Boosting as Entropy Projection

Jyrki Kivinen[*]
Department of Computer Science
P.O. Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki
Finland
Jyrki.Kivinen@cs.Helsinki.FI

Manfred K. Warmuth[†]
Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064
U.S.A.
manfred@cse.ucsc.edu

Abstract

We consider the AdaBoost procedure for boosting weak learners. In AdaBoost, a key step is choosing a new distribution on the training examples based on the old distribution and the mistakes made by the present weak hypothesis. We show how AdaBoost's choice of the new distribution can be seen as an approximate solution to the following problem: Find a new distribution that is closest to the old distribution subject to the constraint that the new distribution is orthogonal to the vector of mistakes of the current weak hypothesis. The distance (or divergence) between distributions is measured by the relative entropy. Alternatively, we could say that AdaBoost approximately projects the distribution vector onto a hyperplane de£ned by the mistake vector. We show that this new view of AdaBoost as an entropy projection is dual to the usual view of AdaBoost as minimizing the normalization factors of the updated distributions.

1 Introduction

Boosting, originally suggested by Schapire [Sch90], is a particular method for improving the performance of a (supervised) learning algorithm by applying it several times on slightly modi£ed training data and then combining the results in a suitable manner. Currently the most popular variants of boosting are based on Freund and Schapire's AdaBoost [FS97b]. The details of the boosting framework of our paper are mainly taken from Schapire and Singer's work on con£dence-rated boosting [SS98].

Let us review the basic idea of boosting on a very rough level. We take as our starting point an arbitrary learning algorithm, which in this context is called the *weak learner* (as opposed to the *master algorithm* that implements the whole boosting procedure). We also have a £xed *training set* of examples. Following Freund [Fre95], we choose some probability distribution over the training set as the initial *training distribution*. We then repeat the following until some termination condition is met. We call the weak learner and draw its training examples from the set of all training examples according to the current training distribution. The weak learner produces a *weak hypothesis*. We use the weak hypothesis to update the old training distribution into a new one. The details of this update will be discussed shortly. We then set the weak hypothesis aside and go to the next iteration with the new training distribution. After the termination condition for the iterations is met, the master algorithm outputs as its hypothesis a suitable weighted combination of all the weak hypotheses produced during this process.

The basic question we consider in this paper is how to update the training distribution between the calls to the weak learner. Assuming we have m examples in the training set, we represent the training distribution for the tth call to the weak learner as a distribution vector $\boldsymbol{d}_t \in [0,1]^m$ such that $\sum_i d_{t,i} = 1$. Lacking a reason to do otherwise we would typically choose the initial distribution to be uniform, with $d_{1,i} = 1/m$ for all i. Based on the training distribution $\boldsymbol{d}_t$, the weak learner produces a weak hypothesis h_t. We describe the performance of the weak hypothesis h_t on the training set by the vector $\boldsymbol{u}_t \in [-1,1]^m$, where $u_{t,i}$ indicated the goodness of the algorithm on example number i. In the most basic case we would choose $u_{t,i} = 1$ if h_t predicts correctly on the ith example and $u_{t,i} = -1$ otherwise, but more £ne-grained measures are also possible. The fundamental idea of boosting is now to concentrate the new training distribution on those examples on which the current weak hypothesis performs badly. In particular, in AdaBoost [FS97b] the updated distribution has the following *exponential form*

$$d_{t+1,i} = \frac{1}{Z_t} d_{t,i} \exp(-\alpha_t u_{t,i}), \tag{1.1}$$

where $\alpha_t > 0$ regulates the amount of change and Z_t is a normalization factor that gives $\sum_i d_{t+1,i} = 1$. The £nal hypothesis H of the master algorithm is given by $H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$. Thus α_t also acts as the weight given to h_t. Schapire and Singer [SS98] show in an elegant proof that the training error of the £nal hypothesis after T boosting iterations is bounded by the prod-

[*]Supported by ESPRIT Working Group NeuroCOLT2
[†]Supported by NSF grant CCR 9700201

uct $\prod_{t=1}^{T} Z_t$ of the normalization factors. They show that the choice of α_t in the original AdaBoost [FS97b], namely $\alpha_t = \ln((1 + \boldsymbol{d}_t \cdot \boldsymbol{u}_t)/(1 - \boldsymbol{d}_t \cdot \boldsymbol{u}_t))/2$, minimizes Z_t in the discrete case (when $u_{t,i} \in \{-1, 1\}$). In the continuous-valued case (when $u_{t,i} \in [-1, 1]$) the same choice of α_t only minimizes a certain upper bound for Z_t. As an alternative, they suggest choosing α_t in the continuous valued case so that Z_t is exactly minimized. Further, they show that minimizing Z_t exactly occurs at a unique value α_t such that the dot product $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t = \sum_i d_{t+1,i} u_{t,i}$ is zero. We call the update (1.1), with α_t such that $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t = 0$, the *corrective update*.

Our central idea is to view the corrective update as a solution to a relative entropy minimization problem and the choice of α_t used by AdaBoost as an approximate solution to the same problem. De£ne the relative entropy between distribution vectors $\boldsymbol{d}$ and $\widetilde{\boldsymbol{d}}$ by

$$\Delta(\boldsymbol{d}, \widetilde{\boldsymbol{d}}) = \sum_{i=1}^{m} d_i \ln \frac{d_i}{\widetilde{d}_i} \quad . \tag{1.2}$$

and consider the minimization problem

$$\min_{\boldsymbol{d} \in P_m} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) \qquad \text{subject to } \boldsymbol{d} \cdot \boldsymbol{u}_t = 0 \quad . \tag{1.3}$$

Here $P_m = \{\, \boldsymbol{d} \in \mathbf{R}^m \mid \sum_{i=1}^{m} d_i = 1, d_i \geq 0 \,\}$ is the set of m-dimensional distribution vectors, or the $m-$dimensional *probability simplex*. We show that the corrective update is the solution to the above constrained minimization problem:

$$\boldsymbol{d}_{t+1} = \operatorname*{argmin}_{\boldsymbol{d} \in P_m} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) \qquad \text{subject to } \boldsymbol{d} \cdot \boldsymbol{u}_t = 0 \quad . \tag{1.4}$$

More speci£cally, employing the standard concepts and tools from constrained convex optimization [Lue84, BSS93, HUL91], we show that the constrained minimization problem (1.3) is the dual of the unconstrained problem of maximizing $-\ln Z_t$ as a function of α_t. The variable α_t of the unconstrained problem is effectively the Lagrange multiplier used to enforce the constraint $\boldsymbol{d} \cdot \boldsymbol{u}_t = 0$ in (1.4). Also the value of the minimization and maximization problems are the same, i.e.,

$$\min_{\substack{\boldsymbol{d} \in P_m \\ \boldsymbol{d} \cdot \boldsymbol{u}_t = 0}} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) = \max_{\alpha \in \mathbf{R}} (-\ln Z_t(\alpha)) \quad . \tag{1.5}$$

The relative entropy is of course a very commonly used tool in statistics and in computational learning theory. We wish here to consider two different aspects of applying the relative entropy in the context of on-line learning. First, we can use the relative entropy to analyse the convergence and other properties of existing on-line algorithms. Second, we can use relative entropy to motivate new algorithms. The purpose of the present paper is to bring out explicitly in the context of boosting the connection between these two aspects by means of the duality property (1.5). Thus, we see more clearly the relationship between boosting as a minimizer for Z_t [SS98] and the analyses of boosting-style algorithms in terms of the relative entropy [FS97a, FS97b].

Of course, since we are basically considering two mathematically equivalent derivations for the single update rule (1.1), most if not all of this is already implicit in earlier work. In particular, a procedure more or less equivalent with the corrective boosting algorithm, but in a context somewhat different from boosting weak learners, was analysed using a duality relation similar to (1.5) by Della Pietra et al. [DDL97]; see Lafferty [Laf99] for connecting this work to boosting as it is understood in computational learning theory.

Considering problems other than boosting, one should notice work on on-line prediction algorithms using experts [FSSW97, KW99] and linear regression [KW97]. In this context, the relative entropy has been used in the same kind of double role as here, both in deriving updates and then proving (worst-case) performance bounds for them.

Outside the context of on-line learning theory, and its worst-case bounds, relative entropy minimization with linear constraints is of course an important method of statistics [KK92, Jum90]. Even more generally, the relative entropy is a special case of a *Bregman divergence* [Bre67, Csi91]. Iterative projection algorithms with respect to arbitrary Bregman divergences in the more general case of inequality constraints have been studied extensively in convex optimization [Bre67, CL81, JB90]. In Appendix B we give some notes on generalizing the boosting update (1.4) and the duality connection (1.5) to arbitrary Bregman divergences (but at this point we are unable to show that such updates actually boost weak learners). Similar generalizations have been done in parallel work by Lafferty [Laf99].

