Supervised Learning: No Loss No Cry

Richard Nock [1 2] **Aditya Krishna Menon** [3]

Abstract

Supervised learning requires the specification of a loss function to minimise. While the theory of admissible losses from both a computational and statistical perspective is well-developed, these offer a panoply of different choices. In practice, this choice is typically made in an *ad hoc* manner. In hopes of making this procedure more principled, the problem of *learning the loss function* for a downstream task (e.g., classification) has garnered recent interest. However, works in this area have been generally empirical in nature.

In this paper, we revisit the SLISOTRON algorithm of Kakade et al. (2011) through a novel lens, derive a generalisation based on Bregman divergences, and show how it provides a principled procedure for learning the loss. In detail, we cast SLISOTRON as learning a loss from a family of composite square losses. By interpreting this through the lens of *proper losses*, we derive a generalisation of SLISOTRON based on Bregman divergences. The resulting BREGMANTRON algorithm jointly learns the loss along with the classifier. It comes equipped with a simple guarantee of convergence for the loss it learns, and its set of possible outputs comes with a guarantee of agnostic approximability of Bayes rule. Experiments indicate that the BREGMANTRON outperforms the SLISOTRON, and that the loss it learns can be minimized by other algorithms for different tasks, thereby opening the interesting problem of *loss transfer* between domains.

1. Introduction

Computationally efficient supervised learning essentially started with the PAC framework of Valiant (1984), in which the goal was to learn in polynomial time a function being able to predict a label (or class, among two possible) for i.i.d. inputs. The initial loss, whose minimization enforces the accurate prediction of labels, was the binary *zero-one loss* which returns 1 iff a mistake is made.

The zero-one loss was later progressively replaced in learning algorithms for tractability reasons, including its non-differentiability and the structural complexity of its minimization (Kearns & Vazirani, 1994; Auer et al., 1995). From the late nineties, a zoo of losses started to be used for tractable machine learning (ML), the most popular ones built from the *square loss* and the *logistic loss*. Recently, there has been a significant push to widen even more the choice of loss; to pick a few, see Grabocka et al. (2019); Kakade et al. (2011); Liu et al. (2019); Mei & Moura (2018); Nock & Nielsen (2008; 2009); Reid & Williamson (2010); Siahkamari et al. (2019); Streeter (2019); Sypherd et al. (2019).

With a few exceptions, seldom do such works ground reasons for change of the loss outside of tractability at large, be it algorithmic or statistical – with for example the introduction of classification calibration, which studies conditions on losses complying with the accurate prediction of labels (Bartlett et al., 2006). It turns out that statistics and Bayes decision theory give a precise reason, one which has long been the object of philosophical and formal debates (de Finetti, 1949). It starts from a simple principle:

Bayes rule is optimal for the loss at hand,

a property known as *properness* (Savage, 1971). Then comes a less known subtlety: a proper loss as commonly used for real-valued prediction, such as the square and logistic loss, involves an implicit *canonical link* (Reid & Williamson, 2010) function that maps class probabilities (such as the output of Bayes rule) to real values. This is exemplified by the sigmoid (inverse) link in deep learning.

Supervised learning in a Bayesian framework can thus be more broadly addressed by learning a classifier *and* a link for the domain at hand, *which implies learning a proper canonical loss* with the classifier. This loss, if suitably expressed, can be used for training. This kills two birds in one shot: we get access not just to real valued predictions, but also a way to embed them into class probability estimates

[1]Data61 (Australia) [2]The Australian National University (Australia) [3]Google Research (USA). Correspondence to: <richard.nock@data61.csiro.au,adityakmenon@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

via the inverse link: we directly learn to estimate Bayes rule.

A large number of papers, especially recently, have tried to push forward the problem of learning the loss, including *e.g.* (Grabocka et al., 2019; Liu et al., 2019; Mei & Moura, 2018; Siahkamari et al., 2019; Streeter, 2019; Sypherd et al., 2019), *but* none of those alludes to properness to ground the choice of the loss, therefore taking the risk of fitting a loss whose (unknown) optima may fail to contain Bayes rule. To the best of our knowledge, Nock & Nielsen (2008) is the first paper grounding the search of the loss within properness and Kakade et al. (2011) brings the first algorithm (SLISOTRON) and associated theoretical results for fitting the link — though subject to restrictive assumptions on Bayes rule and on the target distribution, the risk to fit probabilities outside $[0, 1]$, and finally falling short of showing convergence that would comply with the classical picture of ML, either for training or generalization.

Our major contribution is a new algorithm, the BREGMANTRON (Algorithm 2), a generalisation of the SLISOTRON (Kakade et al., 2011) to learn proper canonical losses. BREGMANTRON exploits two dual views of proper losses, guarantees class probability estimates in $[0, 1]$, and uses Lipschitz constraints that can be tuned at runtime.

Our formal contribution includes a simple convergence guarantee for this algorithm which alleviates all assumptions on the domain and Bayes rule in (Kakade et al., 2011). Our result shows that convergence happens as a function of the discrepancy between our estimate and the true value of of the mean operator – a sufficient statistic for the class (Patrini et al., 2014). As the discrepancy converges to zero, the estimated (link, classifier) by the BREGMANTRON converges to a stable output. To come to this result, we pass through an intermediate step in which we show a particular explicit form for any differentiable proper composite loss, of a Bregman divergence (Bregman, 1967), which are canonical distortion measures. To save space, all proofs are given in a supplementary material, denoted SI.

2. Definitions and notations

The following shorthands are used: $[n] \doteq \{1, 2, ..., n\}$ for $n \in \mathbb{N}_*$, for $z \geq 0, a \leq b \in \mathbb{R}$, denote $z \cdot [a, b] \doteq [za, zb]$ and $z + [a, b] \doteq [z + a, z + b]$. We also let $\overline{\mathbb{R}} \doteq [-\infty, \infty]$. In (batch) supervised learning, one is given a training set of m examples $S \doteq \{(\boldsymbol{x}_i, y_i^*), i \in [m]\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is an observation ($\mathcal{X}$ is called the domain: often, $\mathcal{X} \subseteq \mathbb{R}^d$) and $y_i^* \in \mathcal{Y} \doteq \{-1, 1\}$ is a label, or class. The objective is to learn a *classifier* $h : \mathcal{X} \rightarrow \mathbb{R}$ which belongs to a given set $\mathcal{H}$. The goodness of fit of some h on S is evaluated by a *loss*.

▷ **Losses**: A loss for binary class probability estimation (Buja et al., 2005) is some $\ell : \mathcal{Y} \times [0, 1] \rightarrow \overline{\mathbb{R}}$ whose expression can be split according to *partial* losses ℓ_1, ℓ_{-1},

$$\ell(y^*, u) \doteq [\![y^* = 1]\!] \cdot \ell_1(u) + [\![y^* = -1]\!] \cdot \ell_{-1}(u), \quad (1)$$

Its *conditional Bayes risk* function is the best achievable loss when labels are drawn with a particular positive base-rate,

$$\underline{L}(\pi) \quad \doteq \quad \inf_u \mathsf{E}_{\mathsf{Y} \sim \pi} \ell(\mathsf{Y}, u), \quad (2)$$

where $\Pr[\mathsf{Y} = 1] = \pi$. A loss for class probability estimation ℓ is ***proper*** iff Bayes prediction locally achieves the minimum everywhere: $\underline{L}(\pi) = \mathsf{E}_{\mathsf{Y}} \ell(\mathsf{Y}, \pi), \forall \pi \in [0, 1]$, and strictly proper if Bayes is the unique minimum. Fitting a prediction $h(\boldsymbol{x}) \in \mathbb{R}$ into some $u \in [0, 1]$ as required in (1) is done via a *link function*.

