On the Efficient Minimization of Convex Surrogates in Supervised Learning

Richard Nock[*] Frank Nielsen[†]

Abstract

Bartlett et al (2006) recently proved that a ground condition for convex surrogates, classification calibration, ties up the minimization of the surrogates and classification risks, and left as important open problems the algorithmic questions about the minimization of these surrogates. Our paper gives an answer for a wide subset of these surrogates that we call "balanced surrogates", a set with popular members (logistic loss, squared loss), that contains all surrogates meeting three important requirements about classification. We propose an algorithm that fits linear separators to the minimization of any such surrogate, with guaranteed convergence bounds under a so-called "Weak Learning Assumption", a generalization of the one that grounds celebrated boosting algorithms. Experiments on more than 50 readily available domains of 10 flavors of the algorithm display the performances of new surrogates.

1. Introduction

A very active supervised learning trend has been flourishing over the last decade: it studies functions known as *surrogates* — upperbounds of the empirical risk, generally with particular convexity properties —, whose minimization remarkably impacts on empirical / true risks minimization [2, 6] (and many others). Surrogates play fundamental roles in some of the most successful supervised learning algorithms, including AdaBoost [8, 9], additive logistic regression [4], decision tree induction [6], Support Vector Machines. As their popularity has been rapidly spreading, some authors have begun to stress the need to set in order surrogates, and better understand their properties as wholes. Out of the rationales of any kind that can be found, statistical approaches have so far encompassed the others and explicitly left some of them, like the algorithmic question, as important problems to settle [2]. In this paper, we provide an answer for a large subclass of these surrogates, that we call *balanced convex surrogates* (BCS). We show that this set, which contains surrogates built from the logistic and squared losses, coincides with the set whose members meet 3 of the most common requirements for such functions in supervised learning: lower-boundedness, the optimality of conditional probabilities from the decision standpoint, and symmetries in the cost matrix. We provide a minimization algorithm that works for any BCS, ULS, with the key property that it meets Boosting-type convergence bounds to reach the minimum of the BCS under a weak learning assumption familiar to boosting algorithms [8]. The relevance of this result is also experimental: more freedom to choose surrogates means more space for domain-specific tunings. We provide such experiments on 10 flavors of ULS on a wide benchmark of 52 domains, report new challengers for popular surrogates, and sketch possible domain-specific tuning strategies. Section 2 gives definitions. Section 3 presents BCS. Section 4 presents ULS; Section 5 gives experiments.

2 Preliminary definitions

Unless otherwise stated, bold-faced variables like $\boldsymbol{w}$ denote vectors (components are $w_i, i = 1, 2, ...$), 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^+ (resp. c^-) is the *positive* class (resp. *negative* class). Classes are abstracted by a bijective mapping to one of two other sets:

$$c \in \{c^-, c^+\} \leftrightharpoons y^* \in \{-1, +1\} \leftrightharpoons y \in \{0, 1\} \ . \quad (1)$$

The convention is $c^+ \rightleftharpoons +1 \rightleftharpoons 1$ and $c^- \rightleftharpoons -1 \rightleftharpoons 0$. We thus have three distinct notations for an example: $(\boldsymbol{o}, c)$, $(\boldsymbol{o}, y^*)$, $(\boldsymbol{o}, y)$. We suppose given a set of m examples, $\mathcal{S} = \{(\boldsymbol{o}_i, c_i), i = 1, 2, ..., m\}$. We wish to build a *classifier* H, which can either be a function $H : \mathcal{O} \rightarrow \mathbb{O} \subseteq \mathbb{R}$ (hereafter, $\mathbb{O}$ is assumed to be symmetric with respect to 0), or a function $H : \mathcal{O} \rightarrow [0, 1]$.