For solving the minimization problem (1.3), one can use standard methods of constrained convex optimization; see [HUL91] for an overview. Here we want to point out two earlier papers that use the actual boosting update (1.1) to solve (ostensibly) a different numerical problem. Littlestone, Long and Warmuth [LLW92] suggest this update for solving iteratively a system of linear equations with a sparse solution. Cesa-Bianchi, Krogh, and Warmuth [CBKW94] developed the same algorithm in the context of £nding a maximum likelihood model from an exponential family. Both papers actually give a more general algorithm that corresponds to (1.4) generalized to allow multiple linear constraints, but the algorithms can naturally be specialized to the one-constraint case (1.4). It turns out that in both cases a single iteration step of the one-constraint algorithm is exactly the same as the update step of the original AdaBoost. In particular, the choice of α_t is the same.

In the exponential form update (1.1), there is no obvious reason why we need to have only a single real parameter α_t to adjust at update t. The update (1.1) can naturally be generalized to

$$d_{t+1,i} = \frac{1}{Z'_t} d_{t,i} \exp\left(-\sum_{q=1}^{t} \alpha_{t,q} u_{q,i} \right) \tag{1.6}$$

where again Z'_t is the normalization factor and now a parameter $\alpha_{t,q}$ is chosen for *each* of the past t weak hypotheses. (The update (1.1) uses only a parameter α_t for the most recent hypothesis.) Again, the product of the normalization factors Z'_t bounds the training error of the £nal hypothesis, and it is natural to choose a parameter vector $\boldsymbol{\alpha}_t$ such that Z'_t is minimized. Analogously with the corrective case, the unconstrained problem of maximizing $-\ln Z'_t$ as a function of $\boldsymbol{\alpha}_t \in \mathbf{R}^t$ is dual to the following constrained problem: Minimize the relative entropy $\Delta(\boldsymbol{d}_{t+1}, \boldsymbol{d}_t)$ subject to $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_q = 0$ for all $q \leq t$. Again the original variables

$\alpha_{t,q}$ become Lagrange multipliers in the dual problem. Note that the update (1.6) may be seen as an extended exponential form. We call this update, when $\boldsymbol{\alpha}_t$ is chosen such that $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_q = 0$ holds for $1 \leq q \leq t$, the *totally corrective update*.

As Schapire and Singer [SS98] observe, the property $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t = 0$ of the corrective update has the intuitive meaning that the new distribution should be uncorrolated with the mistakes made by the current weak hypothesis. Then it seems that the new weak hypothesis, trained on the new distribution, should be more likely to give us information not present in the current weak hypothesis. Given this intuitive motivation, it would seem perhaps even better to have the new distribution uncorrelated with all the previous weak hypotheses, leading us to the totally corrective algorithm.

First consider brie¤y implementing the corrective and totally corrective algorithms. For the corrective algorithm, there is only one parameter α_t that can be determined by a simple line search [SS98]. For the totally corrective algorithm there are situations in which all the t constraints $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_q = 0$ cannot be simultaneously satis£ed. Also, even if a solution exists, £nding it is a t-dimensional numerical problem that seems to be nontrivial. A simple method is to repeatedly cycle over the past hypotheses updating one parameter at a time with an approximated corrective algorithm, such as AdaBoost. We discuss these numerical issues brie¤y in Appendix A. One could of course also use the minimization procedures of [LLW92, CBKW94], or any other general convex optimization algorithm for implementing the totally corrective algorithm. However, the convergence bounds in [LLW92, CBKW94] are given in terms of quantities that do not have a natural interpretation in the boosting context.

Instead of getting too involved with the implementation problems of the totally corrective algorithm, we prefer to ignore them and keep the totally corrective algorithm mostly as a conceptual tool for comparison with the corrective algorithm. Actually, we feel that the totally corrective algorithm may not always be the right approach. This is partly because of the problems just mentioned, but we also expect that the totally corrective update might lead boosting to over£t in certain circumstances. However, Della Pietra et al. [DDL97] have succesfully used a method analogous to totally corrective boosting also in practice.

As an alternative to the corrective update, equivalent with (1.4), we suggest the update based on

$$\boldsymbol{d}_{t+1} = \operatorname*{argmin}_{\boldsymbol{d} \in P_m} \left(\Delta(\boldsymbol{d}, \boldsymbol{d}_t) + \eta L(\boldsymbol{d} \cdot \boldsymbol{u}_t) \right), \tag{1.7}$$

where η is a positive parameter and L is some loss function. We assume $L(z) \geq 0$ with equality holding iff $z = 0$. In the limit of η approaching in£nity, (1.7) reduces to minimizing the relative entropy subject to $L(\boldsymbol{d} \cdot \boldsymbol{u}_t) = 0$ (or equivalently $\boldsymbol{d} \cdot \boldsymbol{u}_t = 0$) (1.4). However, by choosing different values of η we can control the trade-off between the tendency to be corrective and the tendency to be conservative, i.e., not move too much in a single update. Thus η can be considered a learning rate parameter. The Exponentiated Gradient algorithm for on-line linear regression has been derived by Kivinen and Warmuth [KW97] as an approximate solution to (1.7). A particularly intriguing connection is that (1.7) with $\eta = 1$ and a certain entropic loss function L gives exactly the AdaBoost update. It would be very interesting to see what kind of boosting results could be proved for algorithms based on other loss functions L and values of η.

We continue by giving in Section 2 a brief review of the boosting algorithms and the error bound of Schapire and Singer [SS98]. Section 3 shows the details of the minimum relative entropy interpretations of the corrective and totally corrective boosting algorithms. We use the minimum relative entropy interpretation in Section 4 for developing some geometric intuitions for the corrective update. The connection to on-line regression algorithms through (1.7) is pursued further in Section 5. Appendix A considers brie¤y some iterative methods for approximately solving the corrective and totally corrective updates. Appendix B discusses generalizations from relative entropy to other Bregman divergences.

2 The boosting algorithms

We take our framework for boosting from Schapire and Singer's work on con£dence-rated boosting [SS98]. We consider classifying elements of an arbitrary set X into two classes, which we denote by -1 and $+1$. Our training set consists of a set of m examples $(x_i, y_i) \in X \times \{ -1, 1 \}$, for $i = 1, \ldots, m$. The interpretation of this input data is that for each *instance* x_i, the *label* y_i gives the correct classi£cation of x_i according to some unknown target classi£er. We allow our weak hypotheses to be arbitrary *con£dence-rated* classi£ers, i.e., mappings from X to $[-1, 1]$. Such a mapping h can be interpreted as predicting classi£cations for the elements of X, with $\operatorname{sign}(h(x))$ the predicted classi£cation and $|h(x)|$ a con£dence rating. The output of the master hypothesis will still be a strict classi£er, i.e., a mapping from X to $\{ -1, 1 \}$. We assume that our weak learner receives a distribution over the training set as its input. We represent these distributions as vectors $\boldsymbol{d}$ from the simplex P_m.

In boosting, we run the weak learner with T different distributions $\boldsymbol{d}_t$, for some suitable number T of rounds, and then combine the resulting weak hypotheses h_t by a weighted majority vote [Sch90, Fre95]. Figure 1 shows the details of AdaBoost and the corrective boosting algorithm, which differ only in the choice of the parameter α_t regulating the amount of change at update t. Otherwise they both share the exponential form (1.1) of the updated distribution, and also the weighted majority form (2.3) of the master algorithm's hypothesis. As a practical point, a value α_t such that $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t = 0$ can be found by a line search except for the degenerate case in which all the components $u_{t,i}$ have the same sign [SS98]. We give in Appendix A some very simple bounds for this line search.

Schapire and Singer noticed that for both AdaBoost and the corrective boosting algorithm the training error of the master hypothesis can be bounded by the product of the normalization factors as

$$\frac{1}{m} | \{ i \mid H(x_i) \neq y_i \} | \leq \prod_{t=1}^{T} Z_t(\alpha_t) \ . \tag{2.4}$$

Since $Z_t(\alpha)$ is minimized when α is such that $\boldsymbol{d}_{t+1}(\alpha) \cdot \boldsymbol{u}_t = 0$, the corrective update is motivated as a minimizer of this upper bound. For AdaBoost, $Z_t(\alpha)$ is replaced by an upper bound (which is exact in the discrete case $u_{t,i} \in \{ -1, 1 \}$) and then α_t is chosen by minimizing this upper bound.

Input a set of m examples $(x_i, y_i) \in X \times \{-1, 1\}$, $i = 1, \ldots, m$.

Initialize $d_{1,i} = 1/m$ for $i = 1, \ldots, m$.

Repeat for $t = 1, \ldots, T$:

- Call the weak learner with the distribution $\boldsymbol{d}_t$ over the examples (x_i, y_i); let the resulting hypothesis be h_t.
- Choose a parameter $\alpha_t \in \mathbf{R}$ as follows. De£ne $\boldsymbol{u}_t \in [-1, 1]^m$ by $u_{t,i} = y_i h_t(x_i)$. Depending on the algorithm, use the following values.

