings $\boldsymbol{w}_{j+1} = (M\boldsymbol{\alpha}_{j+1}) \diamond \boldsymbol{w}_0 = (M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j$. Using notation (33), we thus have

$$\begin{aligned} &D_\phi(\mathbf{0} \| \boldsymbol{w}_{j+1}) - D_\phi(\mathbf{0} \| \boldsymbol{w}_j) = \\ &\quad - [\phi((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j) - \phi(\boldsymbol{w}_j) + \langle \boldsymbol{w}_j, \nabla_\phi(\boldsymbol{w}_j) \rangle] \\ &\quad + \langle (M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j, \nabla_\phi((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j) \rangle. \end{aligned} \tag{36}$$

Vector notations in (36) are used to simplify notations so that $\phi(\boldsymbol{w}_j)$ represents a vector whose entries are $\phi(w_{j,i})$ and $\nabla_\phi(\boldsymbol{w}_j)$ represents the vector whose entries are $\nabla_\phi(w_{j,i})$, and so on; furthermore, $\langle ., . \rangle$ denotes the inner product. Because of (29), the right inner product is just (for short, $r \doteq \langle (M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j, \nabla_\phi(\boldsymbol{w}_j) \rangle$):

$$\begin{aligned} &\langle (M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j, \nabla_\phi((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j) \rangle \\ &= r + \langle (M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j, M\boldsymbol{\delta}_j \rangle \\ &= r - \sum_{i=1}^{m} y_i^* \sum_{t=1}^{T} \delta_{j,t} h_t(\boldsymbol{o}_i)((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j)_i \\ &= r - \sum_{t=1}^{T} \delta_{j,t} \sum_{i=1}^{m} y_i^* h_t(\boldsymbol{o}_i)((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j)_i \\ &= r + \sum_{t=1}^{T} \delta_{j,t} \underbrace{\sum_{i=1}^{m} m_{it}((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j)_i}_{b_t} = r. \end{aligned} \tag{37}$$

Equation (37) holds because $\delta_{j,t} = 0$, or $b_t = 0$ from the choice of $\delta_{j,t}$ in (35). We obtain

$$D_\phi(\mathbf{0} \| \boldsymbol{w}_{j+1}) - D_\phi(\mathbf{0} \| \boldsymbol{w}_j) = -D_\phi(\boldsymbol{w}_{j+1} \| \boldsymbol{w}_j). \tag{38}$$

Let $A_\phi(\boldsymbol{w}_{j+1}, \boldsymbol{w}_j) \doteq -D_\phi(\boldsymbol{w}_{j+1} \| \boldsymbol{w}_j)/b_\phi$, which is just, from (38) and Lemma 2 (11), the difference between two successive PCS. Suppose that ULS works in a classical boosting scheme ($\mathcal{T}_j$ is a singleton) and that we have reached the stage on which for any choice of h_t among the T, $\boldsymbol{w}_{j+1} = \boldsymbol{w}_j$ (ULS has converged). In this case, $\boldsymbol{\delta}_j = \mathbf{0}$ and so $\forall t = 1, 2, \ldots, T$, $\sum_{i=1}^{m} m_{it}(\mathbf{0} \diamond \boldsymbol{w}_j)_i = \sum_{i=1}^{m} m_{it} \boldsymbol{w}_{j,i} = 0$, i.e., $\boldsymbol{w}_j^\top M = \boldsymbol{w}_{j+1}^\top M = \mathbf{0}$. Thus, $\boldsymbol{w}_j, \boldsymbol{w}_{j+1} \in \text{Ker} M^\top$, where $\text{Ker} N$ denotes the kernel of linear operator N:

$$\text{Ker} N \doteq \{\boldsymbol{x} \in \mathbb{R}^m : N\boldsymbol{x} = \mathbf{0}\}.$$

This condition, along with the fact that $A_\phi(\boldsymbol{w}_{j+1}, \boldsymbol{w}_j) < 0$ whenever $\boldsymbol{w}_{j+1} \neq \boldsymbol{w}_j$, makes $A_\phi(\boldsymbol{w}_{j+1}, \boldsymbol{w}_j)$ an *auxiliary function* for ULS, which is enough to prove the convergence of ULS towards the optimum [9]. The case of totally corrective boosting is simpler as, after the last iteration, we would have $\boldsymbol{w}_{J+1} \in \text{Ker} M^\top$. □

In practice, it may be a tedious task to satisfy exactly (38), in particular for totally corrective boosting [21]. With respect to previous approaches that have been built upon [22] (or others [7]), ULS has two technical advantages. First, up to a normalization coefficient, $\boldsymbol{w}_j$ is always a distribution because ϕ is permissible. We do not need explicit constraints to keep nonnegative weights. Second, ULS is always guaranteed to work, as now shown.