▷ **Links, composite and canonical proper losses**. A link $\psi : [0, 1] \rightarrow \mathbb{R}$ allows to connect real valued prediction and class probability estimation. A loss can be augmented with a *link* to account for real valued prediction, $\ell_\psi(y^*, z) \doteq \ell(y^*, \psi^{-1}(z))$ with $z \in \mathbb{R}$ (Reid & Williamson, 2010). There exists a particular link uniquely defined[1] for any proper differentiable loss, the *canonical link*, as: $\psi \doteq -\underline{L}'$ (Reid & Williamson, 2010, Section 6.1). We note that the differentiability condition can be removed (Reid & Williamson, 2010, Footnote 6). As an example, for log-loss we find the link $\psi(u) = \log \frac{u}{1-u}$, with inverse the well-known sigmoid $\psi^{-1}(z) = (1 + e^{-z})^{-1}$. A canonical proper loss is a proper loss using the canonical link.

▷ **Convex surrogates**. When the loss is proper canonical and symmetric ($\ell_1(u) = \ell_{-1}(1 - u), \forall u \in (0, 1)$), it was shown in Nock & Nielsen (2008; 2009) that there exists a convenient dual formulation amenable to direct minimization with real valued classifiers: a *convex surrogate* loss

$$F_\ell(z) \quad \doteq \quad (-\underline{L})^\star(-z), \quad (3)$$

where $\star$ denotes the Legendre conjugate of F, $F^\star(z) \doteq \sup_{z' \in \mathrm{dom}(F)} \{zz' - F(z')\}$ (Boyd & Vandenberghe, 2004). For simplicity, we just call F_ℓ the convex surrogate of ℓ. The logistic, square and Matsushita losses are all surrogates of proper canonical and symmetric losses. Such functions are called surrogates since they all define convenient upper-bounds of the 0/1 loss. Any proper canonical and symmetric loss has $\ell(y^*, z) \propto F_\ell(y^* z)$ so both dual forms are equivalent in terms of minimization (Nock & Nielsen, 2008; 2009).

▷ **Learning**. Given a sample S, we learn h by the empirical minimization of a proper loss on S that we denote $\ell_\psi(S, h) \doteq \mathbb{E}_i[\ell(y_i^*, \psi^{-1}(h(\boldsymbol{x}_i)))]$. We insist on the fact that minimizing any such loss does not just give access to a real valued predictor h: it *also* gives access to a class probability estimator given the loss (Nock & Nielsen, 2008, Section 5), (Nock & Williamson, 2019),

$$\Pr[\mathsf{Y} = 1 | \boldsymbol{x}; h, \psi] \quad \doteq \quad \psi^{-1}(h(\boldsymbol{x})), \quad (4)$$

[1] Up to multiplication or addition by a scalar (Buja et al., 2005).

so in the Bayesian framework, supervised learning can also encompass learning the link ψ of the loss as well. *If the loss is proper canonical, learning the link implies learning the loss*. As usual, we assume S sampled i.i.d. according to an unknown but fixed $\mathcal{D}$ and let $\ell_{.}(\mathcal{D}, h) \doteq \mathbb{E}_{S\sim\mathcal{D}}[\ell_{.}(S, h)]$.

3. Related work

Our problem of interest is learning not only a classifier, but also a *loss function* itself. A minimal requirement for the loss to be useful is that it is proper, i.e., it preserves the Bayes classification rule. Constraining our loss to this set ensures standard guarantees on the classification performance using this loss, e.g., using surrogate regret bounds.

Evidently, when choosing amongst losses, we must have a well-defined objective. We now reinterpret an algorithm of Kakade et al. (2011) as providing such an objective.

▷ **The SLISOTRON algorithm**. Kakade et al. (2011) considered the problem of learning a class-probability model of the form $\Pr(\mathsf{Y} = 1 \mid \boldsymbol{x}) = u(\boldsymbol{w}_*^\top \boldsymbol{x})$ where $u(\cdot)$ is a 1-Lipschitz, non-decreasing function, and $\boldsymbol{w}_* \in \mathbb{R}^d$ is a fixed vector. They proposed SLISOTRON, an iterative algorithm that alternates between gradient steps to estimate $\boldsymbol{w}_*$, and nonparametric *isotonic regression* steps to estimate u. SLISOTRON provably bounds the expected *square loss*, i.e.,

$$\ell^{\text{SQ}}_{\psi}(S, h) \quad = \quad \mathbb{E}_{\boldsymbol{x}\sim S}\left[\mathbb{E}_{y^*\sim S}[(y - \psi^{-1}(h(\boldsymbol{x})))^2|\boldsymbol{x}]\right] \quad (5)$$

where $h(\boldsymbol{x}) = \boldsymbol{w}^\top\boldsymbol{x}$ is a linear scorer and $y \doteq (y^* + 1)/2$. The square loss has $2\cdot\ell^{\text{SQ}}_1(u) \doteq (1-u)^2$, $2\cdot\ell^{\text{SQ}}_{-1}(u) \doteq u^2$, and conditional Bayes risk $2\cdot\underline{L}^{\text{SQ}}(u) \doteq u(1-u)$.

Observe now that the SLISOTRON algorithm can be interpreted as follows: we jointly learn a *classifier* $h \in \mathcal{H}$ and *composite link* ψ for the square loss $\ell \in \mathcal{L}$, as

$$\mathcal{L} \doteq \{(y, z) \mapsto (y - \psi^{-1}(z))^2 : \psi \text{ is 1-Lipschitz, invertible}\}.$$

That is, SLISOTRON can be interpreted as finding a classifier and a link via all compositions of the square loss with a 1-Lipschitz, invertible function. Kakade et al. (2011) in fact do not directly analyze (5) but a *lowerbound* that directly follows from Jensen's inequality:

$$\begin{aligned}\ell^{\text{SQ}}_{\psi}(S, h) &= \mathbb{E}_{\boldsymbol{x}\sim S}\left[\mathbb{E}_{y^*\sim S}[(y - \psi^{-1}(h(\boldsymbol{x})))^2|\boldsymbol{x}]\right], \\ &\geq \mathbb{E}_{\boldsymbol{x}\sim S}\left[(\mathbb{E}_{y\sim S}[y|\boldsymbol{x}] - \psi^{-1}\circ h(\boldsymbol{x}))^2\right]. \quad (6)\end{aligned}$$

This does not change the problem as the slack is the expected (per observation) variance of labels in the sample, a constant given S. We shall return to this point in the sequel.