  AdaBoost: Choose

  $$\alpha_t = \frac{1}{2} \ln \frac{1 + \boldsymbol{d}_t \cdot \boldsymbol{u}_t}{1 - \boldsymbol{d}_t \cdot \boldsymbol{u}_t} \quad .$$

  Corrective: Choose α_t such that

  $$\sum_{i=1}^{m} u_{t,i} d_{t,i} \exp(-\alpha_t u_{t,i}) = 0 \quad .$$

- Update the distribution by $\boldsymbol{d}_{t+1} = \boldsymbol{d}_{t+1}(\alpha_t)$ with

  $$d_{t+1,i}(\alpha) = \frac{1}{Z_t(\alpha)} d_{t,i} \exp(-\alpha u_{t,i}) \qquad (2.1)$$

  and

  $$Z_t(\alpha) = \sum_{i=1}^{m} d_{t,i} \exp(-\alpha u_{t,i}) \quad . \qquad (2.2)$$

Output the master hypothesis H de£ned by

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right) \quad . \qquad (2.3)$$

Figure 1: AdaBoost and the corrective and totally corrective boosting algorith

The requirement $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t = 0$ can also be interpreted as requiring that the new distribution makes the current weak hypothesis totally uncorrelated with the training data. Intuitively, the weak learner is then forced to learn something new for h_{t+1}. Given this motivation, it could seem natural to consider a more general algorithm that enforces the constraint $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_q$ not only for $q = t$ but also for all $q < t$. Satisfying all these constraints naturally requires more than one free variable.

Hence, instead of having at update t just one coef£cient α_t for the tth weak hypothesis, we take t coef£cients $\alpha_{t,q}$, one for each past weak hypothesis h_q, $1 \le q \le t$. Using this t-dimensional parameter vector $\boldsymbol{\alpha}_t$ we now write the update as $\boldsymbol{d}_{t+1} = \boldsymbol{d}_{t+1}(\boldsymbol{\alpha}_t)$ where

$$d_{t+1,i}(\boldsymbol{\alpha}) = \frac{1}{Z_t(\boldsymbol{\alpha})} d_{t,i} \exp\left(-\sum_{q=1}^{t} \alpha_q u_{q,i} \right) \qquad (2.5)$$

and

$$Z_t(\boldsymbol{\alpha}) = \sum_{i=1}^{m} d_{t,i} \exp\left(-\sum_{q=1}^{t} \alpha_q u_{q,i} \right) \quad . \qquad (2.6)$$

We call the algorithm with the extended exponential form (2.5), with $\boldsymbol{\alpha}_t$ chosen such that $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_q$ holds for all $1 \le q \le t$, the *totally corrective* algorithm. In the definition (2.3) of the master hypothesis we use the weights $\alpha_t = \sum_{q=1}^{T} \alpha_{q,t}$.

Schapire and Singer's proof of the error bound (2.4) generalizes easily to give

$$\frac{1}{m} |\{ i \mid H(x_i) \neq y_i \}| \le \prod_{t=1}^{T} Z_t(\boldsymbol{\alpha}_t)$$

for the totally corrective algorithm. Analogously with the corrective algorithm, the totally corrective algorithm chooses at update t the parameter vector $\boldsymbol{\alpha}_t$ such that $Z_t(\boldsymbol{\alpha}_t)$ is minimized. It is also easy to see that assuming the same set of weak hypotheses for the corrective and totally corrective algorithm, the bound $\prod_t Z_t(\boldsymbol{\alpha}_t)$ for the totally corrective algorithm is no larger than the bound $\prod_t Z_t(\alpha_t)$ for the corrective algorithm. (But of course we would not expect to get the same weak hypotheses with different distributions.)

It should be noted that there may not exist any $\boldsymbol{\alpha}_t$ such that $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_q$ holds for all $1 \le q \le t$, and in any case £nding such a vector $\boldsymbol{\alpha}_t$ would be an t-dimensional optimization problem. We consider this issue brie¤y in Appendix A. However, we are not claiming that the totally corrective algorithm would necessarily be a practical learning algorithm. We introduce it here mainly as a theoretical comparison point for the corrective algorithm.

3 Boosting as relative entropy minimization

We saw in Section 2 various boosting algorithms whose updated distribution has exponential form can be seen as minimizing the factors $Z_t(\alpha_t)$ that appear in the bound (2.4). This minimization problem explains the choice of the values α_t. The use of the exponential form (2.1) is essential. In particular, the proof of (2.4) uses the property that $e^{-\alpha y h(x)} \le 1$ if and only if $\operatorname{sign}(h(x)) \neq \operatorname{sign}(y)$. Thus the exponential gives a nice approximation to the discrete loss [SS98]. (See [FHT98] for more discussion).

We now suggest an alternative view, in which the corrective and totally corrective updates appear as solutions to constrained relative entropy minimization problems. The exponential form of the update is an immediate consequence of using the relative entropy as a measure of divergence between distributions. If the relative entropy was replaced by a different Bregman divergence, then the update would have another form. The values of the parameters α_t are the Lagrange multiplier for enforcing the constraints. For simplicity, we show the details only for the corrective algorithm and then explain brie¤y how the results generalize to the totally corrective algorithm.

We view the corrective update as pursuing two con¤icting goals. First, the updated distribution should be uncorrolated with the mistakes made by the previous weak hypothesis, i.e. $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t = 0$. Otherwise the updated distribution should stay closest to the last distribution so as to retain

changes made in previous updates and also resist overreacting to noise. The distance is measured by the relative entropy $\Delta(\boldsymbol{d}_{t+1}, \boldsymbol{d}_t)$ de£ned in (1.2).

The following theorem is basically an application of standard duality techniques from convex optimization [Lue84]. Intuitively, a constrained minimization problem for $\Delta(\boldsymbol{d}, \boldsymbol{d}_t)$ turns out to be equivalent to an unconstrained maximization problem for $-\ln Z_t(\alpha)$. A similar result, but with more emphasis on the special properties of the relative entropy, is given by Della Pietra et al. [DDL97]. The free variable α of the unconstrained problem becomes a Lagrange multiplier in the constrained problem. Although the theorem is rather basic, we give the proof in complete detail for clarity.

Theorem 1 *De£ne $\boldsymbol{d}_{t+1}(\alpha)$ and $Z_t(\alpha)$ as in (2.1) and (2.2), and assume that $\boldsymbol{d}_{t+1}(\alpha) \cdot \boldsymbol{u}_t = 0$ for some $\alpha \in \mathbf{R}$. Then*

$$\min_{\substack{\boldsymbol{d} \in P_m \\ \boldsymbol{d} \cdot \boldsymbol{u}_t = 0}} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) = \max_{\alpha \in \mathbf{R}} (-\ln Z_t(\alpha)) \quad . \tag{3.1}$$

Further,

$$\operatorname*{argmin}_{\substack{\boldsymbol{d} \in P_m \\ \boldsymbol{d} \cdot \boldsymbol{u}_t = 0}} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) = \boldsymbol{d}_{t+1}(\alpha_t) \tag{3.2}$$

where

$$\alpha_t = \operatorname*{argmax}_{\alpha \in \mathbf{R}} (-\ln Z_t(\alpha)) \quad .$$

Recall that the corrective update is given by

$$\boldsymbol{d}_{t+1} = \boldsymbol{d}_{t+1}(\alpha_t) \quad \text{where } \alpha_t = \operatorname{argmax}_{\alpha \in \mathbf{R}} (-\ln Z_t(\alpha)).$$

Equivalently α_t is such that $\boldsymbol{d}_{t+1}(\alpha_t) \cdot \boldsymbol{u}_t = 0$. Hence, Theorem 1 gives a relative entropy interpretation both for the corrective update and the normalization factor $Z_t(\alpha_t)$ in the error bound (2.4).

Proof of Theorem 1 As discussed earlier, the assumption that $\boldsymbol{d}_{t+1}(\alpha) \cdot \boldsymbol{u}_t = 0$ for some α implies that this α is actually the unique minimum point for $Z_t(\alpha)$. Hence, in particular, the maximum on the right-hand side of (3.1) is attained at $\alpha = \alpha_t$ where $\boldsymbol{d}_{t+1}(\alpha_t) \cdot \boldsymbol{u}_t = 0$.

Consider now the constrained minimization on the left-hand side of (3.1). De£ne the Lagrangian

$$F_t(\boldsymbol{d}, \alpha) = \Delta(\boldsymbol{d}, \boldsymbol{d}_t) + \alpha \boldsymbol{d} \cdot \boldsymbol{u}_t \quad .$$

The key step of the proof of (3.1) is the minimax equation

$$\min_{\boldsymbol{d} \in P_m} \max_{\alpha \in \mathbf{R}} F_t(\boldsymbol{d}, \alpha) = \max_{\alpha \in \mathbf{R}} \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha) \quad . \tag{3.3}$$

Before going into the proof of (3.3), let us see how it gives the main equality (3.1).

