On the Convergence of Leveraging[1]

Gunnar Rätsch[2] Sebastian Mika[3]
Manfred K. Warmuth[4]

NeuroCOLT2 Technical Report Series

NC2-TR-2001-098

August 7, 2001

Produced as part of the ESPRIT Working Group
in Neural and Computational Learning II,
NeuroCOLT2 27150

For more information see the NeuroCOLT website
http://www.neurocolt.com
or email neurocolt@neurocolt.com

[1]Supported by DFG grants MU 987/1-1, JA 379/9-1, and NSF grant CCR 9821087; we gratefully acknowledge help from B. Borchers, P. Spellucci, R. Israel and S. Lemm.
[2]raetsch@first.gmd.de GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany
[3]mika@first.gmd.de GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany
[4]manfred@cse.ucsc.edu University of California at Santa Cruz, CA 95060, USA

Abstract

We give an unified convergence analysis of ensemble learning methods including e.g. AdaBoost, Logistic Regression and the Least-Square-Boost algorithm for regression. These methods have in common that they iteratively call a base learning algorithm, which returns hypotheses that are then linearly combined. We show that these methods are related to the *Gauss-Southwell method* known from numerical optimization and show *linear convergence* for all these methods. Our analysis includes ℓ_1 regularized cost functions leading to a clean and general way to regularize ensemble learning.

1 Introduction

We show the non-asymptotical convergence of ensemble learning methods such as AdaBoost [FS97], Logistic Regression (LR) [FHT98, CSS00] and the Least-Square (LS) regression algorithm called LS-Boost [Fri99]. These algorithms have in common that they iteratively call a base learning algorithm L (also called *weak learner*) on a weighted training sample. The base learner is expected to return in each iteration t a hypothesis $\hat{h}_t$ from some hypothesis space $\mathcal{H}$ that has small *weighted training error*.[1] These hypotheses are then linearly combined to form the final or *combined hypothesis* $f_{\hat{\boldsymbol{\alpha}}}(\mathbf{x}) = \sum_t \hat{\alpha}_t \hat{h}_t(\mathbf{x})$. The hypothesis coefficient $\hat{\alpha}_t$ is determined at iteration t, such that a certain objective is minimized or approximately minimized, and is fixed for later iterations. We will work out sufficient conditions on the base learning algorithms to achieve linear convergence, which means that for any starting condition the optimum can be reached with precision $\epsilon > 0$ in only $\mathcal{O}(\log(1/\epsilon))$ iterations.

Relation to Previous Work For AdaBoost and Logistic Regression it has been shown [CSS00] that they generate a combined hypothesis *asymptotically* minimizing a loss functional G only depending on the output of the combined hypothesis $f_{\boldsymbol{\alpha}}$. However, the assumed conditions in [CSS00] on the performance of the base learner are rather strict and can usually not be satisfied in practice. Although the analysis in [CSS00] holds in principle for any strictly convex cost function of Legendre-type (e.g. [Roc70], p. 258, and [BB97]), one needs to show the existence of a so-called *auxiliary function* [DDL97, CSS00] for each cost function other than the exponential or the logistic loss. This can indeed be done [cf. Rät01, Section 4.2], but in any case only leads to asymptotical results.

In an earlier attempt to show the convergence of such methods for arbitrary loss functions [MBBF00], one needed to assume that the hypothesis coefficients $\hat{\alpha}_t$ are upper bounded by a rather small constant. For this case it has been shown that the algorithm asymptotically converges to a combined hypothesis minimizing G. However, since the $\hat{\alpha}_t$'s need to be small, the algorithm requires many iterations to achieve this goal.

In [DH00] it has been shown that for loss functions which are (essentially) exponentially decreasing (including the loss functions of AdaBoost and Logistic

[1] This is the weighted number of false predictions in classification and weighted estimation error in regression

regression), the loss is $\mathcal{O}(1/\sqrt{t})$ in the first $\tilde{t}$ iterations and afterwards $\mathcal{O}(\eta^{\tilde{t}-t})$. This implies linear convergence.[2] However, this only holds, if the loss reaches zero, i.e. if the data is separable.

In this paper we propose a family of algorithms that are able to generate a combined hypothesis f converging to the minimum of $\mathrm{G}[f]$ (if it exists), which is a functional depending on the outputs of the function f evaluated on the training set. Special cases are AdaBoost, Logistic Regression and LS-Boost. While assuming mild conditions on the base learning algorithm and the loss function G, we can show *linear convergence rates* [Lue84] of the type $\mathrm{G}[f_{t+1}] - \mathrm{G}[f^*] \leq \eta(\mathrm{G}[f_t] - \mathrm{G}[f^*])$ for some fixed $\eta \in [0, 1)$. This means that the difference to the minimum loss converges exponentially fast to zero (in the number of iterations). A similar convergence has been proven for AdaBoost in the special case of separable data [FS97], although the constant η shown in [FS97] can be considerable smaller [see also DH00]. To prove the convergence of leveraging, we exploit results of Luo & Tseng [LT92] for a variant of the *Gauss-Southwell method* known from numerical optimization.

Since in practice the hypothesis space $\mathcal{H}$ can be reasonable large, ensemble learning algorithms without any regularization often suffer from overfitting [ROM98, Fri99, BDST00]. Here, the complexity can only be controlled by the size of the base hypothesis space or by early stopping after a few iterations. However, it has been shown that *shrinkage regularization* implied by penalizing some norm of the hypothesis coefficients is the favorable strategy [Cop83, Vap95, Fri99]. We therefore extend our analysis to the case of ℓ_1-norm regularized loss functions. With a slight modification this leads to a family of converging algorithms that e.g. includes the Leveraged Vector Machine [Sin00] and a variant of LASSO [Tib94].