Kakade et al. (2011) make an assumption about Bayes rule, $\mathbb{E}_{y^*\sim\mathcal{D}}[y|\boldsymbol{x}] = \psi^{-1}_{\text{OPT}}(\boldsymbol{w}_{\text{OPT}}^\top\boldsymbol{x})$ with ψ^{-1}_{OPT} Lipschitz and $\|\boldsymbol{w}_{\text{OPT}}\| \leq R$. Under such an assumption, it is shown that *there exists* an iteration $t = O((Rm/d)^{1/3})$ of the SLISOTRON with $\max\{\tilde{\ell^{\text{SQ}}}_{\psi}(S, h), \tilde{\ell^{\text{SQ}}}_{\psi}(\mathcal{D}, h)\} \leq \tilde{O}((dR^2/m)^{1/3})$ with high probability ($\tilde{\ell^{\text{SQ}}}$ is the lower-bound of the loss in (6)). Nothing is guaranteed outside this unknown "hitting" point, which we partially attribute to the lack of convergence results on training. Another potential downside from the Bayesian standpoint is that the estimates learned are not guaranteed to be in $[0, 1]$ by the isotonic regression as modeled.

▷ **Learning the loss**. Over the last decade, the problem of learning the loss has seen a considerable push for a variety of reasons: Sypherd et al. (2019) introduced a family of tunable classification calibrated losses, aimed at increasing robustness in classification. Mei & Moura (2018) formulated the generalized linear model using Bregman divergences, though no relationship with proper losses is made and the loss function used integrates several regularizers breaking properness; the formal results rely on several quite restrictive assumptions and the guarantees are loosened if the true composite link comes from a loss that is not strongly convex "enough". In Streeter (2019), the problem studied is in fact learning the regularized part of the logistic loss, with no approximation guarantee. In Grabocka et al. (2019), the goal is to learn a loss defined by a neural network, without reference to proper losses and no approximation guarantee. Such a line of work also appears in a slightly different form in Liu et al. (2019). In Siahkamari et al. (2019), the loss considered is mainly used for metric learning, but integrates Bregman divergences. No mention of properness is made. Perhaps the most restrictive part of the approach is that it fits piecewise linear divergences, which are therefore not differentiable nor strictly convex.

Interestingly, none of these recent references alludes to properness to constrain the choice of the loss. Only the modelling of Mei & Moura (2018) can be related to properness via Theorem 1 proven below. The problem of learning the loss was introduced as loss *tuning* in Nock & Nielsen (2008) (see also Reid & Williamson (2010)). Though a general boosting result was shown for any tuned loss following a particular construction on its Bayes risk, it was restricted to losses defined from a convex combination of a basis set and no insight on improved convergence rates was given.

4. Learning proper canonical losses

We now present BREGMANTRON, our algorithm to learn proper canonical losses by learning a link function. We proceed in two steps. We first show an explicit form to proper differentiable composite losses and then provide our approach, the BREGMANTRON.

▷ **Every proper differentiable composite loss is Bregman** Let $F : \mathbb{R} \to \mathbb{R}$ be convex differentiable. The Bregman

Algorithm 0 SLISOTRON

Input: sample $S = \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$, iterations $T \in \mathbb{N}_*$.
For $t = 0, 1, ..., T-1$
[Step 1] **If** $t = 0$ **Then** $\boldsymbol{w}_{t+1} = \boldsymbol{w}_1 = \boldsymbol{0}$ **Else** fit $\boldsymbol{w}_{t+1}$ using

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{1}{m}\sum_{i=1}^{m}(u_t(\boldsymbol{x}_i) - y_i) \cdot \boldsymbol{x}_i \quad (7)$$

[Step 2] order indexes in S so that $\boldsymbol{w}_{t+1}^\top \boldsymbol{x}_{i+1} \geq \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_i, \forall i \in [m-1]$;
[Step 3] let $\boldsymbol{z}_{t+1} \doteq \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_{i+1}$
[Step 4] fit next (inverse) link

$$u_{t+1} \leftarrow \mathrm{IsotonicReg}(\hat{\boldsymbol{z}}_{t+1}, S); \quad \text{//fitting of } u_{t+1} \text{ given } \boldsymbol{z}_{t+1} \quad (8)$$

Output: $u_T, \boldsymbol{w}_T$.

Figure 1: The SLISOTRON algorithm of Kakade et al. (2011).

divergence D_F with generator F is:

$$D_F(z\|z') \doteq F(z) - F(z') - (z - z')F'(z'). \quad (12)$$

Bregman divergences satisfy a number of convenient properties, many of which are going to be used in our results. In order not to laden the paper's body, we have summarized in SI (Section II) a factsheet of all the results we use.

Our first result gives a way to move between proper composite losses and Bregman divergences.

Theorem 1 *Let $\ell : \mathcal{Y} \times [0,1] \rightarrow \overline{\mathbb{R}}$ be differentiable and $\psi : [0,1] \rightarrow \mathbb{R}$ invertible. Then ℓ is a proper composite loss with link ψ iff it is a Bregman divergence with:*

$$\begin{aligned} \ell(y^*, h(\boldsymbol{x})) &= D_{-\underline{L}}(y\|\psi^{-1} \circ h(\boldsymbol{x})), \quad (13)\\ &= D_{(-\underline{L})^\star}(-\underline{L}' \circ \psi^{-1} \circ h(\boldsymbol{x})\|-\underline{L}'(y)), \end{aligned}$$

where $\underline{L}$ is the conditional Bayes risk defined in (2), and we remind the correspondence $y = (y^* + 1)/2$.

Though similar forms of this Theorem have been proven in the past in Cranko et al. (2019, Theorem 2), Nock & Nielsen (2008, Lemma 3), Reid & Williamson (2010, Corollary 13), Zhang (2004, Theorem 2.1), Savage (1971, Section 4), none fit exactly to the setting of Theorem 1, which is therefore of independent interest and proven in SI, Section III. We now remark that the approach of Kakade et al. (2011) in (6) in fact *cannot* be replicated for proper canonical losses in general: because any Bregman divergence is convex in its left parameter, we still have as in (6)

$$\ell_\psi(S, h) \geq \mathbb{E}_{\boldsymbol{x}\sim S}[D_{-\underline{L}}(\mathbb{E}_{y\sim S}[y|\boldsymbol{x}]\|\psi^{-1} \circ h(\boldsymbol{x}))],$$

but the slack in the generalized case can easily be found to be the expected Bregman information of the class (Banerjee et al., 2004), $\mathbb{E}_{\boldsymbol{x}\sim S}[I_{-\underline{L}}(\mathsf{Y}|\boldsymbol{x})]$, with $I_{-\underline{L}}(\mathsf{Y}|\boldsymbol{x}) = \mathbb{E}_{y\sim S}[-\underline{L}(y)|\boldsymbol{x}] + \underline{L}(\mathbb{E}_{y\sim S}[y|\boldsymbol{x}])$, which therefore depends on the loss at hand (which in our case is learned as well).

▷ **Learning proper canonical losses** We now focus on learning class probabilities unrestricted to all losses having the expression in (13), but with the requirement that we use the canonical link: $\psi \doteq -\underline{L}'$, thereby imposing that we learn the loss as well via its link. Being the central piece of our algorithm, we formally define this function alongside some key parameters that will be learned. Notably, we in fact learn for algorithmic convenience the *inverse* canonical link of a loss but in order not to laden the paper, we shall also refer to this function as a link for short. Its domain or image makes the distinction clear from context.