[*] CEREGMIA — U. Antilles-Guyane, 97275 Schoelcher, France. Email: rnock@martinique.univ-ag.fr

[†] LIX — Ecole Polytechnique, 91128 Palaiseau, France. Email: nielsen@lix.polytechnique.fr

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

Following a convention of [3], we 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* which quantifies pointwise disagreements:

$$\varepsilon(\mathcal{S}, H) \doteq \sum_i \ell(c_i, H(\boldsymbol{o}_i)) \ . \quad (2)$$

The fundamental loss is the *0/1 loss*, $\ell^{0/1}(c, H)$ (to ease readability, the second argument is written H instead of $H(\boldsymbol{o})$), which takes on two forms depending on $\mathrm{im}(H)$:

$$\ell^{0/1}_{\mathbb{R}}(y^*, H) \doteq 1_{y^* \neq \sigma \circ H} \text{ if } \mathrm{im}(H) = \mathbb{O} \ , \quad (3)$$
$$\ell^{0/1}_{[0,1]}(y, H) \doteq 1_{y \neq \tau \circ H} \text{ if } \mathrm{im}(H) = [0,1] \ . \quad (4)$$

The following notations are introduced in (3-4), and shall be used wherever needed: for a clear distinction of the output of H, we put in index to ℓ and ε an indication of the loss' domain of parameters: $\mathbb{R}$, meaning it is actually some $\mathbb{O} \subseteq \mathbb{R}$, or $[0,1]$. The exponent to ℓ gives the indication of the loss name. Finally, 1_π is the indicator variable that takes value 1 iff predicate π is **true**, and 0 otherwise; $\sigma : \mathbb{R} \rightarrow \{-1, +1\}$ is $+1$ iff $x \geq 0$ and -1 otherwise; $\tau : [0,1] \rightarrow \{0,1\}$ is 1 iff $x \geq 1/2$, and 0 otherwise. Both losses $\ell_{\mathbb{R}}$ and $\ell_{[0,1]}$ are defined simultaneously via popular transforms on H, such as the *logit* transform $\mathrm{logit}(p) \doteq \log(p/(1-p)), \forall p \in [0,1]$ [4]. We have indeed $\ell^{0/1}_{[0,1]}(y, H) = \ell^{0/1}_{\mathbb{R}}(y^*, \mathrm{logit}(H))$ and $\ell^{0/1}_{\mathbb{R}}(y^*, H) = \ell^{0/1}_{[0,1]}(y, \mathrm{logit}^{-1}(H))$. 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 mapped back to $[0,1]$. In supervised learning, the objective is to carry out the minimization of the expectation of the 0/1 loss 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 simply (2) with the 0/1 loss [3]: $\varepsilon^{0/1}(\mathcal{S}, H) \doteq \sum_i \ell^{0/1}(c_i, H(\boldsymbol{o}_i))$. The main classifiers we investigate are linear separators (LS). 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 $\alpha_t \in \mathbb{R}$.

3 Balanced Convex Surrogates

It has been found over the last decade that $\varepsilon^{0/1}(\mathcal{S}, H)$ can be computationally efficiently minimized if we rather focus on the minimization of a *surrogate risk* [2]. This is a function $\varepsilon(\mathcal{S}, H)$ as in (2), whose *surrogate loss* satisfies:

$$\ell^{0/1}(c, H(\boldsymbol{o})) \leq \ell(c, H(\boldsymbol{o})) \ . \quad (5)$$

Three of them are particularly important; they are defined via the following surrogate losses:

$$\ell^{\log}_{\mathbb{R}}(y^*, H) \doteq \log(1 + \exp(-y^* H)) \ , \quad (6)$$
$$\ell^{\mathrm{sqr}}_{\mathbb{R}}(y^*, H) \doteq (1 - y^* H)^2 \ , \quad (7)$$
$$\ell^{\mathrm{hinge}}_{\mathbb{R}}(y^*, H) \doteq \max\{0, 1 - y^* H\} \ . \quad (8)$$