In the following we briefly review AdaBoost, Logistic Regression, and LS-Boost in Section 2 and cast them in a common framework. In Section 3 we present our main results. After relating these results to leveraging algorithms, we present an extension to regularized cost functions in Section 4 and finally conclude.

2 Leveraging algorithms revisited

We first briefly review some of the most well known leveraging algorithms for classification and regression. For more details see e.g. [FS97, FHT98, Fri99, DH99]. We work with Algorithm 1 as a template of a *generic* leveraging algorithm, since these algorithms have the same algorithmical structure. Finally, we will generalize the problem and extend the notation.

2.1 AdaBoost & Logistic Regression

Both methods are designed for classification tasks. In each iteration they call a base learning algorithm on the training set $\mathrm{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subseteq \mathcal{X} \times \{-1, +1\}$ (cf. step 3a in Algorithm 1). Here a weighting $\mathbf{d}^t = [d_1^t, \ldots, d_N^t]$

[2] For AdaBoost holds $\tilde{t} = 0$, which has already been shown before in [FS97].

on the sample is used that is recomputed in each iteration t. The base learner is expected to return a hypothesis $\hat{h}_t$ from some hypothesis space[3] $\mathcal{H} := \{h_j \mid h_j : \mathcal{X} \mapsto \{-1, +1\}, j = 1, \ldots, J\}$ that has a small *weighted classification error* $\epsilon_t = \sum_{n=1}^{N} |d_n^t| \mathbf{I}(y_n \neq \hat{h}_t(\mathbf{x}_n))$ [FS97, FHT98], where $\mathbf{I}(\text{true}) = 1$ and $\mathbf{I}(\text{false}) = 0$. Notice, that different from common convention, we include the label y_n in the weighting d_n to make the presentation simpler. We therefore use $|d_n^t|$ instead of d_n^t. An equivalent requirement can be formulated in terms of the *edge*, which is given by $\hat{\gamma}_t = \sum_{n=1}^{N} y_n |d_n^t| \hat{h}_t(\mathbf{x}_n) = \sum_{n=1}^{N} d_n^t \hat{h}_t(\mathbf{x}_n)$ and is an affine transformation of ϵ_t [Bre97]. After selecting the hypothesis, its weight $\hat{\alpha}_t$ is computed such that it minimizes a certain functional (cf. step 3b). For AdaBoost this is

$$\mathrm{G}^{AB}(\hat{\alpha}) = \sum_{n=1}^{N} \exp\left\{-y_n \left(\hat{\alpha}\hat{h}_t(\mathbf{x}_n) + f_{t-1}(\mathbf{x}_n)\right)\right\} \tag{1}$$

and for Logistic Regression it is

$$\mathrm{G}^{LR}(\hat{\alpha}) = \sum_{n=1}^{N} \log\left\{1 + \exp\left(-y_n(\hat{\alpha}\hat{h}_t(\mathbf{x}_n) + f_{t-1}(\mathbf{x}_n))\right)\right\} \ , \tag{2}$$

where f_{t-1} is the combined hypothesis of the previous iteration given by

$$f_{t-1}(\mathbf{x}_n) = \sum_{r=1}^{t-1} \hat{\alpha}_r \hat{h}_r(\mathbf{x}_n). \tag{3}$$

For AdaBoost it has been shown that $\hat{\alpha}_t$ in (1) can be computed analytically. For LR there exist analytic approximations of the minimum of (2). Based on the new combined hypothesis the weighting $\mathbf{d}$ on the sample is updated:

$$d_n^{t+1} = y_n \exp(-y_n f_t(\mathbf{x}_n)) \qquad \text{and} \qquad d_n^{t+1} = y_n \frac{\exp(-y_n f_t(\mathbf{x}_n))}{1 + \exp(-y_n f_t(\mathbf{x}_n))}, \tag{4}$$

for AdaBoost and Logistic Regression, respectively. The initial weighting $\mathbf{d}^1$ is a fixed constant, which could in principle be arbitrary. In particular, $d_n^1 = 1$ and $d_n^1 = \frac{1}{2}$ $(n = 1, \ldots, 1)$, respectively, can be used resulting from (4), when setting $f_0 = \mathbf{0}$.

In [SS98] the so-called *confidence rated boosting* has been proposed. Here one uses real valued base hypotheses. One finds the hypothesis weight $\hat{\alpha}_t$ and the hypothesis $\hat{h}_t$ in parallel, such that (1) is minimized. This algorithm is similar to the next approach.

2.2 Least-Square-Boost

This algorithm is designed to solve regression tasks [Fri99]. In this case $\mathrm{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subseteq \mathcal{X} \times \mathcal{Y}$ and some finite hypothesis set $\mathcal{H} := \{h_j \mid h_j :$

[3] Notice that $\mathcal{H}$ always contains only a finite number of *different* hypotheses when evaluated on the training set and is effectively finite [BDST00].

Algorithm 1 – A Leveraging algorithm for the loss function G.

1. **Input:** $\mathrm{S} = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\rangle$, No. of Iterations T
   Loss function $\mathrm{G} : \mathbb{R}^N \to \mathbb{R}$
2. **Initialize:** $f_0 \equiv 0$, $d_n^1 = \mathrm{g}'(y_n, f_0(\mathbf{x}_n))$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots, T$,
   (a) Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $\hat{h}_t : \mathcal{X} \to \mathcal{Y}$
   (b) Set $\hat{\alpha}_t = \mathrm{argmin}_{\alpha \in \mathbb{R}} G[f_t + \alpha \hat{h}_t]$
   (c) Update $f_{t+1} = f_t + \hat{\alpha}_t \hat{h}_t$ and $d_n^{t+1} = \mathrm{g}'(y_n, f_{t+1}(\mathbf{x}_n l))$, $n = 1, \ldots, N$
4. **Output:** f_T