Definition 2 *A link u is a strictly increasing function with* $\mathrm{Im} u = [0,1]$, *for which there exists $-\infty \ll z_{\text{MIN}}, z_{\text{MAX}} \ll \infty$ and $0 < n \leq N$ such that (i) $u(z_{\text{MIN}}) = 0, u(z_{\text{MAX}}) = 1$ and (ii) $\forall z \leq z', n(z'-z) \leq u(z') - u(z) \leq N(z'-z)$.*

Notice that relaxing $z_{\text{MIN}}, z_{\text{MAX}} \in \overline{\mathbb{R}}$ (the closure of $\mathbb{R}$) and $n, N \in \overline{\mathbb{R}}_+$ would allow to encompass all invertible links, so definition 2 is not restrictive but rather focuses on simply computable links. Given the canonical link u, we let

$$U(z) \doteq \int_{z_{\text{MIN}}}^{z} u(t)\mathrm{d}t, \quad (14)$$

from which we obtain the convex surrogate $F_u(z) = U(-z)$ and conditional Bayes risk $\underline{L}_u(v) = -U^\star(v)$ for the proper loss $\ell^u_{y^*}(c) \doteq D_{-\underline{L}_u}(y\|c), \forall y \in \{0, 1\}$.

Algorithm 1 BREGMANTRON

Input: sample $S = \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$, iterations $T \in \mathbb{N}_*$.
Initialize $u_0(z) \doteq 0 \vee (1 \wedge (az + b))$;
For $t = 0, 1, ..., T - 1$
[Step 1] **If** $t = 0$ **Then** $\boldsymbol{w}_{t+1} = \boldsymbol{w}_1 = \boldsymbol{0}$ **Else** fit $\boldsymbol{w}_{t+1}$ using a gradient step towards:

$$\boldsymbol{w}^* \doteq \arg\min_{\boldsymbol{w}} \mathbb{E}_S[D_{U_t}(\boldsymbol{w}^\top \boldsymbol{x} \| u_t^{-1}(y))]. \quad \text{//proper canonical fitting of } \boldsymbol{w}_{t+1} \text{ given } \hat{\boldsymbol{y}}_t, u_t \tag{9}$$

[Step 2] order indexes in S so that $\boldsymbol{w}_{t+1}^\top \boldsymbol{x}_{i+1} \geq \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_i, \forall i \in [m-1]$;
[Step 3] fit $\hat{\boldsymbol{y}}_{t+1}$ by solving for global optimum (n_t, N_t chosen so that $0 < n_t \leq N_t$):

$$\begin{aligned} \hat{\boldsymbol{y}}_{t+1} &\doteq \arg\min_{\hat{\boldsymbol{y}}} \mathbb{E}_S[D_{U_t^\star}(\hat{y} \| \hat{\boldsymbol{y}}_t)] \quad \text{//proper composite fitting of } \hat{\boldsymbol{y}}_{t+1} \text{ given } \boldsymbol{w}_{t+1}, u_t \\ &\text{s.t.} \left\{ \begin{array}{l} \hat{y}_{i+1} - \hat{y}_i \in [n_t \cdot (\boldsymbol{w}_{t+1}^\top(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)), N_t \cdot (\boldsymbol{w}_{t+1}^\top(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i))] \ , \forall i \in [m-1] \\ \hat{y}_1 \geq 0, \hat{y}_m \leq 1 \end{array} \right. . \end{aligned} \tag{10}$$

[Step 4] fit next (inverse) link

$$u_{t+1} \leftarrow \text{FIT}(\hat{\boldsymbol{y}}_{t+1}, \boldsymbol{w}_{t+1}, S); \quad \text{//fitting of } u_{t+1} \text{ given } \boldsymbol{w}_{t+1}, \hat{\boldsymbol{y}}_{t+1} \tag{11}$$

Output: $u_T, \boldsymbol{w}_T$.

Figure 2: The BREGMANTRON algorithm.

Inline with Theorem 1, the loss we seek to minimize is

$$\ell(y^*, h(\boldsymbol{x})) = D_{U^\star}(y \| u \circ h(\boldsymbol{x})) = D_U(h(\boldsymbol{x}) \| u^{-1}(y)), \tag{15}$$

with $h(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ a linear classifier.

We present BREGMANTRON, our algorithm for fitting such losses in Figure 2. BREGMANTRON iteratively fits a sequence $u_0, u_1, ...$ of links and losses and $\boldsymbol{w}_0, \boldsymbol{w}_1, ...$ of classifiers. Here, $\wedge, \vee$ are shorthands for $\min, \max$ respectively and in Step 3, we have dropped the iteration in the optimization problem ($\hat{y}_i$ denotes $\hat{y}_{ti}$). Notice that Steps 1 and 3 exploit the two dual views of proper losses presented in §2.

Before analyzing BREGMANTRON, we make a few comments. First, in the initialization, we pick $z_{0\text{MAX}} - z_{0\text{MIN}} \doteq \Delta > 0$ and let $a \doteq 1/\Delta, b \doteq -z_{0\text{MIN}}/\Delta$.

Second, the choice of n_t, N_t is made iteration-dependent for flexibility reasons, since in particular the gradient step does not put an explicit limit on $\boldsymbol{w}_t$. However, there is an implicit constraint which is to ensure $n_t \leq 1/(\boldsymbol{w}_{t+1}^\top(\boldsymbol{x}_m - \boldsymbol{x}_1)) \leq N_t$ to get a non-empty feasible set for u_{t+1}.

Third, BREGMANTRON bears close similarity to SLISOTRON, but with two key points to note. In Step 1, we perform a gradient step to minimise a divergence between the current predictions and the labels; fixing a learning rate of $\eta = 1$, in fact reduces to the SLISOTRON update. Further, in Step 3, we perform fitting of $\hat{y}_{t+1}$ based on the *previous* estimates $\hat{y}_t$, rather than the observed labels themselves as per SLISOTRON. Our Step 3 can thus be seen as "Bregman regularisation" step, which ensures the predictions (and thus the link function) do not vary too much across iterates. Such stability ensures asymptotic convergence, but does mean that the initial choice of link can influence the rate of this convergence.

Finally, with $z_{(t+1)i} \doteq \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_i$, FIT can be summarized as:

[1] linearly interpolate between $(z_{(t+1)i}, \hat{y}_{(t+1)i})$ and $(z_{(t+1)(i+1)}, \hat{y}_{(t+1)(i+1)}), \forall i \in \{2, 3, m-2\}$,

[2] pick $z_{(t+1)\text{MIN}} \leq \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_1, z_{(t+1)\text{MAX}} \geq \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_m$ with:

$$\hat{y}_j \in [l_j, r_j], j \in \{1, m\}, \tag{16}$$

and linearly interpolate between $(z_{(t+1)\text{MIN}}, 0)$ and $(z_{(t+1)1}, \hat{y}_{(t+1)1})$, and $(z_{(t+1)m}, \hat{y}_{(t+1)m})$ and $(z_{(t+1)\text{MAX}}, 1)$.

Here, $r_1 \doteq n_t \cdot (\boldsymbol{w}_{t+1}^\top \boldsymbol{x}_1 - z_{(t+1)\text{MIN}}), l_1 \doteq N_t \cdot (\boldsymbol{w}_{t+1}^\top \boldsymbol{x}_1 - z_{(t+1)\text{MIN}}), r_m \doteq n_t \cdot (z_{(t+1)\text{MAX}} - \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_m), l_m \doteq N_t \cdot (z_{(t+1)\text{MAX}} - \boldsymbol{w}_{t+1}^\top \boldsymbol{x}_m)$. Figure 3 presents a simple example of FIT in the BREGMANTRON. Our theoretical results depend on the sample-wise variation conditions following from (10). As such, the piecewise affine interpolation in FIT, chosen for its simplicity, can be replaced by other interpolation procedures tailored to other constraints, such as second-order differentiability of the surrogate.