First, it is clear that $\max_{\alpha \in \mathbf{R}} F_t(\boldsymbol{d}, \alpha)$ is $\Delta(\boldsymbol{d}, \boldsymbol{d}_t)$ if $\boldsymbol{d} \cdot \boldsymbol{u}_t = 0$ and ∞ otherwise. Therefore, we have

$$\min_{\substack{\boldsymbol{d} \in P_m \\ \boldsymbol{d} \cdot \boldsymbol{u}_t = 0}} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) = \min_{\boldsymbol{d} \in P_m} \max_{\alpha \in \mathbf{R}} F_t(\boldsymbol{d}, \alpha) \quad . \tag{3.4}$$

Now (3.3) gives us

$$\min_{\substack{\boldsymbol{d} \in P_m \\ \boldsymbol{d} \cdot \boldsymbol{u}_t = 0}} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) = \max_{\alpha \in \mathbf{R}} \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha) \quad .$$

To £nish the proof of (3.1), it is now suf£cient to show

$$\min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha) = -\ln Z_t(\alpha) \tag{3.5}$$

for all $\alpha \in \mathbf{R}$. This last claim follows by direct substitution from the more speci£c result that the value $F_t(\boldsymbol{d}, \alpha)$ for £xed α is minimized when $\boldsymbol{d} = \boldsymbol{d}_{t+1}(\alpha)$:

$$\operatorname*{argmin}_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha) = \boldsymbol{d}_{t+1}(\alpha) \quad . \tag{3.6}$$

Equation (3.6) is obtained by a straightforward differentiation with an additional Lagrange coef£cient to enforce $\sum_i d_i = 1$. Since $\Delta(\boldsymbol{w}, \boldsymbol{w}_t)$ is convex in $\boldsymbol{w}$, the zero of the derivatives is the minimum point.

Thus, the proof of (3.3) remains. First, we have

$$\begin{aligned} \max_{\alpha \in \mathbf{R}} \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha) &= \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha_t) \\ &\leq \min_{\boldsymbol{d} \in P_m} \max_{\alpha \in \mathbf{R}} F_t(\boldsymbol{d}, \alpha) \quad , \end{aligned}$$

where we used the fact shown above that the maximum of $\min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha)$ occurs at $\alpha = \alpha_t$. On the other hand, by applying (3.5) and the fact $\boldsymbol{d}_{t+1}(\alpha_t) \cdot \boldsymbol{u}_t = 0$ we get

$$\begin{aligned} \max_{\alpha \in \mathbf{R}} \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha) &= F_t(\boldsymbol{d}_{t+1}(\alpha_t), \alpha_t) \\ &\geq \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha_t) \quad . \end{aligned}$$

From (3.4) we now see that

$$\max_{\alpha \in \mathbf{R}} \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha) \geq \min_{\boldsymbol{d} \in P_m} \max_{\alpha \in \mathbf{R}} F_t(\boldsymbol{d}, \alpha) \quad ,$$

and (3.3) follows.

The proof of (3.3) implies that actually

$$F_t(\boldsymbol{d}_{t+1}(\alpha_t), \alpha_t) = \min_{\boldsymbol{d} \in P_m} F_t(\boldsymbol{d}, \alpha_t) \quad ,$$

so $\Delta(\boldsymbol{d}_{t+1}(\alpha_t), \boldsymbol{d}_t) = \min_{\boldsymbol{d} \in P_m} \Delta(\boldsymbol{d}, \boldsymbol{d}_t)$. Since $\Delta(\boldsymbol{d}, \boldsymbol{d}_t)$ is convex in $\boldsymbol{d}$ and thus has a unique minimum, (3.2) follows. □

Consider now the totally corrective algorithm. The only difference to the corrective one is that now for $\boldsymbol{d}_{t+1}$ we have t constraints instead of just one. Thus, let

$$C = \{\, \boldsymbol{d} \in P_m \mid \boldsymbol{d} \cdot \boldsymbol{u}_q = 0 \text{ for } 1 \leq q \leq t \,\} \quad .$$

An argument similar to the proof of Theorem 1 shows that

$$\min_{\boldsymbol{d} \in C} (\Delta(\boldsymbol{d}_{t+1}, \boldsymbol{d}_t)) = \max_{\boldsymbol{\alpha} \in \mathbf{R}^t} (-\ln Z_t(\boldsymbol{\alpha})) \quad ,$$

and further $\boldsymbol{\alpha}_t = \operatorname{argmax}_{\boldsymbol{\alpha}} (-\ln Z_t(\boldsymbol{\alpha}))$ and

$$\boldsymbol{d}_{t+1}(\boldsymbol{\alpha}_t) = \operatorname*{argmin}_{\boldsymbol{d} \in C} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) \quad .$$

4 Geometric interpretations for boosting

We next show some properties of the corrective update that follow naturally when we interpret (3.2) in a geometric fashion: the new distribution $\boldsymbol{d}_{t+1}$ is obtained by projecting the old distribution $\boldsymbol{d}_t$ onto the hyperplane $U_t = \{\, \boldsymbol{d} \mid \boldsymbol{d} \cdot \boldsymbol{u}_t = 0 \,\}$. Here the projection of the point $\boldsymbol{d}_t$ to the plane U_t is de£ned as the point $\boldsymbol{d}$ on the plane that is closest to the starting point $\boldsymbol{d}_t$. The relative entropy $\Delta(\boldsymbol{d}, \boldsymbol{d}_t)$ is used as our measure of distance. Although geometric

metaphors are very illuminating here, it must be remembered that the relative entropy is not a metric and we have to be careful when we use our intuitions about distance. The ideas sketched here can be applied directly to the totally corrective update by replacing the hyperplane U_t by the intersection of t hyperplanes U_q, $1 \leq q \leq t$.

Assume now that there is at least one $\alpha \in \mathbf{R}$ such that the distribution $\boldsymbol{d}_{t+1}(\alpha)$ in the exponential form (2.1) satis£es $\boldsymbol{d}_{t+1}(\alpha) \cdot \boldsymbol{u}_t = 0$. As discussed earlier, this is a reasonable assumption in the boosting setting, and then actually there is a unique value α_t such that $\boldsymbol{d}_{t+1}(\alpha) \cdot \boldsymbol{u}_t = 0$ holds if and only if $\alpha = \alpha_t$. Now it is easy to show that all distributions $\boldsymbol{d}_{t+1}(\alpha)$ in exponential form (2.1) project to the same point $\boldsymbol{d}_{t+1}$ on the hyperplane U_t, i.e., for any $\alpha \in \mathbf{R}$ we have

$$\operatorname*{argmin}_{\boldsymbol{d} \in P_m \cap U_t} \Delta(\boldsymbol{d}, \boldsymbol{d}_t) = \operatorname*{argmin}_{\boldsymbol{d} \in P_m \cap U_t} \Delta(\boldsymbol{d}, \boldsymbol{d}_{t+1}(\alpha)) = \boldsymbol{d}_{t+1} \ .$$

Thus, for any distribution $\boldsymbol{d}$ on the curve $\{\, \boldsymbol{d}_{t+1}(\alpha) \mid \alpha \in \mathbf{R} \,\}$, the projection of $\boldsymbol{d}$ on the hyperplane U_t is the unique point where the curve intersects the hyperplane.

Another interesting property is analogous to the Pythagorean Theorem. Consider $\boldsymbol{d}_t$ and its projection $\boldsymbol{d}_{t+1}$ onto the hyperplane U_t. Then for any d_* on the plane U_t,

$$\Delta(\boldsymbol{d}_*, \boldsymbol{d}_t) = \Delta(\boldsymbol{d}_*, \boldsymbol{d}_{t+1}) + \Delta(\boldsymbol{d}_{t+1}, \boldsymbol{d}_t) \ . \tag{4.1}$$

It is easy to check this using the properties noted in Section 3. If we replace in (4.1) the relative entropy $\Delta(\boldsymbol{d}, \boldsymbol{d}')$ by the squared Euclidean distance $||\boldsymbol{d} - \boldsymbol{d}'||_2^2$ and $\boldsymbol{d}_{t+1}$ by the usual Euclidean projection of $\boldsymbol{d}_t$ onto U_t, then (4.1) becomes the familiar Pythagorean Theorem. This property of minimum distance projections onto sets de£ned by linear constraints holds even more generally for all Bregman divergences [Bre67]. For applications in on-line learning theory, see [HW98].

5 Boosting in a regression framework

It has been pointed out [ROM98] that on highly noisy training sets, AdaBoost may tend to over£t. Considering this, the strict constraint $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t = 0$ of the corrective algorithm, and the even tighter constraint of the totally corrective algorithm, seems a little uncautious. As a possible means of avoiding this problem, we suggest replacing the constrained minimization problem (1.4) by

$$\boldsymbol{d}_{t+1} = \operatorname*{argmin}_{\boldsymbol{d} \in P_m} \left(\Delta(\boldsymbol{d}, \boldsymbol{d}_t) + \eta_t L(\boldsymbol{d} \cdot \boldsymbol{u}_t) \right), \tag{5.1}$$

where L is some loss function such that $L(z)$ is minimized for $z = 0$ and $\eta_t > 0$ is a parameter controlling the trade-off between minimizing the loss and staying close to the previous distribution. The corrective update is obtained from (5.1) in the limit when η_t approaches in£nity.