$\mathcal{X} \mapsto \mathcal{Y}, j = 1, \ldots, J\}$, where $\mathcal{Y}$ is a bounded subset of $\mathbb{R}$. LS-Boost works in a similar way as AdaBoost and LR. It first selects a hypothesis solving

$$\hat{h}_t = \underset{\hat{h} \in \mathcal{H}}{\mathrm{argmin}} \frac{1}{2} \sum\nolimits_{n=1}^{N} \left(d_n^t - \hat{h}(\mathbf{x}_n) \right)^2, \tag{5}$$

and then finds the hypothesis weight $\hat{\alpha}_t$ by minimizing the squared error of the new combined hypothesis:

$$\mathrm{G}^{LS}(\hat{\alpha}) = \frac{1}{2} \sum\nolimits_{n=1}^{N} \left(y_n - f_{t-1}(\mathbf{x}_n) - \hat{\alpha} \hat{h}_t(\mathbf{x}_n) \right)^2. \tag{6}$$

The "weighting" of the sample is computed as $d_n^{t+1} = y_n - f_t(\mathbf{x}_n)$, which is actually the residual of f_t [Fri99]. In a second version of LS-Boost, the base hypothesis and its weight are found simultaneously by solving [Fri99]:

$$[\hat{h}_t, \hat{\alpha}_t] = \underset{\hat{\alpha} \in \mathbb{R}, \hat{h} \in \mathcal{H}}{\mathrm{argmin}} \frac{1}{2} \sum\nolimits_{n=1}^{N} \left(y_n - f_{t-1}(\mathbf{x}_n) - \hat{\alpha} \hat{h}(\mathbf{x}_n) \right)^2 \tag{7}$$

We call this the *maximum improvement* scheme. Since in (7) one reaches a lower loss function value than in (5) and (6), it might be the favorable strategy. We will in fact show that using (5) and (6), one may *not* converge (cf. Section 3.2).

2.3 The General Case

These algorithms can be summarized[4] in Algorithm 1 for an appropriate function-pair G and g′: plug-in $\mathrm{G}[f] = \sum_{n=1}^{N} g(y_n, f(\mathbf{x}_n))$ and choose g as $\mathrm{g}(y, f(\mathbf{x}))$ $= \exp(-yf(\mathbf{x}))$ for AdaBoost (cf. (1)), and $\mathrm{g}(y, f(\mathbf{x})) = \log(1 + \exp(-yf(\mathbf{x})))$ for Logistic Regression (cf. (2)) and $\mathrm{g}(y, f(\mathbf{x})) = \frac{1}{2}(y - f(\mathbf{x}))^2$ for LS-Boost (cf. (6)). It can easily be verified that the function g′, used for computing the weights $\mathbf{d}$, is the derivative of g with respect to the second argument [Bre97, Fri99], i.e. $\mathrm{g}'(y, o) = [0, 1](\nabla_{[y,o]}\, \mathrm{g}(y, o))$.

[4] Here, the maximum improvement scheme as in (7) is slightly degenerated, but can, however, be understood as a special case (cf. Section 3.2).

2.4 The Optimization Problem

It has been argued in [Bre97, ORM98, FHT98, MBBF00] and finally shown in [CSS00] that AdaBoost and Logistic Regression asymptotically converge to a combined hypothesis f minimizing the respective loss G on the training sample, where f is a linear combination of hypotheses from $\mathcal{H}$, i.e.

$$f_{\boldsymbol{\alpha}} \in \mathrm{lin}(\mathcal{H}) := \left\{ \sum_{j=1}^{J} \alpha_j h_j \mid h_j \in \mathcal{H}, \alpha_j \in \mathbb{R} \right\}.$$

Thus, they solve the optimization problem:

$$\min_{f \in \mathrm{lin}(\mathcal{H})} \mathrm{G}[f] = \min_{\boldsymbol{\alpha} \in \mathbb{R}^J} \mathrm{G}(H\boldsymbol{\alpha}), \tag{8}$$

where we defined a matrix $H \in \mathbb{R}^{N \times J}$ with $H_{nj} = h_j(\mathbf{x}_n)$.

To avoid confusions, note that hypotheses and coefficients generated during the iterative algorithm are marked by a hat. In the algorithms discussed so far, the optimization takes place by employing the leveraging scheme outlined in Algorithm 1. The output of such an algorithm is a set of pairs $(\hat{\alpha}_t, \hat{h}_t)$ and a combined hypothesis $f_t(\mathbf{x}) = \sum_{r=1}^{t} \hat{\alpha}_r \hat{h}_r(\mathbf{x})$. With

$$\alpha_j^t = \sum_{r=1}^{t} \hat{\alpha}_r \mathbf{I}(\hat{h}_r = h_j), \quad j = 1, \ldots, J, \tag{9}$$

it is easy to verify that $\sum_{r=1}^{t} \hat{\alpha}_r \hat{h}_r(\mathbf{x}) = \sum_{j=1}^{J} \alpha_j^t h_j(\mathbf{x})$, which is in $\mathrm{lin}(\mathcal{H})$ (note the missing hat, cf. (3)).

2.5 Preliminaries

Throughout the paper we assume the loss function G is of the form

$$\mathrm{G}[f_{\boldsymbol{\alpha}}] = \mathrm{G}(H\alpha) = \sum_{n=1}^{N} \mathrm{g}(y_n, f_{\boldsymbol{\alpha}}(\mathbf{x}_n)),$$

Although, this assumption is not absolutely necessary, the presentation becomes easier. In [CSS00] a more general case of Legendre-type cost functions is considered. However, note that additive loss functions are commonly used, if one considers samples drawn i.i.d. from some source.