$\phi(x)$	a_ϕ	$\mathrm{im}(\nabla_{\overline{\phi}})$ $\supseteq \mathrm{im}(H)$	$F_\phi(y^* H)$ $= (\overline{\phi}^\star(-y^* H) - a_\phi)/b_\phi$	$\hat{\mathbf{Pr}}[c = c^+ \vert H; \boldsymbol{o}]$ $= \nabla^{-1}_{\overline{\phi}}(H)$
(11)	μ	$\mathbb{R}$	$\frac{-y^* H + \sqrt{(1-\mu)^2 + (y^* H)^2}}{1-\mu}$	$\frac{1}{2} + \frac{H}{2\sqrt{(1-\mu)^2 + H^2}}$
(12)	0	$\mathbb{R}$	$-y^* H + \sqrt{1 + (y^* H)^2}$	$\frac{1}{2} + \frac{H}{2\sqrt{1+H^2}}$
(13)	0	$\mathbb{R}$	$\log(1 + \exp(-y^* H))$	$\frac{\exp(H)}{1+\exp(H)}$
(14)	0	$[-1, 1]$	$(1 - y^* H)^2$	$\frac{1}{2} + \frac{H}{2}$

Table 1. Correspondence between permissible functions, the corresponding BCL**s and the matching** $[0,1]$ **predictions.**

(6) is the logistic loss, (7) is the squared loss and (8) is hinge loss. To state the class BCS, we need some preliminary definitions related to convex analysis. The *Legendre conjugate* $\psi^\star$ of some strictly convex and differentiable function ψ is:

$$\psi^\star(x) \doteq \sup_{x' \in \mathrm{int}(\mathbb{X})} \{x x' - \psi(x')\} \ . \quad (9)$$

Because of the strict convexity of ψ, the Legendre conjugate can be explicitly computed: $\psi^\star(x) \doteq x \nabla^{-1}_{\psi}(x) - \psi(\nabla^{-1}_{\psi}(x))$. $\psi^\star$ is also strictly convex and differentiable. A function $\phi : [0,1] \rightarrow \mathbb{R}_+$ is called permissible iff it is differentiable on $(0,1)$, strictly concave, symmetric about $x = 1/2$, and with $\phi(0) = \phi(1) = a_\phi \geq 0$. We let $b_\phi \doteq \phi(1/2) - a_\phi > 0$.

Definition 1 *Let ϕ permissible and $\overline{\phi} \doteq -\phi$. The Balanced Convex Loss (*BCL*) with generator ϕ, F_ϕ, is: $F_\phi(x) \doteq (\overline{\phi}^\star(-x) - a_\phi)/b_\phi$.*

Any surrogate risk built from a BCL is called a Balanced Convex Surrogates (BCS). All BCL share a common shape. Indeed, it is not hard to show that the asymptotes of any BCL can be summarized as:

$$\underline{\ell}(x) = x(\sigma(x) - 1)/(2 b_\phi) \ . \quad (10)$$

When $b_\phi = 1$, this is the linear hinge loss [5], a generalization of (8) for which $x \doteq y^* H - 1$. Thus, while hinge loss is not a BCL, it defines the limit behavior of any BCL (see Figure 1). Below are examples of permissible functions ϕ:

$$\phi_\mu(x) \doteq \mu + (1-\mu)\sqrt{x(1-x)} \ , \forall \mu \in (0,1) \quad (11)$$
$$\phi_{\mathrm{M}}(x) \doteq \sqrt{x(1-x)} \ , \quad (12)$$
$$\phi_{\mathrm{Q}}(x) \doteq -x \log x - (1-x)\log(1-x) \ , \quad (13)$$
$$\phi_{\mathrm{B}}(x) \doteq x(1-x) \ . \quad (14)$$

When scaled so that $\phi(1/2) = 1$, some confound with popular choices: (14) with Gini index, (13) with the Bit-entropy, and (12) with Matsushita's error [6, 7]. Table 1 (first four columns) gives the expressions of F_ϕ along with the $\mathrm{im}(H) = \mathbb{O} \subseteq \mathbb{R}$ allowed by the BCL, for the permissible functions in (11) — (14). Fig. 1 (right) gives a typical shape plot for $\nabla_{\overline{\phi}}$, similar to those of (11) — (13).