This framework for deriving parameter updates was introduced by Kivinen and Warmuth [KW97] in the context of on-line linear regression. The solution to (5.1) satis£es

$$d_{t+1,i} = \frac{1}{N_t} d_{t,i} \exp\left(-\eta_t L'(\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t) u_{t,i}\right) \ , \tag{5.2}$$

where L' is the derivative of L and N_t a normalization factor. For general L, (5.2) cannot be solved in closed form because of how $\boldsymbol{d}_{t+1}$ appears on the right-hand side. Therefore, Kivinen and Warmuth suggest approximating the derivative $L'(\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_t)$ on the right-hand side by its old value $L'(\boldsymbol{d}_t \cdot \boldsymbol{u}_t)$, which is a reasonable approximation at least for small values of η_t. This results in the Exponentiated Gradient update [KW97]

$$d_{t+1,i} = \frac{1}{M_t} d_{t,i} \exp\left(-\eta_t L'(\boldsymbol{d}_t \cdot \boldsymbol{u}_t) u_{t,i}\right) \ , \tag{5.3}$$

where again M_t is a normalization factor. The trade-off parameter η_t can be interpreted as a learning rate. This approach can be generalized by replacing the relative entropy in (5.1) by any Bregman divergence.

It is here particularly interesting to apply (5.1) with $L(z) = L_{\mathrm{ent}}(z, 0)$, where $L_{\mathrm{ent}}(z, \widehat{z})$ for $z, \widehat{z} \in [-1, 1]$ is the is the usual entropic loss

$$L_{\mathrm{ent}}(z, \widehat{z}) = \frac{1 - z}{2} \ln \frac{1 - z}{1 - \widehat{z}} + \frac{1 + z}{2} \ln \frac{1 + z}{1 + \widehat{z}} \ .$$

Then $L'(z) = \frac{1}{2} \ln((1 + z)/(1 - z))$. By comparing (5.3) with (1.1) and recalling the value $\alpha_t = \ln((1 + \boldsymbol{d}_t \cdot \boldsymbol{u}_t)/(1 - \boldsymbol{d}_t \cdot \boldsymbol{u}_t))/2$ used by AdaBoost, we see that AdaBoost can be interpreted as Exponentiated Gradient with the entropic loss function and learning rate $\eta_t = 1$.

6 Conclusions

We have considered the update step of the standard boosting algorithms as a constrained relative entropy minimizer, or alternatively as projection with respect to the relative entropy distance measure. We hope that our simple observations will be useful in designing better boosting algorithms. Many of the basic properties of the relative entropy are shared more generally by all Bregman divergences [Bre67]. It would be interesting to see whether some other divergences might lead to useful boosting procedures. Some updates motivated by different Bregman divergences are brie¤y discussed in Appendix B, but without any results on the training error of the resulting boosting procedure. Perhaps the GeoLev procedure [DH99] could be related to projections with respect to the squared Euclidean distances. Note that the relative entropy is a special divergence in that it is de£ned on the simplex P_m and this is the natural domain for boosting. For other divergence, an additional projection onto P_m would be needed, but this is not necessarily a problem. See [HW98] for examples of using projections onto arbitrary convex sets in a regression setting.

Another interesting subject for further study is noncorrective updates motivated analogously to regression algorithms as in Section 5. Hopefully, they will provide a means for making the boosting algorithms less prone to over£tting.

Acknowledgments

We wish to thank Nigel Duffy, David Helmbold, Mark Herbster, John Lafferty, and Sandra Panizza for inspiring discussions.

A Finding the corrective parameter values

A.1 The corrective algorithm

As Schapire and Singer [SS98] have observed, there is a unique value α_t such that $\boldsymbol{d}_{t+1}(\alpha_t) \cdot \boldsymbol{u}_t = 0$ for the corrective

algorithm, unless all the components $u_{t,i}$ have the same sign. Such value can be found by a simple line search. However, if the absolute values $|u_{t,i}|$ can be arbitrarily small, then the value α_t can be arbitrarily large. We now give a crude estimate of the range we need to search.

Thus, consider an updated distribution $\boldsymbol{d}_{t+1}(\alpha)$ as in (2.1). We are looking for some upper and lower bounds for the value α_t such that $\boldsymbol{d}_{t+1}(\alpha_t) \cdot \boldsymbol{u}_t = 0$. Assume that $\boldsymbol{d}_t \cdot \boldsymbol{u}_t > 0$. (If $\boldsymbol{d}_t \cdot \boldsymbol{u}_t < 0$, replace $\boldsymbol{u}_t$ with $-\boldsymbol{u}_t$. If $\boldsymbol{d}_t \cdot \boldsymbol{u}_t = 0$, then the solution is $\alpha_t = 0$.) We try to £nd a value $\beta > 0$ such that $\boldsymbol{d}_{t+1}(\beta) \cdot \boldsymbol{u}_t < 0$, so $0 < \alpha_t < \beta$. Let k be such that $u_{t,k} = \min\{\, u_{t,i} \mid d_{t,i} \neq 0 \,\}$. De£ne $P = \{\, i \mid u_{t,i} > 0, d_{t,i} \neq 0 \,\}$, and let j be such that $u_{t,j} = \min_{i \in P} u_{t,i}$. We are assuming $P \neq \emptyset$ and $u_{t,i} < 0$. Consider any α such that $\alpha \geq 1/u_{t,j}$. The function f_α given by $f_\alpha(x) = xe^{-\alpha x}$ has $f'_\alpha(x) < 0$ when $x > 1/\alpha$. Hence, in particular, we have $f_\alpha(u_{t,j}) \geq f_\alpha(u_{t,i})$ for all $i \in P$. We can therefore write

$$\begin{aligned}
&Z_t(\alpha)\boldsymbol{d}_{t+1}(\alpha) \cdot \boldsymbol{u}_t \\
&\quad = \quad d_{t,k}u_{t,k}e^{-\alpha u_{t,k}} + \sum_{i \neq k} d_{t,i} f_\alpha(u_{t,i}) \\
&\quad \leq \quad d_{t,k}u_{t,k}e^{-\alpha u_{t,k}} + \sum_{i \in P} d_{t,i} f_\alpha(u_{t,i}) \\
&\quad \leq \quad d_{t,k}u_{t,k}e^{-\alpha u_{t,k}} + f_\alpha(u_{t,j}) \sum_{i \in P} d_{t,i} \\
&\quad = \quad e^{-\alpha u_{t,k}} \cdot \\
&\qquad \left(d_{t,k}u_{t,k} + u_{t,j}e^{-\alpha(u_{t,j} - u_{t,k})} \sum_{i \in P} d_{t,i} \right) \quad ,
\end{aligned}$$

from which we see that $\boldsymbol{d}_{t+1}(\alpha) \cdot \boldsymbol{u}_t \leq 0$ holds for $\alpha \geq \alpha^*$ where

$$\alpha^* = \frac{1}{u_{t,j} - u_{t,k}} \ln \frac{u_{t,j} \sum_{i \in P} d_{t,i}}{-d_{t,k}u_{t,k}} \quad .$$

Hence, we can take $\beta = \max\{\, 1/u_{t,j}, \alpha^* \,\}$.

We now can do a binary search for α_t in the interval $(0, \beta)$. Alternatively we can start a search for a point γ s.t. $\boldsymbol{d}_{t+1}(\gamma) \cdot \boldsymbol{u}_t < 0$ by starting with $\gamma = 1$ and then doubling γ iteratively. This iterative procedure will terminate quickly because γ can never be much larger than β. Once we found a γ with the property we want we can start the binary serach for α_t in a small region.

A.2 The totally corrective algorithm

As we have seen, the problem of £nding the values for the parameters α for the corrective and totally corrective algorithm is a relative entropy minimization problem. Obviously, there are a large number of general optimization algorithms than could be used to solve such problems. We present here an analysis of an iterative algorithm that is speci£cally tailored to the boosting case. The analysis is very close the bound given in [LLW92] on the number of iterations required for £nding an approximate solution to a system of equations.

Of course, we need to assume that there is at least one distribution $\boldsymbol{d}$ that satis£es all the constraints $\boldsymbol{d} \cdot \boldsymbol{u}_t = 0$. This may not be the case. Consider for example $\boldsymbol{u}_1 = (-1/3, 1/2, 0, 0)$, $\boldsymbol{u}_2 = (0, 0, 1/2, -1/3)$, and $\boldsymbol{u}_3 = (0, 1/2, 0, 1/3)$. Then it is easy to see that there are vectors $\boldsymbol{w}$ that satisfy the three constraints $\boldsymbol{w} \cdot \boldsymbol{u}_t = 0$, but any such vector has both positive and negative components and cannot therefore be normalized into a distribution.