We assume that each element H_{nj} and y_n is finite ($j = 1, \ldots, J$, $n = 1, \ldots, N$) and H does not contain a zero column. Furthermore, the function $\mathrm{g}(y, \cdot) : \mathbb{R} \to \mathbb{R}$ is assumed to be strictly convex for all $y \in \mathcal{Y}$.

For simplicity we assume for the rest of the paper that $\mathcal{H}$ is finite and complementation closed, i.e. for every $h \in \mathcal{H}$ there exists also $-h \in \mathcal{H}$. The assumption on the finiteness is not crucial for classification (cf. footnote 3). For regression problems the hypothesis space might be infinite. This case has explicitely been analyzed in [RDB00, Rät01] and goes beyond the scope of this paper.

3 Main Result

We now state some results known from the field of numerical optimization. Then we show how the reviewed leveraging algorithms fit into this framework.

3.1 Coordinate Descent

The idea of coordinate descent is to iteratively select a coordinate, say the j-th, and find α_j such that some functional $\mathrm{F}([\alpha_1, \dots, \alpha_j, \dots, \alpha_T]^\top)$ is minimized with respect to α_j. There exist several different selection strategies for the coordinates [e.g. Lue84, Ber95], however, we are in particular interested in the *Gauss-Southwell*-type (GS) selection scheme: It selects the coordinate that has the largest absolute value in the gradient vector $\boldsymbol{\mu} := \nabla \mathrm{F}(\boldsymbol{\alpha})$, i.e. $j = \operatorname{argmax}_{j'=1,\dots,J} |\mu_{j'}|$. Instead of doing steps in the direction of the negative gradient as in standard *gradient descent* methods, one only changes the variable that has the largest gradient component.

We start with the following general convergence result, which seemed to be fallen into oblivion even in the optimization community. It will be very useful in the analysis of leveraging algorithms.

Theorem 1 (Convergence of Coordinate Descent [LT92]). *Suppose* $\mathrm{G} : \mathbb{R}^N \to \mathbb{R}$ *is twice continuously differentiable and strictly convex on* $\operatorname{dom} \mathrm{G}$. *Assume that* $\operatorname{dom} \mathrm{G}$ *is open, the set of solutions* $\mathcal{S}^* \subset \mathbb{R}^J$ *to*

$$\min_{\boldsymbol{\alpha} \in \mathcal{S}} \mathrm{F}(\boldsymbol{\alpha}), \quad \text{where} \quad \mathrm{F}(\boldsymbol{\alpha}) = \mathrm{G}(H\boldsymbol{\alpha}) + \gamma^\top \boldsymbol{\alpha} \tag{10}$$

is not empty, where $H \in \mathbb{R}^{N \times J}$ *is a fixed matrix having no zero column,* $\gamma \in \mathbb{R}^J$ *fixed and* $\mathcal{S} \subseteq \mathbb{R}^J$ *is a (possibly unbounded)* box-constrained *set. Furthermore assume that the Hessian* $\nabla^2 \mathrm{G}(H\boldsymbol{\alpha}^*)$ *is a positive definite matrix for all* $\boldsymbol{\alpha}^* \in \mathcal{S}^*$. *Let* $\{\boldsymbol{\alpha}^t\}$ *be the sequence generated by coordinate descent, where the coordinate selection* $j_1, j_2, \dots$ *satisfies the following* β*-optimality criterion:*

$$|\alpha_{j_t}^{t+1,j_t} - \alpha_{j_t}^t| \geq \beta \max_{j=1,\dots,J} |\alpha_j^{t+1,j} - \alpha_j^t| \tag{11}$$

for some $\beta \in (0, 1]$, *where* $\alpha_j^{t+1,j}$ *is the optimal value of* α_j^{t+1} *if it would be selected,*[5] *i.e.*

$$\alpha_j^{t+1,j} := \min_{\alpha \in \mathcal{S}_j} \mathrm{G}\left(H\boldsymbol{\alpha}^t + H_j(\alpha - \alpha_j^t)\right) + \gamma_j \alpha. \tag{12}$$

Then, $\{\boldsymbol{\alpha}^t\}$ *converges to an element in* $\mathcal{S}^*$.

This theorem is slightly more general than we currently need.[6] In this section we consider only $\mathcal{S} = \mathbb{R}^J$ and $\gamma = 0$. To simplify notation let in the following $\partial_j = \left|\frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_j}\right| = \left|\frac{\partial \mathrm{G}(H\boldsymbol{\alpha}^t)}{\partial \alpha_j} + \gamma_j\right|$.

The coordinate selection defined in the theorem is slightly different from the Gauss-Southwell selection rule described before. We therefore need the following:

[5] For convenience, let the rest of the vector be unchanged, i.e. $\alpha_{j'}^{t+1,j} = \alpha_{j'}^t$, for all $j \neq j'$.

[6] [LT92] also prove the (linear) convergence for an almost cyclic selection scheme.

Proposition 2 (GS scheme on $\mathbb{R}^J$). *Assume the conditions on* G, F, H, γ *and* $\mathcal{S} = \mathbb{R}^J$ *as in Theorem 1. Let* $\mathcal{S}^b$ *be a convex subset of* $\mathcal{S}$*. Then a coordinate selection* $j_1, j_2, \dots$ *starting at* $\boldsymbol{\alpha}^0$ *satisfying*

1. $\partial_{j_t} \geq \delta \max_{j=1,\dots,J} \partial_j$, *for some fixed* $\delta \in (0,1]$ $(t = 1, 2, \dots)$,
2. $\boldsymbol{\alpha}^{t,j} \in \mathcal{S}^b$ $(j = 1, \dots, J,\ t = 1, 2, \dots)$,
3. $\dfrac{\partial^2 \mathrm{G}(H\boldsymbol{\alpha})}{\partial \alpha_j^2} \leq \eta_u$, *and* $\dfrac{\partial^2 \mathrm{G}(H\boldsymbol{\alpha})}{\partial \alpha_j^2} \geq \eta_l$, *for some fixed* $\eta_l, \eta_u > 0$ *and any* $\boldsymbol{\alpha} \in \mathcal{S}^b$, $j = 1, \dots, J$

also satisfies condition (11) *of Theorem 1.*

Thus the *approximate Gauss-Southwell method on* $\mathbb{R}^J$ as defined above converges in the sense of Theorem 1.