Lemma 5. *Suppose that there does not exist some h_t with all m_{it} of the same sign, $\forall i = 1, 2, \ldots, m$. Then, for any choice of $\mathcal{T}_j$ in ULS, (35) has always a finite solution.*

Proof. Let

$$Z \doteq D_\phi(\mathbf{0} \| (M\boldsymbol{\alpha}_{j+1}) \diamond \boldsymbol{w}_0). \tag{39}$$

We have $Z = -ma_\phi + \sum_{i=1}^{m} \phi^\star((M(\boldsymbol{\delta}_j + \boldsymbol{\alpha}_j))_i)$ (Lemma 2), a function convex in all leveraging coefficients. Define $|\mathcal{T}_j| \times |\mathcal{T}_j|$ matrix E with $e_{uv} \doteq \partial^2 Z / (\partial \delta_{j,u} \partial \delta_{j,v})$ (for the sake of simplicity, $\mathcal{T}_j = \{1, 2, \ldots, |\mathcal{T}_j|\}$, where $|.|$ denotes the cardinal). We have $e_{uv} = \sum_{i=1}^{m} m_{iu} m_{iv} / \varphi(((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j)_i)$, with

$$\varphi(x) \doteq \mathrm{d}^2\phi(x)/\mathrm{d}x^2, \tag{40}$$

a function strictly positive in $(0,1)$ since ϕ is permissible. Let $q_{i,j} \doteq 1/\varphi(((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j)_i) > 0$. It is easy to show that

$$\boldsymbol{x}^\top E \boldsymbol{x} = \sum_{i=1}^{m} q_{i,j} \langle \boldsymbol{x}, \tilde{\boldsymbol{m}}_i \rangle^2 \geq 0, \forall \boldsymbol{x} \in \mathbb{R}^{|\mathcal{T}_j|}, \tag{41}$$

with $\tilde{\boldsymbol{m}}_i \in \mathbb{R}^{|\mathcal{T}_j|}$ the vector containing the entries m_{it} with $t \in \mathcal{T}_j$. Thus, E is positive semidefinite; as such (35), which is the same as solving $\partial Z/\partial \delta_{j,u} = 0$, $\forall u \in \mathcal{T}_j$ (i.e., minimizing Z), has always a solution. □

The condition for the lemma to work is absolutely not restrictive, as if such an h_t were to exist, we would not need to run ULS; indeed, we would have either $\varepsilon^{0/1}(\mathcal{S}, h_t) = 0$ or $\varepsilon^{0/1}(\mathcal{S}, -h_t) = 0$.

4.1.3 Geometric Properties of ULS

The auxiliary function in the analysis of ULS shifts the analysis of the learning problem on labels to that of a geometric problem on weights. Its definition involves (38), the generalized Pythagorean theorem [21], [7]. It is not the purpose of our paper to further develop on this geometric aspect of learning, which has already been explained earlier in [7], as ULS exploits the same geometric flatness, the same fibrations and foliations of the weight space as $\mathcal{U}$-Boost to progress towards better solutions [7]. Our setting is, however, significantly different, resulting in different algorithms. The multiclass framework of [7] makes them work directly with linear combinations of indicator functions—thus, mixing the real and the $[0,1]$ predictions and the matching real prediction they use to exemplify their approach on two-class classification (their equation 3.35) does not depend on any signature of the loss used, contrary to ours in (26). Also, we do not need to integrate an auxiliary function in selecting classifiers and updating weights, resulting in different choices. One last, big difference is that ULS works with simultaneous updates of leveraging coefficients, guaranteeing a final solution which is the same as the sequential update if the optimum is achieved by a single classifier (modulo scaling factors). This setting is only sketched in [7] and these guarantees are not proven.

Keeping these differences in mind, it is interesting to notice that our analysis is dramatically simplified compared to the analysis which leads to the analogous of (38) for $\mathcal{U}$-Boost; we are also able to go further than the mild convergence, and conclude to the convergence to the *global* optimum of the PCS. This optimum is defined by the LS with features in M that realizes the smallest PCS [9].

4.1.4 Decision Making with ULS

We can go further in parallel with game theory developed above for **A2**: using notations in [8], the loss function of the decision maker can be written $\varepsilon_{[0,1]}(\mathcal{S}, x) = D_\phi(1 \| q(X))$. **A3** makes it easy to recover losses like the log loss or the Brier score [8], respectively ,from ϕ_{Q} and ϕ_{B} in (22) and (23). In this sense, ULS is also a sound learner for decision making in the zero-sum game of [8]. Notice, however, that, to work, it requires that Nature has a restricted sample space size, more precisely $\{0,1\}$.

4.1.5 Boosting Features of ULS

ULS provides the scaling to the whole set of PCS of two boosting properties [22], [21] that are easy to check. First, the weight of an example decreases iff the example has been given the right class by the classifier which updates the current LS, $H_j \doteq \sum_{t \in \mathcal{T}_j} \delta_{j,t} h_t$. To see this, we only have to look at Fig. 3 and take $x = -y_i^* H_j(\boldsymbol{o}_i)$, which serves as the update for the next weight of example $(\boldsymbol{o}_i, y_i^*)$. Ignoring ties where $H_j = 0$, we obtain that $\sigma(H_j(\boldsymbol{o}_i))$ has the same sign as y_i^*; hence, the example is given the right class iff $x < 0$, i.e., the new weight is smaller than the former weight.