▹ **Analysis of BREGMANTRON** We are now ready to analyze the BREGMANTRON. Our main result shows that provided the link does not change too much between iterations, we are guaranteed to decrease the following loss:

$$\ell^r_t(S, \boldsymbol{w}_{t'}) \quad \doteq \quad \mathbb{E}_S[D_{U^\star_r}(y \| u_t(\boldsymbol{w}_{t'}^\top \boldsymbol{x}))] \ , \qquad (17)$$

for $r, t, t' = 1, 2, ...$, which gives (13) for $-\underline{L} \doteq U^\star_r$, $u_t \doteq \psi^{-1}$ and $h(\boldsymbol{x}) \doteq \boldsymbol{w}_{t'}^\top \boldsymbol{x}$. We do not impose $r = t$, as our algorithm incorporates a step of proper composite fitting of the next link given the current loss.

We formalise the stability of the link below.

Definition 3 *Let $\alpha_t, \beta_t \geq 0$.* BREGMANTRON *is (α_t, β_t)-stable at iteration t iff the solution $\hat{\boldsymbol{y}}_{t+1}$ in Step 3 satisfies $\hat{y}_1 \in u_t(\boldsymbol{w}_{t+1}^\top \boldsymbol{x}_1) \cdot [1 - \beta_t, 1 + \alpha_t]$.*

Since $\hat{y}_1 \doteq u_{t+1}(\boldsymbol{w}_{t+1}^\top \boldsymbol{x}_1)$ from FIT, it comes that stability requires a bounded local change in inverse link for *a single example* (but implies a bounded change for all via the Lipschitz constraints in Step 3; this is explained in the proof of the main Theorem of this Section). We define the following *mean operators* (Patrini et al., 2014): $\hat{\boldsymbol{\mu}}_y \doteq \mathbb{E}_S[y \cdot \boldsymbol{x}]$ (sample), $\hat{\boldsymbol{\mu}}_t \doteq \mathbb{E}_S[\hat{y}_t \cdot \boldsymbol{x}], \forall t \geq 1$ (estimated), where $\hat{y}_t$ is defined in the BREGMANTRON. We also let $p^*_t \doteq \max\{\mathbb{E}_S[y], \mathbb{E}_S[u_t(\boldsymbol{w}_{t+1}^\top \boldsymbol{x})]\} \ \in [0,1]$ denote the $\max$ estimated $\hat{\Pr}(\mathsf{Y} = 1)$ using both our model and the sample S. Assume $p^*_t > 0$ as otherwise the problem is trivial. Finally, $X \doteq \max_i \|\boldsymbol{x}_i\|_2$ (we consider the L_2 norm for simplicity; our result holds for any norm on $\mathcal{X}$).

Definition 4 BREGMANTRON *is said to be in the δ_t-regime at iteration t, for some $\delta_t > 0$ iff:*

$$\|\hat{\boldsymbol{\mu}}_y - \hat{\boldsymbol{\mu}}_t\|_2 \quad \geq \quad 2\sqrt{p^*_t \delta_t} X, \forall t. \qquad (18)$$

To simplify the statement of our Theorem, we let $f(z) \doteq z/(1+z)$, which satisfies $f(\overline{\mathbb{R}}_+) = [0,1]$.

Theorem 5 *Suppose that* BREGMANTRON *is in the δ_t-regime at iteration t, and the following holds:*

- *in Step 1, the learning rate*

$$\eta_t \quad = \quad \frac{1-\gamma_t}{2N_t X^2} \cdot \left(1 - \frac{f(\delta_t)(1+f(\delta_t))p^*_t X}{\|\hat{\boldsymbol{\mu}}_y - \hat{\boldsymbol{\mu}}_t\|_2}\right),$$

  for some user-fixed $\gamma_t \in \left[0, \sqrt{f \circ f(\delta_t)/2}\right]$;

- *in Step 3, N_t, n_t satisfy ${}^{N_t}/_{n_t}, {}^{N_{t-1}}/_{n_t} \leq 1 + f(\delta_t)$.*

Then if the BREGMANTRON *is $(f(\delta_t), f(\delta_t))$-stable, then:*

$$\ell^{t+1}_{t+1}(S, \boldsymbol{w}_{t+1}) \quad \leq \quad \ell^t_t(S, \boldsymbol{w}_t) - \frac{p^*_t f(\delta_t)}{n_t}. \qquad (19)$$

(proof in SI, Section IV) Explicitly, it can be shown that the learning rate at iteration t lies in the following interval:

$$\eta \quad \in \quad \frac{1-\gamma_t}{2N_t X^2} \cdot \left(1 - \frac{\sqrt{\delta_t p^*_t}(2+\delta_t)}{2(1+\delta_t)^2} \cdot \left[\sqrt{\delta_t p^*_t}, 1\right]\right),$$

Theorem 5 essentially says that as long as $\hat{\boldsymbol{\mu}}_y \neq \hat{\boldsymbol{\mu}}_t$, we can hope to get better results. This is no surprise: the gradient step in Step 1 of BREGMANTRON is proportional to $\hat{\boldsymbol{\mu}}_y - \hat{\boldsymbol{\mu}}_t$.

The conditions on Steps 1 and 3 are easily enforceable at any step of the algorithm, so the Theorem essentially says that whenever the link does not change too much between iterations, we are guaranteed a decrease in the loss and therefore a better fit of the class probabilities. Stability is the only assumption made: unlike Kakade et al. (2011), no assumptions are made about Bayes rule or the distribution $\mathcal{D}$, and no constraints are put on the classifier $\boldsymbol{w}$.

We can also choose to *enforce* stability in the update of u in Step 3. Interestingly, while this restricts the choice of links (at least when $\|\hat{\boldsymbol{\mu}}_y - \hat{\boldsymbol{\mu}}_t\|_2$ is small), this *guarantees* the bound in (19) at no additional cost or assumption.

Corollary 6 *Suppose* BREGMANTRON *is run with so that in Step 3, constraint $\hat{y}_1 \geq 0$ is replaced by*

$$\hat{y}_1 \quad \in \quad u_t(\boldsymbol{w}_{t+1}^\top \boldsymbol{x}_1) \cdot [1 - \beta_t, 1 + \alpha_t], \qquad (20)$$

and all other constraints are kept the same. Suppose BREGMANTRON *is in the δ_t-regime at iteration t, parameters η_t, N_t, n_t are fixed as in Theorem 5 and furthermore $\alpha_t, \beta_t \in [0, f(\delta_t)]$. Then (19) holds.*

(proof in SI, Section V) There exists an alternative reading of Corollary 6: there exists a way to fix the key parameters $(\eta_t, n_t, N_t, \alpha_t, \beta_t)$ at each iteration such that a decrease of the loss is guaranteed *if* our current estimate of the sample mean operator, $\hat{\boldsymbol{\mu}}_t$, is not good enough.