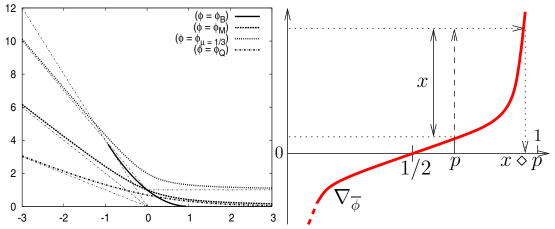

Figure 1. Left: bold curves are plots of $\overline{\phi}^{\star}(-x)$ for ϕ in (11) — (14); thin dotted half-lines are its asymptotes. Right: a typical $\nabla_{\overline{\phi}}$ (bold red, symmetric around point $(1/2, 0)$), with the Legendre dual $x \diamond p$ shown, a concept extensively used in ULS.

For any strictly convex function $\psi : \mathbb{X} \rightarrow \mathbb{R}$ differentiable on $\mathrm{int}(\mathbb{X})$, the *Bregman Loss Function* (BLF, [1]) D_ψ with generator ψ is:

$$D_\psi(x||x') \doteq \psi(x) - \psi(x') - (x - x')\nabla_\psi(x') \quad (15)$$

The following Lemma states an important relationship that is easy to check.

Lemma 1 *$\forall$ F_ϕ a* BCL*, $D_{\overline{\phi}}(y||\nabla_{\overline{\phi}}^{-1}(H)) = b_\phi F_\phi(-y^* H)$.*

Lemma 1 is important because it ties real predictions (right) with their *matching* $[0, 1]$ predictions (left). Following notations in (6) — (8), we can write any BCL as $\ell^{\phi}_{\mathbb{R}}(y^*, H) \doteq F_\phi(-y^* H)$. In fact, BCL matches the set of losses that satisfy the main requirements about losses used in machine learning. This is a very strong rationale for this set. Consider the following requirements about loss $\ell_{[0,1]}(y, H)$ (with $\mathrm{im}(H) \subseteq [0, 1]$):

- **(R1)** *The loss is lower-bounded.* $\exists z \in \mathbb{R}$ such that $\inf_{y,H} \ell_{[0,1]}(y, H) = z$.
- **(R2)** *Conditional probabilities are a generalized Bayes rule.* Consider a singleton domain $\mathcal{O} = \{\boldsymbol{o}\}$. Then, the best (constant) prediction is: $\arg\min_{x\in[0,1]} \varepsilon_{[0,1]}(\mathcal{S}, x) = \mathbf{Pr}[c = c^+|\boldsymbol{o}] \in [0, 1]$.
- **(R3)** *The loss is symmetric* as follows: $\forall y \in \{0, 1\}$, $\forall H \in [0, 1]$, $\ell_{[0,1]}(y, H) = \ell_{[0,1]}(1 - y, 1 - H)$.

R1 is standard. **R2** may be viewed as some consistency requirement for the surrogate to be minimized as $p(.)$ defines Bayes classifier. **R3** 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. For these domains indeed, it is assumed $\ell_{[0,1]}(1, 0) = \ell_{[0,1]}(0, 1)$. The following Lemma establishes the basis for the rationale (proof involves Theorem 3 in [1]).

Lemma 2 *Assume $\mathrm{im}(H) \subseteq [0, 1]$. Loss $\ell_{[0,1]}(., .)$ is properly defined and satisfies requirements **R1**, **R2**, **R3** iff $\ell_{[0,1]}(y, H) = z + D_{\overline{\phi}}(y||H)$ for some permissible ϕ.*

ϕ is thus the "signature" of the BCL/BCS. In the next Section, we show how to efficiently minimize any BCS.

```
Algorithm 1: Algorithm ULS($M, \phi$)
Input: $M \in \mathbb{R}^{m\times T}$, permissible function $\phi$;
Let $\boldsymbol{\alpha}_1 \leftarrow \boldsymbol{0}$; Let $\boldsymbol{w}_0 \leftarrow (1/2)\boldsymbol{1}$;
for $j = 1, 2, ...J$ do
    $\boldsymbol{w}_j \leftarrow (M\boldsymbol{\alpha}_j) \diamond \boldsymbol{w}_0$ ;   (16)
    Let $\mathcal{T}_j \subseteq \{1, 2, ..., T\}$; let $\boldsymbol{\delta}_j \leftarrow \boldsymbol{0}$;
    $\forall t \in \mathcal{T}_j$, pick $\delta_{j,t}$ such that:
        $\sum_{i=1}^{m} m_{it}((M\boldsymbol{\delta}_j) \diamond \boldsymbol{w}_j)_i = 0$ ;   (17)
    Let $\boldsymbol{\alpha}_{j+1} \leftarrow \boldsymbol{\alpha}_j + \boldsymbol{\delta}_j$;
Output: $H(\boldsymbol{x}) \doteq \sum_{t=1}^{T} \alpha_{J+1,t} h_t(\boldsymbol{x}) \in$ LS
```