Proof. We first bound $|\lambda_j^*| := |\alpha_j^{t+1,j} - \alpha_j^t|$. By condition 3 we have

$$\left| \frac{\partial \mathrm{G}(H\boldsymbol{\alpha}^t)}{\partial \alpha_j} - \frac{\partial \mathrm{G}(H\boldsymbol{\alpha}^t + \lambda_j H_j)}{\partial \alpha_j} \right| \geq |\lambda_j| \eta_l.$$

By (12) holds $\frac{\partial \mathrm{G}(H\boldsymbol{\alpha}^t + \lambda_j^* H_j)}{\partial \alpha_j} + \gamma_j = 0$ and we have $|\lambda_j^*| \leq \frac{1}{\eta_l} \partial_j$, $j = 1, \dots, J$. By a symmetric argument we also have $|\lambda_{j_t}^*| \geq \frac{1}{\eta_u} \partial_{j_t}$, for all $j = 1, \dots, J$. Condition 1 implies

$$\frac{|\alpha_{j_t}^{t+1,j_t} - \alpha_{j_t}^t|}{|\alpha_j^{t+1,j} - \alpha_j^t|} = \frac{|\lambda_{j_t}^*|}{|\lambda_j^*|} \geq \frac{\frac{1}{\eta_u}\partial_{j_t}}{\frac{1}{\eta_l}\partial_j} \geq \frac{\delta \eta_l}{\eta_u} \qquad \forall j = 1, \dots, J.$$

Thus, (11) is satisfied for $\beta = \frac{\delta \eta_l}{\eta_u} > 0$ and we can apply Theorem 1. □

To show the convergence of the second variant of LS-Boost (using the maximal improvement (MI) scheme, cf. (7)), we need the following:

Proposition 3 (MI scheme on $\mathbb{R}^J$). *Assume the conditions on* G, F *and* H *as in Theorem 1 are satisfied. Let* $\mathcal{S} = \mathbb{R}^J$ *and* $\mathcal{S}^b$ *be a convex subset of* $\mathcal{S}$*. Then a coordinate selection* $j_1, j_2, \dots$ *starting at* $\boldsymbol{\alpha}^0$ *that satisfies*

1. $\mathrm{F}(\boldsymbol{\alpha}^t) - \mathrm{F}(\boldsymbol{\alpha}^{t+1,j_t}) \geq \delta \max_{j=1,\dots,J} \left(\mathrm{F}(\boldsymbol{\alpha}^t) - \mathrm{F}(\boldsymbol{\alpha}^{t+1,j}) \right)$ *for some fixed* $\delta \in (0,1]$ $(\forall t = 1, 2, \dots)$,

and conditions 2–3 of Proposition 2, also satisfies (11) *of Theorem 1.*

Thus the *approximate maximal improvement scheme* as defined above converges in the sense of Theorem 1.

Proof. Let $\Delta_j = \mathrm{F}(\boldsymbol{\alpha}^t) - \mathrm{F}(\boldsymbol{\alpha}^{t+1,j}) \geq 0$. From the proof of Proposition 2 we have $|\lambda_j^*| \leq \frac{1}{\eta_l}\partial_j$ and $|\lambda_j^*| \geq \frac{1}{\eta_u}\partial_j$ for all $j = 1, \dots, J$. We first bound Δ_j in terms of ∂_j in both directions:

1. By condition 3 we have $\left|\left|\frac{\partial \mathrm{G}(H\boldsymbol{\alpha}^t + \lambda H_j)}{\partial \alpha_j} + \gamma_j\right| - \partial_j\right| \leq |\lambda|\eta_u$ for all λ and $j = 1, \ldots, J$. Since the gradient does not change the sign for smaller steps than $|\lambda_j^*|$, holds $\left|\frac{\partial \mathrm{G}(H\boldsymbol{\alpha}^t + \lambda H_j)}{\partial \alpha_j} + \gamma_j\right| \geq \partial_j - \eta_u|\lambda|$ for all $|\lambda| \leq |\lambda_j^*|$, $\lambda\lambda_j^* \geq 0$.

   It holds $\Delta_j \geq \left|\int_0^{\lambda_j} \left|\frac{\partial \mathrm{G}(H\boldsymbol{\alpha}^t + \lambda H_j)}{\partial \lambda} + \gamma_j\right| \mathrm{d}\lambda\right|$ for all $|\lambda_j| \leq \lambda_j^*$ and with equality if $\lambda_j = \alpha_j^t - \alpha_j^{t+1,j}$. Thus

$$\Delta_j \geq \int_0^{|\lambda_j|} (\partial_j - \eta_u|\lambda|)\mathrm{d}\lambda = \partial_j|\lambda_j| - \frac{\eta_u}{2}|\lambda_j|^2 \tag{13}$$

   for all $|\lambda| \leq |\lambda_j^*|$. Using $|\lambda_j^*| \geq \frac{1}{\eta_u}\partial_j$, we may set $|\lambda_j| = \frac{\partial_j}{\eta_u}$ and get $\Delta_j \geq \frac{\partial_j^2}{\eta_u} - \frac{\partial_j^2}{2\eta_u} = \frac{\partial_j^2}{2\eta_u}$ and therefore $\partial_j \leq \sqrt{2\Delta_j\eta_u}$.