5. Discussion

Bregman divergences have had a rich history outside of convex optimisation, where they were introduced (Bregman, 1967). They are the canonical distortions on the manifold of parameters of exponential families in information geometry (Amari & Nagaoka, 2000), they have been introduced in normative economics in several contexts (Magdalou & Nock, 2011; Shorrocks, 1980). In machine learning, their rediscovery was grounded in their representation and algorithmic properties, starting with the work of M. Warmuth and

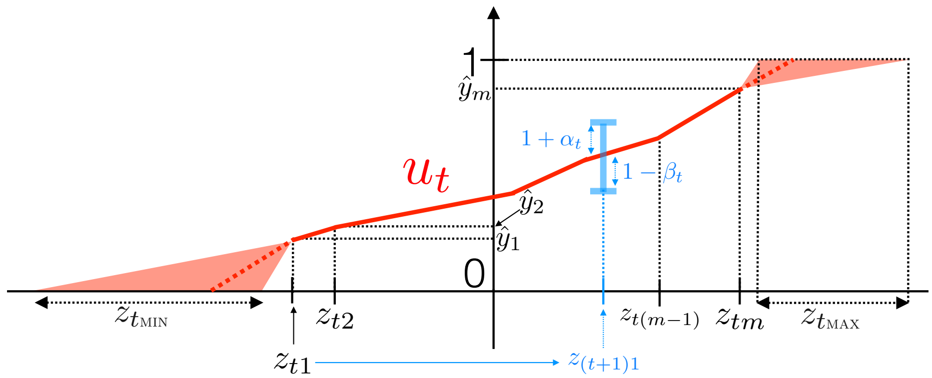

Figure 3: (Inverse) link u of a proper canonical loss learned (red), as computable in FIT ($z_{ti} \doteq \boldsymbol{w}_t^\top \boldsymbol{x}_i$). In blue, we have depicted the stability constraint in the update of the link (Definition 3). Stability imposes just a constraint in the change of (inverse) link for a single example. It does not impose any constraint on the classifier update (which, in this example, is significant for example $(\boldsymbol{x}_1, y_1)$, see text).

collaborators (Helmbold et al., 1995; Herbster & Warmuth, 1998), later linked back to exponential families (Azoury & Warmuth, 2001), and then axiomatized in unsupervised learning (Banerjee et al., 2004; 2005), and then in supervised learning (See Section 4).

Classical uniform convergence bounds apply to the convex surrogate of any proper canonical loss (1-Lipschitz), so we refer to Bartlett & Mendelson (2002, Section 2) for available tools. The setting of the BREGMANTRON raises two questions, the first of which is crucial for the algorithm. We make no assumption about the optimal link, which resorts to a powerful *agnostic* view of machine learning chased in a number of work (Bousquet et al., 2019), but it makes much more sense if we can prove that the link fit by FIT belongs to a set with reasonable approximations of the target. This set contains piecewise affine links, which is a bit more general than Definition 2 but matches the links learned by the BREGMANTRON. We remove the index notation T in u_T and $z_{T.}$, and consider the following ℓ_1 restricted discrepancy,

$$E(u, \ell) \quad \doteq \quad \int_{z_1}^{z_m} |(-\underline{L}')^{-1}(z) - u(z)| \mathrm{d}z, \qquad (21)$$

where ℓ is proper canonical with invertible canonical link. It is restricted because we do not consider set $(-\infty, z_1) \cup (z_m, +\infty)$, whose fitting in fact does not depend on data (see Figure 3). Denote $\mathcal{U}_{n,N}(\boldsymbol{w}, S)$ the set of piecewise affine links with non-$\{0,1\}$ breakout points on abscissae $z_1, z_2, ..., z_m$ ($\boldsymbol{w}^\top \boldsymbol{x}_i \doteq z_i < z_{i+1} \doteq \boldsymbol{w}^\top \boldsymbol{x}_{i+1}, \forall i \in \{0, 1, ..., m-1\}$, wlog), satisfying Definition 2. Let $\|.\|$ be any norm on $\mathcal{X}$ and $\|.\|_*$ its dual. For any $\varepsilon > 0$, $G_\varepsilon(S)$ is the graph whose vertices are the observations in S and an edge links $\boldsymbol{x}, \boldsymbol{x}' \in S$ iff $\|\boldsymbol{x} - \boldsymbol{x}'\| \leq \varepsilon$. G is said 2-connected iff it is connected when any single vertex is removed.

Lemma 7 *For any S of size m, any $\varepsilon > 0$ such that $G_\varepsilon(S)$ is 2-connected and any proper canonical loss ℓ, $\exists n, N \ll \infty$ such that* $\inf_{u \in \mathcal{U}_{n,N}(\boldsymbol{w},S)} E(u, \ell) \leq 2Nm\varepsilon^2 \cdot \|\boldsymbol{w}\|_*^2$.

2-connectivity is a lightweight connectivity condition that essentially prevents the graph from being constituted of two almost separate subgraphs. The proof of the Lemma relies on an old result from graph theory connecting 2-connectivity and Hamiltonicity, known as Fleischner's Theorem (Fleischner, 1974), Gross & Yellen (2004, p. 265, F17). Crucially, N can be much smaller than the Lipschitz constant of $(-\underline{L}')^{-1}$. Lemma 7 does guarantee that the set of links in which the BREGMANTRON finds u_T is powerful enough to approximate a link provided we sample enough examples to drag ε small enough while guaranteeing $G_\varepsilon(S)$ 2-connected. This does not require i.i.d. sampling but would require additional assumptions about $\mathcal{X}$ to be tractable (such as boundedness), or the possibility of active learning in $\mathcal{X}$. This also does not guarantee that FIT finds a link with small $E(., \ell)$, and this brings us to our second question: is it possible that (near-)optimal solutions contain very "different" couples $(u, \boldsymbol{w})$, for which useful notion(s) of "different" ? This, we believe, has ties with the transferability of the loss.

Last, deep learning has achieved tremendous success on how one can learn a mapping φ from a general space $\mathcal{X}$ to $\mathbb{R}^d$, where $\mathcal{X}$ possesses only weak mathematically amenable properties. Our work can be directly branched after (or during) training φ to get a complete proper learning pipeline for $\Pr(\mathsf{Y} = 1 \mid \boldsymbol{x}) \doteq u(\boldsymbol{w}^\top \varphi(\boldsymbol{x}))$. To fold BREGMANTRON within the training of φ shall however likely require significant algorithmic improvements of Lipschitz isotonic regression for an efficient training pipeline.

6. Experimental results

We present experiments illustrating:

(a) the viability of the BREGMANTRON as an alternative to classic GLM or SLISOTRON learning.

(b) the nature of the loss functions learned by the BREGMANTRON, which are potentially *asymmetric*.

(c) the potential of using the loss function learned by the BREGMANTRON as input to some downstream learner.

▷ **Predictive performance of BREGMANTRON** We compare BREGMANTRON as a generic binary classification method against the following baselines: logistic regression, GLMTron Kakade et al. (2011) with $u(\cdot)$ the sigmoid, and SLISOTRON. We also consider two variants of BREGMANTRON: one where in Step 4 we do not find the global optimum (BREGMANTRON$_{\text{approx}}$), but rather a feasible solution with minimal $\hat{y}_m$; and another where in Step 4 we fit against the labels, rather than $\hat{y}_t$ (BREGMANTRON$_{\text{label}}$).