Our approach is quite similar to so-called row-action optimization methods [Bre67, CL81]. We wish to £nd a distribution $\boldsymbol{d}_{t+1} \in P_m$ that minimizes the relative entropy $\Delta(\boldsymbol{d}_{t+1}, \boldsymbol{d}_t)$ subject to t constraints $\boldsymbol{d}_{t+1} \cdot \boldsymbol{u}_q = 0$ for $1 \leq q \leq t$. We de£ne a sequence $\widehat{\boldsymbol{d}}_j$, $j = 1, 2, \ldots$, as follows. We start with $\widehat{\boldsymbol{d}}_1 = \boldsymbol{d}_t$. Then at step j pick one constraint that is not satis£ed with the current distribution $\widehat{\boldsymbol{d}}_j$, and de£ne $\widehat{\boldsymbol{d}}_{j+1}$ be the projection of $\widehat{\boldsymbol{d}}_j$ onto the hyperplane de£ned by that constraint. The limit point to which the sequence $\widehat{\boldsymbol{d}}_j$ converges must then satisfy all the constraints. Further, because of our choice of starting point and because of properties of projections such as discussed in Section 4, this limit point also minimized the relative entropy from $\boldsymbol{d}_t$.

As a minor change to the above outline, we do not here do the projections exactly but are satis£ed with the AdaBoost update step that does an approximate projection, as discussed earlier. Also notice that in the totally corrective algorithm the correct distribution $\boldsymbol{d}_t$ has been obtained from the initial distribution $\boldsymbol{d}_1$ by (perhaps approximate) entropy projections with respect to constraints $\boldsymbol{d}_t \cdot \boldsymbol{u}_q = 0$, $q < t$. Therefore, by the properties of projections in Section 4, the constrained relative entropy minimization problem does not really change if we take $\boldsymbol{d}_1$ instead of $\boldsymbol{d}_t$ as the starting point. Thus, we have an iterative procedure that starts with $\widehat{\boldsymbol{d}}_1 = \boldsymbol{d}_1$ and then for $j = 1, 2, \ldots$ repeats the following:

1. Let q_j be such that $|\widehat{\boldsymbol{d}}_j \cdot \boldsymbol{u}_{q_j}|$ is maximized.
2. Let $\widehat{\alpha}_j = \ln((1 + \widehat{\boldsymbol{d}}_j \cdot \boldsymbol{u}_{q_j})/(1 - \widehat{\boldsymbol{d}}_j \cdot \boldsymbol{u}_{q_j}))/2$.
3. De£ne $\widehat{\boldsymbol{d}}_{j+1}$ by

$$\widehat{d}_{j+1,i} = \frac{1}{\widehat{Z}_j} \widehat{d}_{j,i} \exp\left(-\widehat{\alpha}_j u_{q_j,i}\right)$$

where $\widehat{Z}_j$ is the normalization factor.

Then $\widehat{Z}_j \leq (1 - (\widehat{\boldsymbol{d}}_j \cdot \boldsymbol{u}_{q_j})^2)^{1/2}$, as shown by the usual analysis of AdaBoost [FS97b]. Fix a distribution vector $\boldsymbol{d}_* \in P_m$. It follows straight from the de£nitions that

$$\Delta(\boldsymbol{d}_*, \widehat{\boldsymbol{d}}_j) - \Delta(\boldsymbol{d}_*, \widehat{\boldsymbol{d}}_{j+1}) = -\widehat{\alpha}_j \boldsymbol{d}_* \cdot \boldsymbol{u}_{q_j} - \ln \widehat{Z}_j \quad .$$

Fix now some parameter $0 < \gamma < 1$, and let J be the largest index such that $|\widehat{\boldsymbol{d}}_j \cdot \boldsymbol{u}_{q_j}| \geq \gamma$ holds for all $j \leq J$. Then for $j \leq J$ we have

$$\begin{aligned}
-\ln \widehat{Z}_j \quad &\geq \quad -(1/2)\ln(1 - (\widehat{\boldsymbol{d}}_j \cdot \boldsymbol{u}_{q_j})^2) \\
&\geq \quad -(1/2)\ln(1 - \gamma^2) \quad ,
\end{aligned}$$

so in particular $\sum_{j=1}^{J}(-\ln \widehat{Z}_j) \geq -(J/2)\ln(1 - \gamma^2)$. On the other hand, assume now that $\boldsymbol{d}_*$ can be chosen such that $\boldsymbol{d}_* \cdot \boldsymbol{u}_q = 0$ holds for all $1 \leq q \leq t$. (That is, there is a

solution $\boldsymbol{d}_*$ that satis£es the constraints.) Then

$$\begin{aligned}\sum_{j=1}^{J}(-\ln \widehat{Z}_j) &= \sum_{j=1}^{J}\Big(\Delta(\boldsymbol{d}_*,\widehat{\boldsymbol{d}}_j)-\Delta(\boldsymbol{d}_*,\widehat{\boldsymbol{d}}_{j+1})\Big)\\ &= \Delta(\boldsymbol{d}_*,\widehat{\boldsymbol{d}}_1)-\Delta(\boldsymbol{d}_*,\widehat{\boldsymbol{d}}_{J+1})\\ &\le \ln m\end{aligned}$$

if we in addition choose $\boldsymbol{d}_1 = \widehat{\boldsymbol{d}}_1$ to be the uniform start vector. Hence, we have

$$J \le \frac{2\ln m}{-\ln(1-\gamma^2)} \le \frac{2\ln m}{\gamma^2} \ .$$

This is an upper bound for the number of iteration rounds before all the dot products $\widehat{\boldsymbol{d}}_j \cdot \boldsymbol{u}_q$ get smaller than γ in absolute value.

The above iterative method can also be used in the case where there is just one constraint (i.e., the case of the corrective update). Thus iterating AdaBoost on the last weak hypothesis can be used to £nd the corrective update. This is a simple alternate to the binary search method discussed in the previous section.

B Other distance measures

As we have mentioned above, relative entropy is a special case of *Bregman divergences* [Bre67], and much of the discussion about the boosting update and its motivation applies directly to the case of general Bregman divergences. In this section we explain the connection of the minimax results of Theorem 1 to the general duality properties of constrained convex optimization problems. Similar duality properties have been analysed in parallel work by Lafferty [Laf99]. Notice that these ideas only generalize the motivation of boosting as relative entropy minimization. The training error bounds for boosting [FS97b, SS98] are based on the speci£c way the exponential function appears in the update that minimizes relative entropy, and we know of no way of generalizing this to updates minimizing other Bregman divergences.

The discussion here is on a general level, and we omit regularity conditions such as having the optimal solution lie in the interior of the feasible region. To obtain a rigorous proof, such details would need to be considered, but it seems easier to do this individually for each Bregman divergence we wish to consider (like we did in Theorem 1 for the relative entropy) rather than to try to obtain general necessary and suf£cient conditions. For a more complete treatment of duality in convex optimization, see standard textbooks such as Luenberger [Lue84, pp. 396–401] or Bazaraa et al. [BSS93, pp. 199–210].

Consider now a continuously differentiable strictly convex function F from some convex set $X \subseteq \mathbf{R}^m$ to $\mathbf{R}$. Hence, the gradient $\nabla F = f$ is a one-to-one mapping from $\mathbf{R}^m$ to $\mathbf{R}^m$. We de£ne the Bregman divergence Δ_F for vectors $\boldsymbol{w}$ and $\widetilde{\boldsymbol{w}}$ by

$$\Delta_F(\widetilde{\boldsymbol{w}},\boldsymbol{w}) = F(\widetilde{\boldsymbol{w}}) - F(\boldsymbol{w}) - (\widetilde{\boldsymbol{w}}-\boldsymbol{w})\cdot f(\boldsymbol{w}) \ . \quad \text{(B.1)}$$

Thus $\Delta_F(\widetilde{\boldsymbol{w}},\boldsymbol{w})$ is the difference between $F(\widetilde{\boldsymbol{w}})$ and its approximation based on the £rst order Taylor polynomial of F around $\boldsymbol{w}$. Since F is strictly convex, this difference is strictly positive for $\widetilde{\boldsymbol{w}} \neq \boldsymbol{w}$.