4. ULS

Let $H \in$ LS, and suppose that the permissible function ϕ is such that $\mathrm{im}(\nabla_{\overline{\phi}}) = \mathbb{R}$ (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

$$x \diamond p \quad \doteq \quad \arg_{p'\in[0,1]} \sup\{xp' - D_{\overline{\phi}}(p'||p)\} \ .(18)$$

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 [3]. The technical look and feel of (18) hides an appealing representation, given in Figure 1 and explain below.

Because ϕ is permissible, the Legendre dual is unique and it is always in $[0, 1]$. We follow the setting of [3] and suppose that we have T features h_t $(t = 1, 2, ..., T)$ known in advance, the problem thus reducing to the computation of the leveraging coefficients. We define $m \times T$ matrix M with $m_{it} \doteq -y_i^* h_t(\boldsymbol{o}_i)$. Given leveraging coefficients vector $\boldsymbol{\alpha} \in \mathbb{R}^T$, we thus get:

$$-y_i^* H(\boldsymbol{o}_i) \quad = \quad (M\boldsymbol{\alpha})_i \ . \qquad (19)$$

Armed with these notations, algorithm ULS above provides a learning algorithm that provably minimize any BCS on any matrix M.

The explanation of the Legendre dual in ULS follows from (16). Consider example $(\boldsymbol{o}_i, y_i)$, and its weight update, $w_{j,i} \leftarrow (M\boldsymbol{\alpha}_j)_i \diamond w_{0,i} = (-y_i^* H(\boldsymbol{o}_i)) \diamond w_{0,i}$. Fix $p = w_{0,i}$ and $x = -y_i^* H(\boldsymbol{o}_i)$ in Figure 1. It comes that the new weight of the example is larger iff $x > 0$, *i.e.* iff the example is given the *wrong* class by H. This characteristic is one of the most popular of formal boosting algorithms like AdaBoost [8, 9]. It turns

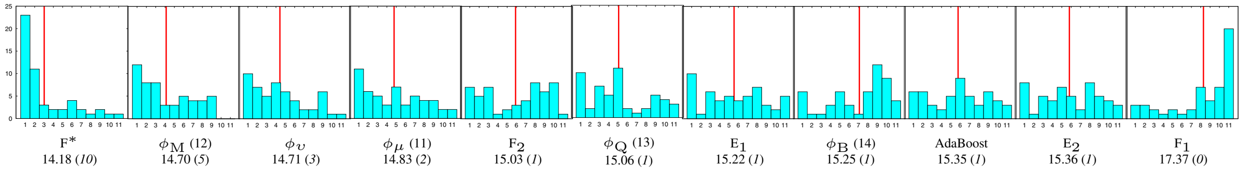

Figure 2. Summary of our results over the 52 domains for the 11 algorithms ($l = 2, r = 10$). Vertical (red) bars show the average rank over all domains.

out that ULS also generalizes the most important property of boosting algorithms, which relies on a so-called "Weak Learning Assumption" (WLA) [9]. To state the WLA, we define $Z_j \doteq ||\boldsymbol{w}_j||_1$ ($||.||_k$ is the L_k norm). The WLA is:

$$(\textbf{WLA}) \forall j, \exists \gamma_j > 0 : \left| \frac{1}{|\mathcal{T}_j|} \sum_{t \in \mathcal{T}_j} \frac{1}{Z_j} \sum_{i=1}^{m} m_{it} w_{j,i} \right| \geq \gamma_j \quad (20)$$

The WLA in (20) tells that the average *edge* of the features in $\mathcal{T}_j$ exceeds random (for which $\gamma_j = 0$) by a guaranteed — even if small — amount; it is a generalization of conventional WLAs [8]. To state the following Theorem, we need few more definitions. Let $\boldsymbol{m}_t$ denote the t^{th} 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 $a_\varphi \doteq \min_{x \in (0,1)} \mathrm{d}^2\overline{\phi}(x)/\mathrm{d}x^2$.