Second, (35) implies that this LS has zero *edge* on $\boldsymbol{w}_{j+1}$ [21]. In the particular case where each $h_t \in \{-1,+1\}$, (35) implies the so-called "error-rate property" proved in [7], which makes that for each $t \in \mathcal{T}_j$, h_t has empirical risk $1/2$ over the next distribution ($\boldsymbol{w}_{j+1}$). Along with a third one, these two properties represent the most popular boosting properties and the most intuitive rationales for the use of boosting algorithms. An in-depth analysis reveals that ULS also scales this third property.

This third property is the most important of all: the guarantee on the convergence rate under very weak assumptions. This paraphrases the so-called "Weak Learning Assumption" (WLA) [10]. To state the WLA, we plug the iteration in the index of the distribution normalization coefficient in (39), and define $Z_j \doteq \|\boldsymbol{w}_j\|_1$ ($\|.\|_k$ denotes the L_k norm). Suppose that, at any step j (and corresponding subset $\mathcal{T}_j$ of indices), the following holds:

$$\exists \gamma_j > 0 : \left| \frac{1}{|\mathcal{T}_j|} \sum_{t \in \mathcal{T}_j} \frac{1}{Z_j} \sum_{i=1}^{m} m_{it} w_{j,i} \right| \geq \gamma_j. \tag{42}$$

The WLA in (42) tells that the (unleveraged) LS defined by the set of features related to $\mathcal{T}_j$ beats *random* by a guaranteed—even if small—amount. A set of random features would indeed yield $\gamma_j = 0$, so we only require the set of features to define a classifier not completely "useless" from the classification standpoint. The WLA in (42) is a generalization of the usual WLA for boosting algorithms that we obtain taking $|\mathcal{T}_j| = 1$, $h_t \in \{-1,+1\}$ [16]. Few boosting algorithms are known that formally boost weak learning in the sense that requiring only the WLA implies guaranteed rates for the loss minimization [10], [16]. We show that ULS satisfies this property *regardless* of ϕ.

To state and prove the property, we need few more definitions. Let $\boldsymbol{m}_t$ denote the tth column vector of M, $a_{\boldsymbol{m}} \doteq \max_t \|\boldsymbol{m}_t\|_2$, and $a_Z \doteq \min_j Z_j$. Let a_γ denote the average of γ_j over all j, and $a_\varphi \doteq \min_{x \in (0,1)} \varphi(x)$ (φ is defined in (40)).

Theorem 2. *Provided the WLA holds, ULS reaches the minimum of the PCS at hand in at most*

$$J = \left\lceil \frac{4 m b_\phi a_{\boldsymbol{m}}^2}{a_\varphi a_Z^2 a_\gamma^2} \right\rceil \tag{43}$$

steps.

Proof. We use Taylor expansions with Lagrange remainder for ϕ, and then the mean value theorem, and obtain that $\forall w, w + \Delta \in [0,1], \exists w^\star \in [\min\{w + \Delta, w\}, \max\{w + \Delta, w\}]$ such that

$$D_\phi(w+\Delta\|w) = \frac{\Delta^2\varphi(w^\star)}{2} \geq \frac{\Delta^2}{2}a_\varphi \geq 0. \tag{44}$$

Cauchy-Schwartz inequality yields

$$\begin{aligned}\forall t \in \mathcal{T}_j, &\sum_{i=1}^m m_{it}^2 \sum_{i=1}^m (w_{j+1,i}-w_{j,i})^2 \\ &\geq \left(\sum_{i=1}^m m_{it}(w_{j+1,i}-w_{j,i})\right)^2 = \left(\sum_{i=1}^m m_{it}w_{j,i}\right)^2,\end{aligned} \tag{45}$$

where (45) holds because of (35). There remains to use m times (44) with $w = w_{j,i}$ and $\Delta = (w_{j+1,i} - w_{j,i})$, sum the inequalities, combine with (45) over $t \in \mathcal{T}_j$ to obtain (with the use of notation (33)):

$$\begin{aligned}D_\phi(\boldsymbol{w}_{j+1}\|\boldsymbol{w}_j) &\geq \frac{a_\varphi}{2|\mathcal{T}_j|}\sum_{t\in\mathcal{T}_j}\frac{\left(\sum_{i=1}^m m_{it}w_{j,i}\right)^2}{\sum_{i=1}^m m_{it}^2} \\ &= \frac{a_\varphi}{2|\mathcal{T}_j|}\sum_{t\in\mathcal{T}_j}\left(\sum_{i=1}^m \frac{m_{it}}{\|\boldsymbol{m}_t\|_2}w_{j,i}\right)^2 \qquad (46) \\ &\geq \frac{a_\varphi Z_j^2}{2a_{\boldsymbol{m}}^2}\left(\frac{1}{|\mathcal{T}_j|}\sum_{t\in\mathcal{T}_j}\frac{1}{Z_j}\sum_{i=1}^m m_{it}w_{j,i}\right)^2 \\ &\geq a_\varphi\left(\frac{a_Z\gamma_j}{2a_{\boldsymbol{m}}}\right)^2. \qquad (47)\end{aligned}$$