In all experiments, we fix the following parameters for BREGMANTRON: we use a constant learning rate of $\eta = 1$ to perform the gradient update in Step 1, For Step 3, we fix $n_t = 10^{-2}$ and $N_t = 1$ for all iterations.

We compare performance on two standard benchmark datasets, the MNIST digits (mnist) and the fashion MNIST (fmnist) – on this latter domain, we sub-sample the data to 1000 points for computational efficiency. We converted the former to a binary classification problem of the digits 0 versus 8, and the latter of the odd versus even classes. We also consider a synthetic dataset (synth), comprising 2D Gaussian class-conditionals with means $\pm(1, 1)$ and identity covariance matrix. The Bayes-optimal solution for $\Pr(\mathsf{Y} = 1 \mid \mathsf{X})$ can be derived in this case: it takes the form of a sigmoid, as assumed by logistic regression, composed with a linear model proportional to the expectation. In this case therefore, logistic regression works on a search space much smaller than BREGMANTRON and guaranteed to contain the optimum.

On a given dataset, we measure the predictive performance for each method via the area under the ROC curve. This assesses the ranking quality of predictions, which provides a commensurate means of comparison; in particular the BREGMANTRON optimises for a bespoke loss function that can be vastly different from the square-loss.

Table 1 summarises the results. We make three observations. First, BREGMANTRON is consistently competitive with the mature baseline of logistic regression. Interestingly, this is even so on the synth problem, wherein logistic regression is correctly specified. Although the difference in performance here is minor, it does illustrate that BREGMANTRON can infer a meaningful pair of $(u, \boldsymbol{w})$.

	synth	mnist	fmnist
Logistic regression	92.2%	99.9%	98.5%
GLMTron	92.2%	99.6%	98.1%
SLISOTRON	91.6%	94.6%	90.7%
BREGMANTRON$_{\text{approx}}$	92.2%	99.3%	94.6%
BREGMANTRON$_{\text{label}}$	90.1%	99.6%	97.7%
BREGMANTRON	92.3%	99.7%	97.9%

Table 1: Test set AUC of various methods on binary classification datasets. See text for details.

Second, BREGMANTRON and BREGMANTRON$_{\text{label}}$ are generally *superior* to the performance of the SLISOTRON. We attribute this to the latter's reliance on an isotonic regression step to fit the links, as opposed to a Bregman regularisation.

Third, while BREGMANTRON$_{\text{approx}}$ also performs reasonably, it is typically worse than the full BREGMANTRON. This illustrates the value of (at least approximately) solving Step 4 in the BREGMANTRON. Further, while BREGMANTRON$_{\text{label}}$ generally performs slightly worse than standard BREGMANTRON, it remains competitive. A formal analysis of this method would be of interest in future work.

▷ **Illustration of learned losses** As with the SLISOTRON, a salient feature of BREGMANTRON is the ability to automatically learn a link function. Unlike the SLISOTRON, however, the link in the BREGMANTRON has an interpretation of corresponding to a canonical loss function.

Figure 4 illustrates the link functions learned by BREGMANTRON on each dataset. We see that these links are generally *asymmetric* about $\frac{1}{2}$. This is in contrast to standard link functions such as the sigmoid. Recall that each link corresponds to an underlying canonical loss, given by $\ell'(y, v) = u(v) - y$. Asymmetry of $u(\cdot)$ thus manifests in $\ell(+1, v) \neq \ell(-1, -v)$. We illustrate these implicit canonical losses in Figure 5. As a consequence of the links not being symmetric around $\frac{1}{2}$, the losses on the positive and negative classes are not symmetric for the synth dataset. This is unlike the *theoretical* link, but the theoretical link may not be optimal at all on *sampled data*. This, we believe, also illustrates the intriguing potential of the BREGMANTRON to detect and exploit hidden asymmetries in the underlying data distribution.

▷ **Transferability of the loss between domains** Finally, we illustrate the potential of "recycling" the loss function implicitly learned by BREGMANTRON for some other task. We take the fmnist dataset, and first train BREGMANTRON to classify the classes 0 versus 6 ("T-shirt" versus "Shirt"). This classifier achieves an AUC of 0.85, which is competitive with the logistic regression AUC of 0.86.

Recall that BREGMANTRON gives us a learned link u, which per the above discussion also defines an implicit canonical

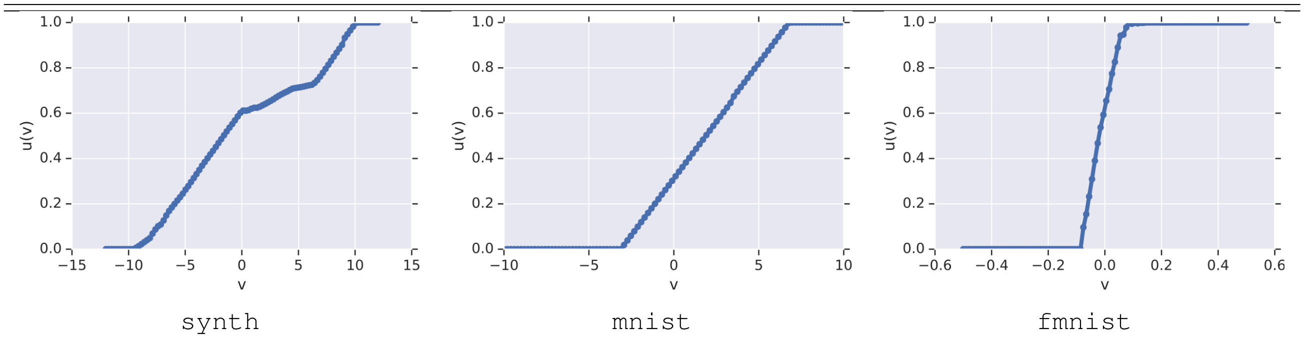

Figure 4: (Inverse) link functions estimated by BREGMANTRON on each dataset. On all datasets, the losses are seen to be (slightly) asymmetric around $\frac{1}{2}$, i.e., $u(v) \neq 1 - u(-v)$. In particular, $u(0) \neq \frac{1}{2}$.

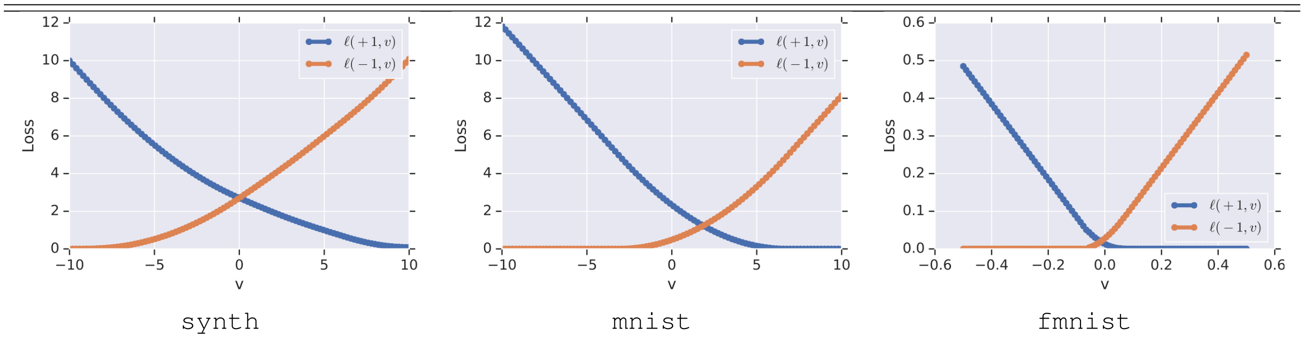

Figure 5: Loss functions estimated by BREGMANTRON on each dataset. The losses are (slightly) asymmetric, and have the flavour of the square-hinge loss; this is a consequence of the linear interpolation used when fitting.