2. Furthermore, by convexity holds $\Delta_j \leq |\lambda_j^*|\partial_j$ and therefore $\partial_j \geq \sqrt{\Delta_j\eta_l}$ for all $j = 1, \ldots, J$ – in particular for j_t.

Using both inequalities and assumption 1, we can bound for all $j = 1, \ldots, J$:

$$\frac{\partial_{j_t}}{\partial_j} \geq \frac{\sqrt{\Delta_{j_t}\eta_l}}{\sqrt{2\Delta_j\eta_u}} \geq \frac{\sqrt{\delta\eta_l}}{\sqrt{2\eta_u}}. \tag{14}$$

Hence, we can apply Proposition 2 and conclude that (11) is satisfied for $\beta = \sqrt{\delta}\frac{\eta_l^{3/2}}{\sqrt{2}\eta_u^{3/2}}$. □

Note that we have also shown that Proposition 3 implies Proposition 2. By inverting (14) it is straight forward to see, that also the opposite is true, i.e. Proposition 2 implies Proposition 3.

Finally we can also state a rate of convergence, which is surprisingly not worse than the rates for standard gradient descent methods:

Theorem 4 (Rate of Convergence, [LT92]). *Under the conditions of Theorem 1 and the assumption that* F *is strongly convex along the path of the* $\boldsymbol{\alpha}^t$*'s, we have*

$$\epsilon_{t+1} := \mathrm{F}(\boldsymbol{\alpha}^{t+1}) - \mathrm{F}(\boldsymbol{\alpha}^*) \leq \left(1 - \frac{1}{\eta}\right)(\mathrm{F}(\boldsymbol{\alpha}^t) - \mathrm{F}(\boldsymbol{\alpha}^*)), \tag{15}$$

where $\boldsymbol{\alpha}^t$ *is the estimate after the* t*-th coordinate descent step,* $\boldsymbol{\alpha}^*$ *denotes a optimal solution, and* $1 < \eta < \infty$*. In particular at iteration* t *holds:*

$$\epsilon_t \leq \left(1 - \frac{1}{\eta}\right)^t \epsilon_0.$$

Following [LT92] one can show that the constant η is at most $\mathcal{O}\left(\frac{\kappa^2\rho\sigma J^4N^2}{\beta^2}\right)$. Here, ρ is the Lipschitz constant of $\nabla \mathrm{G}$, σ is a lower bound on the eigenvalues of $\nabla^2 \mathrm{G}(\boldsymbol{\alpha}^t)$, $t = 1, \ldots$ and κ is a constant that only depends on H [cf. LT92, Hof52, Rob73, MS87] for details). While the upper bound on η can be rather large – making the convergence slow – it is important to note (i) that this is only a rough estimate of the true constant and (ii) still guarantees an *exponential decrease* in the error functional with the number of iterations.

3.2 Leveraging and Coordinate Descent

We now return from the abstract convergence results in Section 3.1 to our examples of leveraging algorithms. We show how to retrieve the Gauss-Southwell algorithm on $\mathbb{R}^J$ as a part of Algorithm 1. For now we set $\gamma = 0$ and have $\mathrm{F} \equiv \mathrm{G}$. The gradient of $\mathrm{G}(H\boldsymbol{\alpha})$ with respect to α_j is given by

$$\begin{aligned}\frac{\partial \mathrm{F}(\boldsymbol{\alpha})}{\partial \alpha_j} &= \sum_{n=1}^{N} \frac{\partial}{\partial \alpha_j} \mathrm{g}(y_n, f_{\boldsymbol{\alpha}}(\mathbf{x}_n)) = \sum_{n=1}^{N} \mathrm{g}'(y_n, f_{\boldsymbol{\alpha}}(\mathbf{x}_n)) \frac{\partial}{\partial \alpha_j} f_{\boldsymbol{\alpha}}(\mathbf{x}_n) \\ &= \sum_{n=1}^{N} \mathrm{g}'(y_n, f_{\boldsymbol{\alpha}}(\mathbf{x}_n)) h_j(\mathbf{x}_n) = \sum_{n=1}^{N} d_n h_j(\mathbf{x}_n) \qquad (16)\end{aligned}$$

where d_n is given as in step 3c of Algorithm 1. Thus, the coordinate with maximal absolute gradient corresponds to the hypothesis with largest absolute edge (see definition on page 3). However, according to Proposition 2 and 3 we need to assume less on the base learner. It either has to return a hypothesis that (approximately) maximizes the edge, or alternatively (approximately) minimizes the loss function:

Definition 5 (δ-Optimality). *Let $\mathcal{H}$ be a finite, complementation closed hypothesis space and* G *be a functional as defined above. A base learning algorithm* L *is called* δ-optimal *for $\mathcal{H}$ and* G*, if it always returns a hypothesis $h_j \in \mathcal{H}$ that either satisfies*

1. $\gamma_j \geq \delta \max_{j'=1,\ldots,J} \gamma_{j'}$, *where* $\gamma_j = \sum_{n=1}^{N} d_n h_j(\mathbf{x}_n)$ *for any weighting* $\mathbf{d} \in \mathrm{range}(\nabla \mathrm{G})$,

or

2. $\mathrm{F}(\boldsymbol{\alpha}^t) - \mathrm{F}(\boldsymbol{\alpha}^{t+1,j}) \geq \delta \max_{j'=1,\ldots,J} \left(\mathrm{F}(\boldsymbol{\alpha}^t) - \mathrm{F}(\boldsymbol{\alpha}^{t+1,j'})\right)$ *for any*[7] $\boldsymbol{\alpha}^t \in \mathbb{R}_+^J$,

for some fixed $\delta \in (0, 1]$ and never returns a hypothesis with $h(\mathbf{x}_n) = 0$ for all $n = 1, \ldots, N$.