Theorem 1 *For any* BCS *with signature ϕ, for any M, ULS achieves the minimum of the* BCS *on M in at most $J = \lceil \frac{4mb_\phi a_{\boldsymbol{m}}^2}{a_\varphi a_Z^2 a_\gamma^2} \rceil$ steps.*

As another important property, we can show that (17) has always a solution in the non-trivial cases (when no feature h_t has zero empirical risk), so that ULS is always guaranteed to work.

5. Experiments

We have compared against each other 10 flavors of ULS + AdaBoost [9], on a benchmark of 52 domains (49 from the UCI repository). True risks are estimated via stratified 10-fold cross validation; ULS is ran for r (fixed) features h_t, each of which is a boolean rule: **If** Monomial **then** Class= ±1 **else** Class = ∓1, with at most l (fixed) literals, induced following the greedy minimization of the BCS at hand. Leveraging coefficients (17) are approximated up to 10^{-10} precision. Figure 2 summarizes the results. Histograms are ordered from left to right in increasing average true risk over all domains (shown below histograms). The *italic* numbers give, for each algorithm, the number of algorithms it *beats* according to a Student paired t-test over all domains with .1 threshold probability. Out of the 10 flavors of ULS, the first four flavors pick ϕ in (11) — (14). The fifth uses another generalization of (12): $\phi_\upsilon(x) \doteq (x(1-x))^\upsilon$, $\forall \upsilon \in (0,1)$. The last five adaptively tune the BCS at hand out-of-a-bag of BCS. The first four fit the BCS at *each stage* of the inner loop (**for** j ...) of ULS. Two (noted "$F_.$") pick the BCS which minimizes the empirical risk in the bag; two others (noted "$E_.$") pick the BCS which maximizes the current edge. There are two different bags corresponding to four permissible functions each: the first (index "1") contains (11) — (14), the second (index "2") contains (11) — (13) and ϕ_υ. We wanted to evaluate (14) because it forces to renormalize the leveraging coefficients in H each time it is selected, to ensure that the output of H lies in $[-1, 1]$. The last adaptive flavor, F^*, "externalizes" the choice of the BCS: it selects for each fold the BCS which yields the smallest empirical risk in a bag corresponding to five ϕ: (11) — (13) and ϕ_υ.

All results in Figure 2 advocate for the superiority of F^* against all other approaches. Furthermore, stronger concave regimes for ϕ (e.g. (12)) tend to improve performances, a fact previously remarked for decision tree induction in [6]. In the light of these results, Matsushita's BCL (built from (12)) appears to be a serious alternative to the Logistic loss.

Acknowledgments : authors supported by ANR "Blanc" programme ANR-07-BLAN-0328-01.

References

[1] A. Banerjee, X. Guo, and H. Wang. On the optimality of conditional expectation as a bregman predictor. *IEEE Trans. on Information Theory*, 51:2664–2669, 2005.

[2] P. Bartlett, M. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *JASA*, 101:138–156, 2006.

[3] M. Collins, R. Schapire, and Y. Singer. Logistic regression, adaboost and Bregman distances. In *COLT'00*, pages 158–169, 2000.

[4] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic Regression : a Statistical View of Boosting. *Ann. of Stat.*, 28:337–374, 2000.

[5] C. Gentile and M. Warmuth. Linear hinge loss and average margin. In *NIPS*11*, pages 225–231, 1998.

[6] M. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. *JCSS*, 58:109–128, 1999.

[7] K. Matsushita. Decision rule, based on distance, for the classification problem. *Ann. ISM*, 8:67–77, 1956.

[8] R. Nock and F. Nielsen. A $\mathbb{R}$eal Generalization of discrete AdaBoost. *Artificial Intelligence*, 171:25–41, 2007.

[9] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *COLT'98*, pages 80–91, 1998.