Equation (46) follows from Jensen's inequality. Calling once again to notation (33), we also have

$$\begin{aligned}D_\phi(\mathbf{0}\|\boldsymbol{w}_{J+1})/b_\phi &= \frac{1}{b_\phi}D_\phi(\mathbf{0}\|\boldsymbol{w}_1) \\ &\quad + \frac{1}{b_\phi}\sum_{j=1}^J \left(D_\phi(\mathbf{0}\|\boldsymbol{w}_{j+1}) - D_\phi(\mathbf{0}\|\boldsymbol{w}_j)\right) \qquad (48) \\ &= m - \frac{1}{b_\phi}\sum_{j=1}^J D_\phi(\boldsymbol{w}_{j+1}\|\boldsymbol{w}_j).\end{aligned}$$

Equation (48) follows from (38) and the fact is that $\boldsymbol{w}_1 = \boldsymbol{w}_0$ in ULS. Equation (11) in Lemma 2, together with the definition of $\boldsymbol{w}_j$ in (34), yields $D_\phi(0\|w_{J+1,i})/b_\phi = F_\phi(y_i^* H(\boldsymbol{o}_i))$, $\forall i = 1, 2, \ldots, m$. We have $F_\phi(.) \geq 0$ and (47) gives the minimal quantity by which the PCS is guaranteed to decrease under the WLA (48). Thus, ULS must have stopped when the right-hand side of (48) becomes negative. Combining with (47) and one more use of Jensen's inequality yields the statement of the theorem. □

The bound in (43) is mainly useful to prove that the WLA guarantees a convergence rate of order $\mathcal{O}(m/a_\gamma^2)$ for ULS, thereby providing an answer to [7] for a learning algorithm relying on a geometrical content close to that of $\mathcal{U}$-Boost. It is much less useful to provide the best possible bound as it is in fact far from being optimal [21], but it seems that specializing this bound as a function of ϕ would require analysis on a case-by-case basis, such as in [3], [23]. The bound in (43) is inversely proportional to the minimum value of the second derivative of ϕ. Thus, to make the bound smaller, we can pick permissible functions with a stronger concave regime: This observation closely matches one that was given, in the same context, for decision tree induction [3].

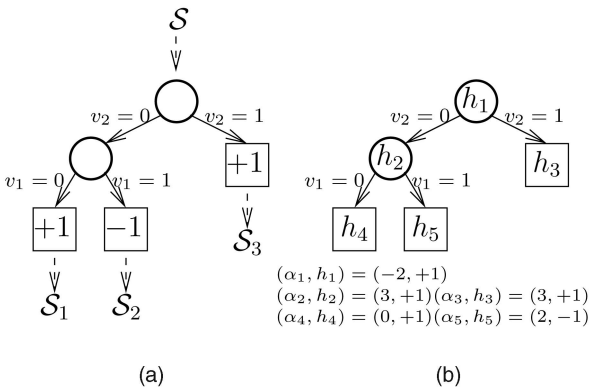

Fig. 5. (a) A decision tree with three leaves (squares) and two internal nodes (circles); (b) an equivalent linearized decision tree, for the proof of Theorem 3.

4.2 Decision Trees

A DT H is a classifier shaped as a rooted directed tree, with internal nodes and leaves. Leaves are labeled by values that are either in $[0, 1]$, or in $\mathbb{R}$, and used to decide the classes as shown in Section 2. Without loss of generality, we suppose that $\mathcal{O} = \{0, 1\}^n$, i.e., all description variables are Boolean. The left part of Fig. 5 presents an example of a DT, where v_1 and v_2 denote description variables. Each internal node is labeled by an observation variable and has outdegree 2, each outgoing arc being labeled by a distinct Boolean test over the node's variable. The classification of some observation starts from the root. For each internal node visited, the observation follows the arc that corresponds to the Boolean test on the variable that it satisfies, until it reaches a leaf which gives its prediction for the class. For example, in Fig. 5, an observation with $v_1 = 1$ and $v_2 = 0$ would be classified by the center leaf labeled "-1".

A DT H induces a partition of $\mathcal{S}$ according to subsets $\mathcal{S}_k$, where $k \in \mathcal{L}(H) \subset \mathbb{N}_*$, and $\mathcal{L}(H)$ is a subset of natural integers in bijection with the set of leaves of the DT (see Fig. 5). We let $S_k^+ \doteq \{(\boldsymbol{o}, c^+) \in \mathcal{S}_k\}$ denote the subset of positive examples that fall on leaf k. To decide a class, we can label leaves using real values to make predictions, following the convention of linear separators (used in Fig. 5). There is, however, a more convenient labeling, which exploits the fact that each leaf makes a constant prediction for a subset of $\mathcal{S}$. Using assumption **A2**, we get the best constant prediction for leaf k:

$$\hat{\mathbf{Pr}}[c = c^+ | H; \boldsymbol{o} \text{ reaches leaf } k] = \frac{|S_k^+|}{|S_k|} \in [0, 1].$$