Remark 6. *Since we have assumed $\mathcal{H}$ is closed under complementation, there always exist two hypotheses having the same absolute gradient (h and $-h$).*

[7] Here we only require $\boldsymbol{\alpha}^t \geq 0$ since $\mathcal{H}$ is closed under complementation.

To fulfill condition 1 of Proposition 3, we therefore only need to consider the hypothesis with maximum edge *as opposed to the* maximum absolute edge*. For classification this means that, if the base learner returns the hypothesis with (approximately) smallest weighted training error, this condition is satisfied.*

We will show in Theorem 7 that Definition 5 states sufficient conditions on the base learner to achieve the desired convergence. Although Definition 5 is not a necessary condition for convergence, it is easy to construct cases where a violation of the conditions in Definition 5 destroys the convergence. For instance, suppose the base learner would e.g. violate condition 1 and there exists no $\delta > 0$. Then the base learning algorithm is allowed to return a hypothesis that does not change the solution, whereas the solution might not be optimal.

We can finally show the general convergence result for leveraging algorithms without regularization:

Theorem 7. *Given a training sample S and a finite and complementation closed hypothesis space $\mathcal{H}$. Let $\mathrm{G} : \mathbb{R}^N \to \mathbb{R}$ be twice differentiable, strongly convex*[8] *and its Hessian be uniformly upper bounded on any bounded subset of $\mathbb{R}^N$. Suppose Algorithm 1 generates a sequence of combined hypotheses $f_{\boldsymbol{\alpha}^0}, f_{\boldsymbol{\alpha}^1}, \dots$ (cf. (9)) using a δ-optimal learning algorithm* L *for $\mathcal{H}$ and* G*. If any solution of* (8) *is finite and $\limsup_{t\to\infty} \|\boldsymbol{\alpha}^t\| < \infty$, then any limit point of $\{\boldsymbol{\alpha}^t\}$ is a solution of* (8) *and the sequence converges* linearly *in the sense of Theorem 4.*

Proof. Let $\mathcal{S}^*$ be the solution set of (8) and

$$\mathcal{S}^b = \mathrm{conv}\{\boldsymbol{\alpha}^{t,j} \mid j = 1, \dots, J, t = 1, 2, \dots\}.$$

We claim $\mathcal{S}^b$ is bounded. By the assumption on the finiteness of the $\boldsymbol{\alpha}^t$'s, the set $\mathrm{conv}\{\boldsymbol{\alpha}^t \mid t = 1, 2, \dots\}$ is bounded. We therefore need to show that for any finite $\boldsymbol{\alpha}^t$, the vectors $\boldsymbol{\alpha}^{t+1,j}$, $j = 1, \dots, J$ are each finite. We consider loss functions of the form $\mathrm{G}(\mathbf{o}) = \sum_{n=1}^{N} \mathrm{g}_{y_n}(o_n)$, where $\mathrm{g}_{y_n}(o)$ is strongly convex at finite o. There are two cases: g_{y_n} grows (1) in both directions to infinity, i.e. $g_{y_n}(\pm\infty) = \infty$, and (2) it grows only in one direction to infinity. The third case that it grows in neither direction to ∞ cannot happen, since g_{y_n} is strongly convex.

(1) If $\boldsymbol{\alpha}$ is finite, $\mathrm{G}(H\boldsymbol{\alpha})$ is finite. Denote by H_j the j-th column of H. A single step in one direction j cannot be optimal at infinity, since $H_j \neq \mathbf{0}$ by assumption and so $H\boldsymbol{\alpha}$ and therefore $\mathrm{G}(H\boldsymbol{\alpha})$ would be $+\infty$. Thus, the infinite step cannot be optimal.

(2) Let

$$\hat{y}_n := \begin{cases} +1 & g_{y_n}(-\infty) = \infty \\ -1 & g_{y_n}(+\infty) = \infty \end{cases},$$

then $\alpha_j^{t+1,j}$ can only be infinite, if $H_{j,n}\hat{y}_n \geq 0$ for all $n = 1, \dots, N$. Otherwise the objective would grow to infinity. Since g_{y_n} is monotonically

[8] A function is called strongly convex on $\mathcal{S}$, if there exists a fixed $\eta > 0$ such that the matrix $\nabla^2_{\mathbf{o}} G(\mathbf{o}) - \eta\mathbf{I}$ is positive for all $\mathbf{o} \in \mathcal{S}$ [cf. Ber95, p. 563]. Here, we require strong convexity on any bounded subset or $\mathbb{R}^N$.

decreasing in direction $\hat{y}_n \infty$, any solution $\boldsymbol{\alpha}^*$ does not get worse when increasing α_j^* to infinity. Thus there must exist a infinite solution, which is a contradiction to the assumption.

Since this holds for any finite $\boldsymbol{\alpha}$, it also holds for any limit point of $\{\boldsymbol{\alpha}^t\}$ and any solution. We can therefore conclude that $\mathcal{S}^b$ is bounded. Then holds condition 2 of Proposition 2.

Each element H_{nj} of H is finite. Thus the set $\mathcal{S}^o = \{H\boldsymbol{\alpha} \mid \boldsymbol{\alpha} \in \mathcal{S}^b\}$ is bounded. Since G is strongly convex on $\mathcal{S}^o$ and $\|\nabla^2_{\mathbf{o}} G(\mathbf{o})\|$ is bounded by assumption, condition 3 of Proposition 2 is satisfied.

By Definition 5 and using the complementation closeness of $\mathcal{H}$, we may apply Proposition 2 or Proposition 3 and can conclude that the sequence $\boldsymbol{\alpha}^0, \boldsymbol{\alpha}^t, \ldots$ satisfies (11).