Let us now consider a generalized boosting update, which we obtain by replacing the relative entropy by an arbitrary Bregman divergence in (1.4):

$$\boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w}} \Delta_F(\boldsymbol{w},\boldsymbol{w}_t) \quad \text{subject to } \boldsymbol{w}\cdot\boldsymbol{u}_t = 0 \ . \quad \text{(B.2)}$$

Notice that we have omitted the constraint that the weights must be in the probability simplex and, with this in mind, use the symbol $\boldsymbol{w}$ instead of $\boldsymbol{d}$. If one wants to keep the weights as probability vectors, which of course is needed in the standard boosting scenario, one can enforce the constraint $\boldsymbol{w} \in P_m$ by the usual method of Lagrange multipliers. (This involves one multiplier for the constraint $\sum_i d_i = 1$ and m multipliers for the constraints $d_i \ge 0$.) Another possibility would be to £rst obtain a solution $\widetilde{\boldsymbol{w}}$ to the minimization problem ignoring the constraint $\boldsymbol{w} \in P_m$ and then obtain the £nal solution $\boldsymbol{w}_{t+1} \in P_m$ as the projection of $\widetilde{\boldsymbol{w}}$ into P_m with respect to Δ_F. This method was used by Herbster and Warmuth [HW98] in a regression setting. It is also possible to entirely ignore the interpretation of the weights as probabilities and apply the boosting framework simply as a certain kind of a parameter £tting procedure. This is the approach taken by Friedman et al. [FHT98], and also the method suggested by Della Pietra et al. in an earlier work [DDL97] (see Lafferty [Laf99] for later developments). Of course, if the weights do not represent a probability distribution, the weak learner must somehow be able to use the weights directly instead of via sampling. Because of this variety of possibilities, we concentrate here on properties of the update (B.2) on a level that leaves our position with respect to the constraint $\boldsymbol{w} \in P_m$ open.

To solve (B.2), introduce now the Lagrangian by

$$U_t(\boldsymbol{w},\alpha) = \Delta_F(\boldsymbol{w},\boldsymbol{w}_t) + \alpha\boldsymbol{w}\cdot\boldsymbol{u}_t \ . \quad \text{(B.3)}$$

First we wish to solve for a £xed α the value

$$\boldsymbol{w}_{t+1}(\alpha) = \operatorname*{argmin}_{\boldsymbol{w}} U_t(\boldsymbol{w},\alpha) \ . \quad \text{(B.4)}$$

By substituting the de£nition (B.1) into (B.3) and differentiating with respect to $\boldsymbol{w}$ we see that

$$f(\boldsymbol{w}_{t+1}(\alpha)) = f(\boldsymbol{w}_t) - \alpha\boldsymbol{u}_t \ . \quad \text{(B.5)}$$

Notice that since f is one-to-one, this determines $\boldsymbol{w}_{t+1}(\alpha)$ uniquely. Now, in particular, the solution to (B.2) is given by $\boldsymbol{w}_{t+1} = \boldsymbol{w}_{t+1}(\alpha_t)$ where α_t is such that the constraint $\boldsymbol{w}_{t+1}(\alpha)\cdot\boldsymbol{u}_t = 0$ is satis£ed.

As we noticed, f is one-to-one; let g be the inverse of f, so we can write $\boldsymbol{w} = g(\boldsymbol{\theta})$ for any $\boldsymbol{\theta} = f(\boldsymbol{w})$. We can then alternatively give the solution as an additive update in the $\boldsymbol{\theta}$ parameters: we have $\boldsymbol{w}_t = g(\boldsymbol{\theta}_t)$ where $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_{t+1}(\alpha_t)$ for

$$\boldsymbol{\theta}_{t+1}(\alpha) = \boldsymbol{\theta}_t - \alpha\boldsymbol{u}_t \ . \quad \text{(B.6)}$$

To see the connection to the original boosting update, consider $F(\boldsymbol{w}) = \sum_{i=1}^m (w_i \ln w_i - w_i)$, for which the gradient is given by $f_i(\boldsymbol{w}) = \ln w_i$ and the Bregman divergence is the *unnormalized relative entropy*

$$\Delta_F(\boldsymbol{w},\boldsymbol{w}_t) = \sum_{i=1}^{m}\left(w_i \ln\frac{w_i}{w_{t,i}} - w_i + w_{t,i}\right) \ . \quad \text{(B.7)}$$

The update correponding to this divergence satis£es

$$\ln w_{t+1,i} = \ln w_{t,i} - \alpha_t u_{t,i} \ ,$$

which differs from the boosting update

$$\ln w_{t+1,i} = \ln w_{t,i} - \alpha_t u_{t,i} - \ln Z_t$$

only by the normalization. As we mentioned, the normalization constraint $\sum_i w_i = 1$ can be included into the problem (B.2) via an additional Lagrange multiplier in (B.3). The positivity constraints $w_i \geq 0$ turn out to hold automatically for the special case of unnormalized relative entropy.

Consider now the dual problem. For the Lagrangian U_t we de£ne the usual dual function Q_t by

$$Q_t(\alpha) = \min_{\boldsymbol{w}} U_t(\boldsymbol{w}, \alpha) \ . \tag{B.8}$$

Basic duality results (see, e.g., [BSS93, Theorem 6.2.4]) now imply that the constrained problem of minimizing $\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)$ subject to $\boldsymbol{w} \cdot \boldsymbol{u}_t = 0$ is equivalent to the unconstrained problem of maximizing $Q_t(\alpha)$. To be more precise, notice £rst that $\Delta_F(\boldsymbol{w}, \boldsymbol{w}_t)$ is convex in $\boldsymbol{w}$, and $Q_t(\alpha)$ can easily shown to be concave in α, so they have a unique minimum and maximum point, respectively. These are now known to give the same value, i.e.,

$$\min_{\boldsymbol{w}\cdot\boldsymbol{u}_t=0} \Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) = \max_{\alpha} Q_t(\alpha) \ . \tag{B.9}$$

Further, the minimum and maximum points correspond to each other [BSS93, Theorem 6.2.5] in the sense that

$$\operatorname*{argmin}_{\boldsymbol{w}\cdot\boldsymbol{u}_t=0} \Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) = \boldsymbol{w}_{t+1}(\alpha_t)$$

where $\boldsymbol{w}_{t+1}(\alpha)$ is as in (B.4) and

$$\alpha_t = \operatorname*{argmax}_{\alpha} Q_t(\alpha) \ . \tag{B.10}$$

To make more use of this, let us write $Q_t(\alpha)$ out in a more explicit form. For this, it is useful to introduce the *convex conjugate* of F [Roc70]. This is the function G that satis£es

$$G(\boldsymbol{\theta}) + F(\boldsymbol{w}) = \boldsymbol{\theta} \cdot \boldsymbol{w} \tag{B.11}$$

for $\boldsymbol{\theta} = f(\boldsymbol{w})$. From the convexity of F, we know that G is well-de£ned and convex. Further, by differentiating the de£nition (B.11) we see directly the gradient $g = \nabla G$ is the inverse of the gradient of F, i.e., $g(\boldsymbol{\theta}) = \boldsymbol{w}$ when $f(\boldsymbol{w}) = \boldsymbol{\theta}$. By substituting (B.11) into the de£nition (B.1) we see the connection [Ama85, AW99]

$$\Delta_G(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = \Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w})$$

(notice the change in the order of variables). In particular, from (B.6) we get

$$\begin{aligned} Q_t(\alpha) &= \Delta_F(\boldsymbol{w}_{t+1}(\alpha), \boldsymbol{w}_t) + \alpha \boldsymbol{w}_{t+1}(\alpha) \cdot \boldsymbol{u}_t \\ &= \Delta_G(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1}(\alpha)) + \alpha \boldsymbol{w}_{t+1}(\alpha) \cdot \boldsymbol{u}_t \\ &= G(\boldsymbol{\theta}_t) - G(\boldsymbol{\theta}_{t+1}(\alpha)) \\ &\quad - (\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}(\alpha)) \cdot \boldsymbol{w}_{t+1}(\alpha) \\ &\quad + \alpha \boldsymbol{w}_{t+1}(\alpha) \cdot \boldsymbol{u}_t \\ &= G(\boldsymbol{\theta}_t) - G(\boldsymbol{\theta}_t - \alpha \boldsymbol{u}_t) \ . \end{aligned}$$

Hence, (B.9) becomes

$$\min_{\boldsymbol{w}\cdot\boldsymbol{u}_t=0} \Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) = \max_{\alpha} \left(G(\boldsymbol{\theta}_t) - G(\boldsymbol{\theta}_t - \alpha \boldsymbol{u}_t) \right) \ ,$$

and (B.10) gives

$$\alpha_t = \operatorname*{argmax}_{\alpha} \left(-G(\boldsymbol{\theta}_t - \alpha \boldsymbol{u}_t) \right) \ .$$

As a simple example, consider the unnormalized relative entropy (B.7), which is the Bregman divergence for the convex function $F = \sum_{i=1}^m (w_i \ln w_i - w_i)$ de£ned in $X = \mathbf{R}_+^m$. The gradient f now is given by $f_i(\boldsymbol{w}) = \ln w_i$, so for its inverse g we get $g(\boldsymbol{\theta}) = e^{\theta_i}$. Clearly $g = \nabla G$ for $G(\boldsymbol{\theta}) = \sum_{i=1}^m e^{\theta_i}$, and indeed this combination of F and G satis£es the condition (B.11). The update we get from this divergence is then $w_{t+1,i} = w_{t,i} \exp(-\alpha_t u_{t,i})$, and

$$\begin{aligned} &\min_{\boldsymbol{w}\cdot\boldsymbol{u}_t=0} \Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) \\ &\quad = \Delta_F(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) \\ &\quad = \max_{\alpha} \left(G(\boldsymbol{\theta}_t) - G(\boldsymbol{\theta}_t - \alpha \boldsymbol{u}_t) \right) \\ &\quad = \max_{\alpha} \left(\sum_{i=1}^m w_{t,i} (1 - \exp(-\alpha u_{t,i})) \right) . \end{aligned}$$