The most popular DT induction algorithms integrate a stage in which a large DT is induced in a top-down fashion, the so-called TDIDT scheme (Top-Down Induction of DT). This scheme consists, after having initialized the DT to a single leaf, in repeatedly replacing a leaf by a subtree with two leaves (a *stump*) [18], [3], [19]. For this reason, it is convenient to define, for any $k \in \mathcal{L}(H)$ and any Boolean description variable v, $H_{|k\to v}$ to be the DT built from H after having replaced leaf k by the subtree of two

Algorithm 2: Algorithm UDT($\mathcal{S}, \phi$)

Input: Learning sample $\mathcal{S}$, permissible function ϕ;

Let $H \leftarrow$ single leaf;

for $j = 1, 2, ...J$ **do**

Pick some leaf $k' \in \mathcal{L}(H)$, and some observation variable v such that:

$$A_\phi(H, H_{|k' \to v}) \doteq \sum_{k \in \mathcal{L}(H_{|k' \to v})} \frac{|\mathcal{S}_k|}{b_\phi} \left(-\phi \left(\frac{|S_k^+|}{|S_k|} \right) \right) - \sum_{k \in \mathcal{L}(H)} \frac{|\mathcal{S}_k|}{b_\phi} \left(-\phi \left(\frac{|S_k^+|}{|S_k|} \right) \right) < 0 \ ; \quad (49)$$

Let $H \leftarrow H_{|k' \to v}$;

Output: $H \in$ DT

Fig. 6. Algorithm UDT.

leaves rooted at v. The TDIDT scheme can be conveniently abstracted as displayed in Fig. 6. In UDT, ϕ is the free parameter which is instantiated with different choices to yield all popular schemes: (23) is chosen in [18], (22) is chosen in [19], and (21) is chosen in [3]. In fact, it is the *opposite* of the permissible function which is used (see (49) in ULS), but we keep ϕ in order not to laden our notations. All popular TDIDT schemes would also normalize A_ϕ (division by m), but this does not change the choices made for k' and v as, after having picked k', they all pick the best stump, i.e., the one that minimizes (49). It is not hard to prove that UDT is a Universal Minimization Algorithm because of the choice of A_ϕ. The proof of the following theorem shows much more: While bitterly different from each other on paper, UDT and ULS are offshoots of the same algorithm, thereby generalizing an observation of [24] to the whole family of losses that meet assumptions **A1**, **A2**, and **A3**.

Theorem 3. *UDT is a Universal Minimization Algorithm.*

Proof. The proof makes use of linearized decision trees (LDT) of [24]. An LDT has the same graph shape as a DT, but real values are put on every node (not just on leaves). The classification of some observation sums these real values over the whole path that it follows, from the root to a leaf. To each path from the root to a leaf can thus be associated a constant LS that sums these real values. The right part of Fig. 5 presents how to generate the equivalent LDT from the DT given on the left. We can indeed check that $\alpha_1 h_1 + \alpha_2 h_2 + \alpha_5 h_5 = -1$ for the center leaf, and so on for the other leaves.

Thus, we can use ULS to build each of these LS: Each feature h_t is constant and put on some tree node, ULS is run on the subset of $\mathcal{S}$ that reaches the node, in order to compute the leveraging coefficient α_t. The splits are computed after a further minimization of the given PCS.

Suppose that the current LDT H has T nodes, and we wish to compute α_k for some h_k located at leaf node index k. To do so, we number the internal nodes using natural integers, excluding from the choices the integers chosen for the leaves. Let $\wp(k)$ be the set of indices corresponding to the path from the root to leaf k. The solution of (35) can be computed exactly and yields

$$\alpha_k = \frac{1}{h_k} \left(\nabla_\phi \left(\frac{|S_k^+|}{|S_k|} \right) - \sum_{t \in \wp(k) \setminus \{k\}} \alpha_t h_t \right).$$

Thus, for any observation $\boldsymbol{o}$ that reaches leaf k, we get

$$H(\boldsymbol{o}) = \nabla_\phi \left(\frac{|S_k^+|}{|S_k|} \right), \quad (50)$$

the inverse of (26). Finally, the PCS of H simplifies as

$$\begin{aligned} \varepsilon_{\mathbb{R}}^{\phi}(\mathcal{S}, H) &\doteq \sum_i F_\phi(y_i^* H(\boldsymbol{o}_i)) = \frac{1}{b_\phi} \sum_i D_\phi(y_i \| \nabla_\phi^{-1}(H(\boldsymbol{o}_i))) \\ &= \frac{1}{b_\phi} \sum_{k \in \mathcal{L}(H)} \sum_{(\boldsymbol{o}, y) \in \mathcal{S}_k} D_\phi \left(y \middle\| \frac{|S_k^+|}{|S_k|} \right) \\ &= \frac{1}{b_\phi} \sum_{k \in \mathcal{L}(H)} |\mathcal{S}_k| \times \begin{cases} \frac{|S_k^+|}{|S_k|} D_\phi \left(1 \middle\| \frac{|S_k^+|}{|S_k|} \right) \\ + \left(1 - \frac{|S_k^+|}{|S_k|} \right) D_\phi \left(0 \middle\| \frac{|S_k^+|}{|S_k|} \right) \end{cases} \\ &= -\frac{m a_\phi}{b_\phi} + \sum_{k \in \mathcal{L}(H)} \frac{|\mathcal{S}_k|}{b_\phi} \left(-\phi \left(\frac{|S_k^+|}{|S_k|} \right) \right). \end{aligned}$$