loss. By training a classifier to distinguish classes 2 versus 4 (“Pullover” versus “Coat”) using this loss function, we achieve an AUC of $\mathbf{0.879}$. This slightly outperforms the $\mathbf{0.877}$ AUC of logistic regression, and is also competitive with the $\mathbf{0.879}$ AUC attained by training BREGMANTRON directly on classes 2 versus 4. This indicates that the loss learned by BREGMANTRON on one domain could be useful in related domains to *another* classification algorithms just training a classifier. To properly develop this possibility is out of the scope of this paper, and as far as we know such a perspective is new in machine learning.

7. Conclusion

Fitting a loss that complies with Bayes decision theory implies not just to be able to learn a classifier, but also a canonical link of a proper loss, and therefore a proper canonical loss. In a 2011 seminal work, Kakade *et al.* made with the SLISOTRON algorithm the first attempt at solving this bigger picture of supervised learning. We propose in this paper a more general approach grounded on a general Bregman formulation of differentiable proper canonical losses. From a formal standpoint, an interesting avenue for future work is the inclusion of a regulariser in the loss: we conjecture that the choice of a gradient step in Step 1 of BREGMANTRON makes it convenient to devise and analyse such extensions.

Experiments tend to confirm the ability of our approach, the BREGMANTRON, to significantly beat the SLISOTRON, and compete with classical supervised approaches even when they are informed with the optimal choice of link. Interestingly, they seem to illustrate the importance of a stability requirement made by our theory. More interesting is perhaps the observation that the loss learned by the BREGMANTRON on one domain can be useful to other learning algorithms to fit classifiers on related domains, a *transferability* property of the loss learned that deserves further thought.

Acknowledgments

The authors warmly thank Manfred Warmuth for discussions around the introduction of Bregman divergences in machine learning. Thanks are also due to the reviewers and Lalitha Sankar for numerous remarks and discussions.

References

Amari, S.-I. and Nagaoka, H. *Methods of Information Geometry*. Oxford University Press, 2000.

Auer, P., Herbster, M., and Warmuth, M. Exponentially many local minima for single neurons. In *NIPS*8*, pp. 316–322, 1995.

Azoury, K. S. and Warmuth, M. K. Relative loss bounds for on-line density estimation with the exponential family of distributions. *MLJ*, 43(3):211–246, 2001.

Banerjee, A., Merugu, S., Dhillon, I., and Ghosh, J. Clustering with bregman divergences. In *Proc. of the* 4^{th} *SIAM International Conference on Data Mining*, pp. 234–245, 2004.

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

Bartlett, P., Jordan, M., and McAuliffe, J. D. Convexity, classification, and risk bounds. *J. of the Am. Stat. Assoc.*, 101:138–156, 2006.

Bartlett, P.-L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, 2002.

Bousquet, O., Kane, D., and Moran, S. The optimal approximation factor in density estimation. In *COLT'19*, pp. 318–341, 2019.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

Bregman, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comp. Math. and Math. Phys.*, 7:200–217, 1967.

Buja, A., Stuetzle, W., and Shen, Y. Loss functions for binary class probability estimation ans classification: structure and applications, 2005. Technical Report, University of Pennsylvania.

Cranko, Z., Menon, A.-K., Nock, R., Ong, C. S., Shi, Z., and Walder, C.-J. Monge blunts Bayes: Hardness results for adversarial training. In *36*th *ICML*, pp. 1406–1415, 2019.

de Finetti, B. Rôle et domaine d'application du théorème de Bayes selon les différents points de vue sur les probabilités (*in French*). In *18*th *International Congress on the Philosophy of Sciences*, pp. 67–82, 1949.

Fleischner, H. The square of every two-connected graph is Hamiltonian. *Journal of Combinatorial Theory, Series B*, 16:29–34, 1974.

Grabocka, J., Scholz, R., and Schmidt-Thieme, L. Learning surrogate losses. *CoRR*, abs/1905.10108, 2019.

Gross, J.-L. and Yellen, J. *Handbook of graph theory*. CRC press, 2004. ISBN 1-58488-090-2.

Helmbold, D.-P., Kivinen, J., and Warmuth, M.-K. Worst-case loss bounds for single neurons. In *NIPS*8*, pp. 309–315, 1995.

Herbster, M. and Warmuth, M. Tracking the best regressor. In *9* th *COLT*, pp. 24–31, 1998.

Kakade, S., Kalai, A.-T., Kanade, V., and Shamir, O. Efficient learning of generalized linear and single index models with isotonic regression. In *NIPS*24*, pp. 927–935, 2011.

Kearns, M. J. and Vazirani, U. V. *An Introduction to Computational Learning Theory*. M.I.T. Press, 1994.

Liu, L., Wang, M., and Deng, J. UniLoss: Unified surrogate loss by adaptive interpolation. https://openreview.net/forum?id=ryegXAVKDB, 2019.

Magdalou, B. and Nock, R. Income distributions and decomposable divergence measures. *Journal of Economic Theory*, 146(6):2440–2454, 2011.

Mei, J. and Moura, J.-M.-F. SILVar: Single index latent variable models. *IEEE Trans. Signal Processing*, 66(11): 2790–2803, 2018.

Nock, R. and Nielsen, F. On the efficient minimization of classification-calibrated surrogates. In *NIPS*21*, pp. 1201–1208, 2008.

Nock, R. and Nielsen, F. Bregman divergences and surrogates for learning. *IEEE Trans.PAMI*, 31:2048–2059, 2009.

Nock, R. and Williamson, R.-C. Lossless or quantized boosting with integer arithmetic. In *36*th *ICML*, pp. 4829–4838, 2019.

Patrini, G., Nock, R., Rivera, P., and Caetano, T. (Almost) no label no cry. In *NIPS*27*, 2014.

Reid, M.-D. and Williamson, R.-C. Composite binary losses. *JMLR*, 11:2387–2422, 2010.

Savage, L.-J. Elicitation of personal probabilities and expectations. *J. of the Am. Stat. Assoc.*, 66:783–801, 1971.

Shorrocks, A.-F. The class of additively decomposable inequality measures. *Econometrica*, 48:613–625, 1980.

Siahkamari, A., Saligrama, V., Castanon, D., and Kulis, B. Learning Bregman divergences. *CoRR*, abs/1905.11545, 2019.

Streeter, M. Learning effective loss functions efficiently. *CoRR*, abs/1907.00103, 2019.

Sypherd, T., Diaz, M., Laddha, H., Sankar, L., Kairouz, P., and Dasarathy, G. A tunable loss function for classification. *CoRR*, abs/1906.02314, 2019.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

Zhang, T. Statistical behaviour and consistency of classification methods based on convex risk minimization. *Annals of Mathematical Statistics*, 32:56—-134, 2004.