If we constrain the weights to satisfy $\sum_i w_i = 1$, then of course the unnormalized relative entropy becomes the usual relative entropy, but the above derivation for the value $\Delta_F(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t)$ in terms of G becomes invalid. To see the algorithm for the relative entropy, i.e., the corrective boosting algorithm, in this light, £rst notice that the boosting update (1.1) can be written as $\boldsymbol{w}_t = g(\boldsymbol{\theta}_t)$ with $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_{t+1}(\alpha)$ when g is the *softmax function*

$$g_i(\boldsymbol{\theta}) = \frac{e^{\theta_i}}{\sum_{j=1}^m e^{\theta_j}} \ .$$

Then $g = \nabla G$ for

$$G(\boldsymbol{\theta}) = \ln \left(\sum_{i=1}^m e^{\theta_i} \right) \ .$$

We then get for the relative entropy $\Delta(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t)$ the value $G(\boldsymbol{\theta}_t) - G(\boldsymbol{\theta}_{t+1})$ as expected. Unfortunately, G is not *strictly* convex, and accordingly g is not one-to-one, so the derivation given above is not valid without modi£cations. The simplest way to resolve this is to represent the weights $\boldsymbol{w} \in P_m$ by lower-dimensional weights $\boldsymbol{w}' \in [0,1]^{m-1}$ with $w_i = w'_i$ for $1 \leq i \leq m-1$ and $w_m = 1 - \sum_{i=1}^{m-1} w'_i$. We omit the details of this reduction, but the result is that we also get Theorem 1 for the usual relative entropy as a special case of the derivation given here.

Another related divergence is the sum of binary relative entropies used by Bylander [Byl97] to analyse on-line linear regression. This divergence is de£ned for vectors in $[0,1]^m$, with

$$\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) = \sum_{i=1}^m \left(\widetilde{w}_i \ln \frac{\widetilde{w}_i}{w_i} + (1 - \widetilde{w}_i) \ln \frac{1 - \widetilde{w}_i}{1 - w_i} \right)$$

for $F(\boldsymbol{w}) = \sum_{i=1}^m (w_i \ln w_i + (1-w_i) \ln(1-w_i))$. The gradient is now given by $f_i(\boldsymbol{w}) = \ln(w_i/(1-w_i))$, from which we get $g_i(\boldsymbol{\theta}) = e^{\theta_i}/(1+e^{\theta_i})$. The update then becomes

$$w_{t+1,i} = \frac{w_{t,i} e^{-\alpha_t u_{t,i}}}{1 - w_{t,i} + w_{t,i} e^{-\alpha_t u_{t,i}}} \ .$$

We also get $G(\boldsymbol{\theta}) = \sum_{i=1}^{m} \ln(1 + e^{\theta_i})$, so

$$\begin{aligned}
&\min_{\boldsymbol{w}\cdot\boldsymbol{u}_t=0} \Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) \\
&\quad = \quad \Delta_F(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) \\
&\quad = \quad \max_{\alpha} \left(\sum_{i=1}^{m} (\ln(1 + e^{\theta_{t,i}}) - \ln(1 + e^{\theta_{t,i} - \alpha u_{t,i}}) \right) \\
&\quad = \quad \max_{\alpha} \left(\sum_{i=1}^{m} -\ln(1 - w_{t,i} + w_{t,i} e^{-\alpha u_{t,i}}) \right) \quad .
\end{aligned}$$

Finally, if we take $F(\boldsymbol{w}) = ||\boldsymbol{w}||^2/2 = \sum_{i=1}^{m} w_i^2/2$, we get the squared Euclidean distance $\Delta_F(\widetilde{\boldsymbol{w}}, \boldsymbol{w}) = ||\widetilde{\boldsymbol{w}} - \boldsymbol{w}||^2/2$. The gradient is the identity function, so $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha_t \boldsymbol{u}_t$, and $G = F$, so

$$\begin{aligned}
&\min_{\boldsymbol{w}\cdot\boldsymbol{u}_t=0} \Delta_F(\boldsymbol{w}, \boldsymbol{w}_t) \\
&\quad = \quad \Delta_F(\boldsymbol{w}_{t+1}, \boldsymbol{w}_t) \\
&\quad = \quad \frac{1}{2} \max_{\alpha} \left(||\boldsymbol{w}_t||^2 - ||\boldsymbol{w}_t - \alpha \boldsymbol{u}_t|| \right) \\
&\quad = \quad \frac{(\boldsymbol{w}_t \cdot \boldsymbol{u}_t)^2}{||\boldsymbol{u}||^2} \quad .
\end{aligned}$$

(In this special case we were thus able to solve the maximization in closed form.) Geometrically, $\boldsymbol{w}_{t+1}$ is the point closest to $\boldsymbol{w}_t$ on the hyperplane $\boldsymbol{w} \cdot \boldsymbol{u}_t = 0$.

References

[Ama85] S. Amari. *Differential Geometrical Methods in Statistics*. Springer, Berlin, 1985.

[AW99] K. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. In *Proc. 15th Conf. on Uncertainty in Arti£cial Intelligence*. Morgan Kaufmann, San Francisco, CA, 1999.

[Bre67] L. M. Bregman. The relaxation method of £nding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Physics*, 7:200–217, 1967.

[BSS93] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, New York, NY, 1993. Second edition.

[Byl97] T. Bylander. The binary exponentiated gradient algorithm for learning linear functions. In *Proc. 10th Annu. Conf. on Comput. Learning Theory*, pages 184–192. ACM, New York, 1997.

[CBKW94] N. Cesa-Bianchi, A. Krogh, and M. K. Warmuth. Bounds on approximate steepest descent for likelihood maximization in exponential families. *IEEE Transaction on Information Theory*, 40(4):1215–1220, 1994.

[CL81] Y. Censor and A. Lent. An iterative row-action method for interval convex programming. *Journal of Optimization Theory and Applications*, 34(3):321–353, 1981.

[Csi91] I. Csiszar. Why least squares and maximum entropy? An axiomatic approach for linear inverse problems. *The Annals of Statistics*, 19(4):2032–2066, 1991.

[DDL97] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random £elds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[DH99] N. Duffy and D. P. Helmbold. A geometric approach to leveraging weak learners. In *Computational Learning Theory: 4th European Conference (EuroCOLT '99)*, pages 18–33. Springer, Berlin, 1999.

[FHT98] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Stanford University, 1998.

[Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, September 1995.

[FS97a] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. To appear in *Games and Economic Behavior*.

[FS97b] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[FSSW97] Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. In *Proc. 29th Annual ACM Symposium on Theory of Computing*, pages 334–343. ACM, New York, 1997.

[HUL91] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Springer, Berlin, 1991.

[HW98] M. Herbster and M. K. Warmuth. Tracking the best regressor. In *Proc. 11th Annu. Conf. on Comput. Learning Theory*, pages 24–31. ACM, New York, 1998.

[JB90] L. Jones and C. Byrne. General entropy criteria for inverse problems, with applications to data compression, pattern classi£cation and cluster analysis. *IEEE Transactions on Information Theory*, 36(1):23–30, 1990.

[Jum90] G. Jumarie. *Relative information*. Springer, Berlin, 1990.

[KK92] J. N. Kapur and H. K Kesavan. *Entropy Optimization Principles with Applications*. Academic Press, 1992.

[KW97] J. Kivinen and M. K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Information and Computation*, 132(1):1–64, 1997.

[KW99] J. Kivinen and M. K. Warmuth. Averaging expert predictions. In *Computational Learning Theory: 4th European Conference (EuroCOLT '99)*, pages 153–167. Springer, Berlin, 1999.

[Laf99] J. Lafferty. Additive models, boosting, and inference for generalized divergences. In *Proc. 12th Annu. Conf. on Comput. Learning Theory*.

ACM, New York, 1999.

[LLW92] N. Littlestone, P. M. Long, and M. K. Warmuth. On-line learning of linear functions. Technical Report UCSC-CRL-91-29, University of California, Santa Cruz, 1992. Short version appeared in *Journal of Computational Complexity*, 5:1–23, 1995.

[Lue84] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA, 1984. Second edition.

[Roc70] R. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[ROM98] G. Rätch, T. Onoda, and K. Müller. Soft margins for AdaBoost. Technical Report NC-TR-1998-021, NeuroCOLT2 Technical Report Series, 1998.

[Sch90] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[SS98] R. E. Schapire and Y. Singer. Improved boosting algorithms using con£dence-rated predictions. In *Proc. 11th Annu. Conf. on Comput. Learning Theory*, pages 80–91. ACM, New York, 1998.