It is straightforward to check from this last equality and Lemma 2 that the auxiliary function of ULS becomes exactly (49) in UDT. The LDT obtained is equivalent [24] (see also Fig. 5) to a twin DT in which we put at leaf k either $|S_k^+|/|S_k| \in [0, 1]$ or $\nabla_\phi(|S_k^+|/|S_k|) \in \mathrm{im}(\nabla_\phi) \subseteq \mathbb{R}$, and we finally end up with UDT. □

From (50), we note that the $[0, 1]$ value put at leaf k satisfies $|S_k^+|/|S_k| = \nabla_\phi^{-1}(H(\boldsymbol{o}))$, with $\boldsymbol{o}$ any observation that reaches leaf k. From Section 3.2 and Lemma 4, we note that the leaf value is also $\nabla_\phi^{-1}(\theta)$ and, so, fitting a DT to the minimization of a PCS yields local (leaves-based) maximum likelihood estimators of the expectation parameter of the exponential family defined by signature ϕ.

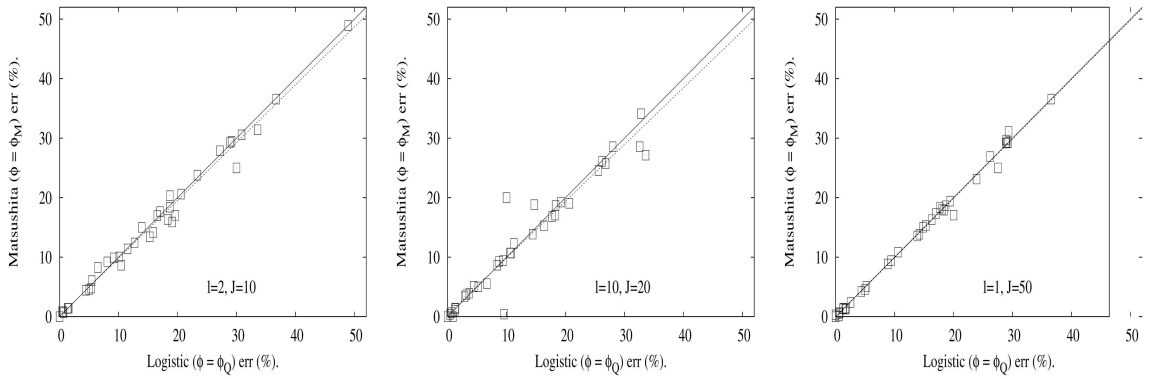

Fig. 7. (a) Error scatterplot ULS(., ϕ_{M}) versus ULS(., ϕ_{Q}), for $l = 2, J = 10$; lines are $y = x$ (plain) and linear regression's (dotted); point below $y = x$ are domains on which ULS(., ϕ_{M}) performs better. (b) Idem for $l = 10, J = 20$. (c) Idem for $l = 1, J = 50$.

5 Experiments

We have compared three flavors of ULS (for $\phi_{\mathrm{M}}, \phi_{\mathrm{Q}}, \phi_{\mu=.99}$) on 40 domains, mostly coming from the UCI repository of ML databases [25], via a stratified 10-fold cross validation to estimate both the error and margin distributions in generalization, where margins are computed as in (27). In domains with more than two classes, we predicted class 1 against all others.

The cumulative margin distribution (CMD) of H is the curve that gives, $\forall \theta \in [-1, 1]$, the proportion of examples whose margin does not exceed θ. Its intersection with line $\theta = 0$ is approximately the estimated true risk. Features have the form:

$$\textbf{If } \mathrm{Monomial} \textbf{ then } \mathrm{Class} = \pm 1 \textbf{ else } \mathrm{Class} = \mp 1.$$

Monomials (Boolean rules) have at most l (fixed) variables. Monomials are induced following TDIDT (UDT) for the repetitive minimization of (48). Leveraging coefficients (35) are approximated up to 10^{-10} precision on the classical boosting scheme. We have run three sets of experiments. In the first, $J = 50$ and $l = 1$: Everything is as if we had in M all size-1 rules ($\approx$ stumps). In the other two, $J = 10$, $l = 2$ and $J = 20$, $l = 10$: The main difference with the first is that we cannot systematically hope UDT to find the best monomial of size $\leq l$ because of its greedy nature.

Beyond testing the new PCS defined by (21) (hereafter called *Matsushita's PCS*) against the popular logistic surrogate, we wanted to address two objectives. First, if we follow [3] and Theorem 3 that binds ULS and UDT, stronger convexity on ϕ might bring a faster decrease of the PCS during the early rounds. The three sets of experiments should provide further hints on this important property, at both levels of ULS (outer level) *and* UDT (inner level). Second, we wanted to have more hints between the quality of estimation (26) and generalization abilities. This is important because there are also links between estimation and statistical consistency [2]. For example, as $\mu \to 1$ in (20), Table 1 shows that there should be a dilation of the estimations (26) around $p = 1/2$ (they $\to 0$ or 1), margins should get dilated around $\theta = 0$ (they $\to \pm 1$), the CMD should gradually become piecewise constant, and performances on testing should be worse than for Mastushita's PCS ($\phi_{\mathrm{M}} = \phi_{\mu=0}$).

Fig. 7 displays the results obtained. The left scatterplot tends to display the ability of Matsushita's PCS (ϕ in (21)) to provide better results than the logistic surrogate (ϕ in (22)) at the early rounds. A Student paired test over the 40 domains confirms this tendency with a P value ≈ 0.07 for rejecting the identity hypothesis. The linear regression gives $y \approx 0.97x + 0.09$, indicating that the tendency is more pronounced for harder data sets. Increasing both l and J ($l = 10, J = 20$) accentuates the pattern ($y \approx 0.95x + 0.35$), even when larger deviations between the algorithms increase the P value (≈ 0.26). This phenomenon, which was previously underlined from the theory standpoint for decision tree induction in [3], is actually predicted up to some extent by Theorem 2, as the bound on J is inversely proportional to the minimum of the second derivative of ϕ (40).

This phenomenon becomes predictably dampened as classifiers become large: running ULS for a larger number of iterations and smaller sizes ($l = 1, J = 50$) considerably dampens the differences. The linear regression gives now $y \approx 0.99x - 0.08$ and Student paired test's P value ≈ 0.17: We cannot conclude for a difference on testing. The comparison with ϕ_μ (20) is also very informative. The scatterplots against Matsushita's PCS (not shown) gives a very clear winning to Matsushita's PCS, with a P value $\ll 10^{-4}$ in both cases. The CMDs (Fig. 8) give an indication of the reason why this happens. As J increases, a majority (≈ 66 percent) of the CMDs approach piecewise constancy for ϕ_μ with respect to ϕ_{M}, like in `heart-cleve`. Those for which ϕ_μ yields a true risk smaller than the other two PCSs (like `heart-cleve`) are few (≈ 10 percent), indicating that the tendency to overfit the estimation (26) damages, in most cases, the true risk. Even when the stair shape is less visible (Fig. 8, `glass`), ϕ_μ is generally beaten.

6 Conclusion

Bartlett et al. [1] solve an important statistical issue about convex surrogates. They prove that if it meets a technical condition known as "classification calibration," then any consistent minimization procedure for the surrogate also meets Bayes consistency, i.e., it enjoys efficient true risk minimization as the sample size m increases. Our results show that there exists a subclass of classification calibrated

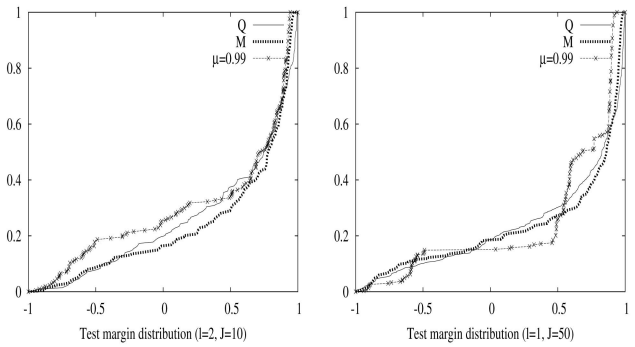

Fig. 8. (a) CMDs over UCI domain `glass` for ULS (., ϕ_{M}) (dotted blue), ULS (., ϕ_{Q}) (red), and ULS(., $\phi_{\mu=.99}$) (green) when $l = 2, J = 10$. (b) Idem for $l = 1, J = 50$ and UCI domain `heart-cleve`.

losses, PCS, that exhibits strong algorithmic minimization properties for the empirical risk, an issue that was left open in [1] (in which the connection to Bregman divergences was also not made). The induction of linear separators for the minimization of any PCS follows a geometrical approach previously presented in [7], and we show that this class of geometric approaches yield interesting convergence rates under weak assumptions, an issue also left open in [7].

One issue for future work is to go deeper and finely tune the convergence rate to the surrogate to exhibit a ranking of permissible surrogates in terms of both upper and lower bounds. This might be a tedious task, as the state of the art involves case-by-case studies and was developed only for three of them [3], [23].

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for constructive comments that significantly helped to improve the manuscript. Both authors are supported by ANR *Blanc* project ANR-07-BLAN-0328-01 "Computational Information Geometry and Applications."

REFERENCES

[1] P. Bartlett, M. Jordan, and J.D. McAuliffe, "Convexity, Classification, and Risk Bounds," *J. Am. Statistical Assoc.,* vol. 101, pp. 138-156, 2006.
[2] P. Bartlett and M. Traskin, "Adaboost is Consistent," *Proc. Neural Information Processing Systems Conf.,* 2006.
[3] M.J. Kearns and Y. Mansour, "On the Boosting Ability of Top-Down Decision Tree Learning Algorithms," *J. Computer and System Sciences,* vol. 58, pp. 109-128, 1999.
[4] R.E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-Rated Predictions," *Proc. Conf. Computational Learning Theory,* pp. 80-91, 1998.
[5] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics,* vol. 28, pp. 337-374, 2000.
[6] V. Vapnik, *Statistical Learning Theory.* John Wiley, 1998.
[7] N. Murata, T. Takenouchi, T. Kanamori, and S. Eguchi, "Information Geometry of $\mathcal{U}$-Boost and Bregman Divergence," *Neural Computation,* vol. 16, pp. 1437-1481, 2004.
[8] P. Grünwald and P. Dawid, "Game Theory, Maximum Entropy, Minimum Discrepancy and Robust Bayesian Decision Theory," *Annals of Statistics,* vol. 32, pp. 1367-1433, 2004.
[9] M. Collins, R. Schapire, and Y. Singer, "Logistic Regression, Adaboost and Bregman Distances," *Proc. Conf. Computational Learning Theory,* pp. 158-169, 2000.
[10] R.E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-Rated Predictions," *Machine Learning,* vol. 37, pp. 297-336, 1999.
[11] A. Azran and R. Meir, "Data Dependent Risk Bounds for Hierarchical Mixture of Experts Classifiers," *Proc. Conf. Computational Learning Theory,* pp. 427-441, 2004.
[12] A. Banerjee, X. Guo, and H. Wang, "On the Optimality of Conditional Expectation As a Bregman Predictor," *IEEE Trans. Information Theory,* vol. 51, pp. 2664-2669, 2005.
[13] C. Gentile and M. Warmuth, "Linear Hinge Loss and Average Margin," *Proc. 1998 Conf. Advances in Neural Information Processing Systems,* pp. 225-231, 1998.
[14] D. Helmbold, J. Kivinen, and M. Warmuth, "Relative Loss Bounds for Single Neurons," *IEEE Trans. Neural Networks,* vol. 10, no. 6, pp. 1291-1304, Nov. 1999.
[15] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman Divergences," *J. Machine Learning Research,* vol. 6, no. 6, pp. 1705-1749, Nov. 2005.
[16] R. Nock and F. Nielsen, "A ℝeal Generalization of Discrete AdaBoost," *Artificial Intelligence,* vol. 171, pp. 25-41, 2007.
[17] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *Annals of Statistics,* vol. 26, pp. 1651-1686, 1998.
[18] L. Breiman, J.H. Freidman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees.* Wadsworth, 1984.
[19] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
[20] K. Matsushita, "Decision Rule, Based on Distance, for the Classification Problem," *Annals of the Inst. of Statistical Math.,* vol. 8, pp. 67-77, 1956.
[21] M. Warmuth, J. Liao, and G. Rätsch, "Totally Corrective Boosting Algorithms that Maximize the Margin," *Proc. Int'l Conf. Machine Learning,* pp. 1001-1008, 2006.
[22] J. Kivinen and M. Warmuth, "Boosting As Entropy Projection," *Proc. Conf. Computational Learning Theory,* pp. 134-144, 1999.
[23] R. Nock and F. Nielsen, "On Domain-Partitioning Induction Criteria: Worst-Case Bounds for the Worst-Case Based," *Theoretical Computer Science,* vol. 321, pp. 371-382, 2004.
[24] C. Henry, R. Nock, and F. Nielsen, "ℝeal Boosting *a la Carte* with an Application to Boosting Oblique Decision Trees," *Proc. 21st Int'l Joint Conf. Artificial Intelligence,* pp. 842-847, 2007.
[25] C.L. Blake, E. Keogh, and C.J. Merz, "UCI Repository of Machine Learning Databases," http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

Richard Nock received the agronomical engineering degree from Agro Montpellier, France, in 1993, the MSc degree in computer science in 1993 the PhD degree in computer science in 1998, and an accreditation to lead research in computer science (HDR) in 2002, all from the University of Montpellier II, France. He joined the Universite Antilles-Guyane in 1998, where he is currently a full professor of computer science. His research interests include machine learning, data mining, computational complexity, and image processing. He received the Best Paper Award at the European Conference on Artificial Intelligence in 2006. His research is being funded by ANR (Young Researcher, White and Thematic programs).

Frank Nielsen defended his PhD on adaptive computational geometry prepared at INRIA Sophia-Antipolis, France. In 1997, he served in the army as a scientific member in the Computer Science Laboratory (LIX) of the Ecole Polytechnique, France's leading Engineering School. In 1998, he joined Sony Computer Science Laboratories, Inc., Tokyo, first as a researcher, and then as a senior researcher. Since 2007, he is also a full professor of computer science at Ecole Polytechnique. His current research interests include geometry, vision, learning, graphics, and optimization. He recently published the book *Visual Computing: Geometry, Graphics and Vision* (2005), and won the best paper award at the European Conference on Artificial Intelligence in 2006. He is a member of the IEEE.