Since G is strongly convex on $\mathcal{S}^*$, the Hessian $\nabla^2 \mathrm{G}(H\boldsymbol{\alpha}^*)$ is a strictly positive matrix for all $\boldsymbol{\alpha}^* \in \mathcal{S}^*$. Thus, we can apply Theorems 1 & 4 and have proven the theorem. □

For the special cases described in Section 2 we get:

Corollary 8. *Given a training sample S and a finite hypothesis space $\mathcal{H}$. Suppose AdaBoost, the Logistic Regression algorithm or LS-Boost (with selection scheme (7)) as described in Section 2 (cf. Algorithm 1) generate a sequence of hypotheses $\hat{h}_1, \hat{h}_1, \ldots$ and weights $\hat{\alpha}_1, \hat{\alpha}_2, \ldots$ using a δ-optimal base learner. Assume every solution of (8) using the respective loss function is finite. Then any limit point of $\{f_t\}$ solves (8).*

Proof. We first need to show that the loss functions of AdaBoost, Logistic regression and LS-Boost are strongly convex on any bounded subset $\mathcal{S}^o$ of $\mathbb{R}^N$ and fulfill the conditions in Theorem 1: The loss functions can be written as $\mathrm{G}(\mathbf{o}) = \sum_{n=1}^N g(y_n, o_n)$. Therefore, the Hessian $\nabla^2_{\mathbf{o}} \mathrm{G}(\mathbf{o})$ is diagonal with elements $\frac{\partial^2}{\partial o_n^2} g(y_n, o_n)$, $n = 1, \ldots, N$. Since the second derivatives of the one dimensional functions (cf. Section 2.3) can each be lower bounded by a positive constant on a bounded subset of $\mathbb{R}$, each element of the diagonal and therefore each eigenvalue is bounded away from 0. This implies the strong convexity of G. Using similar arguments one can also uniformly upper bound $\|\nabla^2_{\mathbf{o}} \mathrm{G}(\mathbf{o})\|$. Thus, we can fulfill the conditions of Theorem 7. □

A similar result for AdaBoost and Logistic Regression has been obtained using different techniques in [CSS00], but assuming the base learner always performs *optimal* (i.e. $\delta = 1$) and also not showing convergence rates. Note that Theorem 4 also applies to AdaBoost, Logistic regression and LS-Boost under the conditions of Corollary 8.

For the selection scheme of LS-Boost given by (5) and (6), both conditions in Definition 5 *cannot* be satisfied in general, unless $\sum_{n=1}^N h_j(\mathbf{x}_n)^2$ is constant for all hypotheses. Since $\sum_{n=1}^N (d_n - h_j(\mathbf{x}_n))^2 = \sum_{n=1}^N (h_j(\mathbf{x}_n)^2 - 2 d_n h_n(\mathbf{x}_n) +$ const.), the base learner prefers hypotheses with small $\sum_{n=1}^N h_j(\mathbf{x}_n)^2$ and could therefore stop improving the objective while being not optimal (see [RDB00, Section 4.3] and [Rät01, Section 5] for more details).

Note that Corollary 8 also applies to the confidence rated boosting algorithm, if the base learner satisfies condition 2 of Definition 5 and the hypothesis space is finite.

We also have to note that for the analysis in the PAC setting the linear convergence in general is not sufficient. Here one needs that there exists an upper bound on η that does not depend on the sample size. In the case of AdaBoost is has been shown that the loss function is decreased by a factor that only depends on the achievable margin of the problem at hand. However, this requires that the data is separable.

4 Regularized Leveraging approaches

We have not yet exploited all features of Theorem 1. It additionally allows box constraints and a linear function in terms of the hypothesis coefficients. Here, we are in particular interested in ℓ_1-norm penalized loss functions of the type $\mathrm{F}(\boldsymbol{\alpha}) = \mathrm{G}(H\boldsymbol{\alpha}) + \gamma\|\boldsymbol{\alpha}\|_1$. The ℓ_1-norm regularization is frequently used in machine learning and favors sparse solutions.[9] It can be shown that at optimality most of the hypothesis coefficients will be zero [e.g. Rät01]. The LASSO algorithm for regression [Tib94], the PBVM algorithm (using the ℓ_1-norm) for classification [Sin00], the sparse (kernel) Fisher discriminant (SKFD) algorithm [MRM01], and the algorithm proposed in [Rät01, Section 5.3] are examples [see also CDS95, BMR98].

Since we assumed complementation closeness of $\mathcal{H}$, we can assume without loss of generality that a solution $\boldsymbol{\alpha}^*$ satisfies $\boldsymbol{\alpha}^* \geq \mathbf{0}$. We can therefore implement the ℓ_1-norm regularization using the linear term $\gamma^\top\boldsymbol{\alpha}$, where $\gamma = \gamma\mathbf{1}$ and $\gamma \geq 0$ is the *regularization constant*. We get the following two equivalent optimization problems:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^J} \mathrm{G}(H\boldsymbol{\alpha}) + \gamma\|\boldsymbol{\alpha}\|_1 = \min_{\mathbf{0}\leq\boldsymbol{\alpha}\in\mathbb{R}^J} \mathrm{G}(H\boldsymbol{\alpha}) + \gamma\sum_{j=1}^{J}\alpha_j, \tag{17}$$

where H as is defined in (8). Clearly, the regularization defines a structure of nested subsets of $\mathcal{H}$, where the hypothesis space is restricted to a smaller set for larger values of γ [cf. Vap95, Smo98, p. 14]. One has to use model-selection techniques to find the right trade-off between the size of the function class and the empirical error – by determining γ.

For simplicity, we consider in the following only base learning algorithms that fulfill the condition 1 in Definition 5. We conjecture that the analysis can be extended to other conditions on the base learning algorithm (in particular condition 2 in Definition 5) [see also Zha01].

[9] Roughly speaking, a reason for the induced sparseness is the fact that vectors far from the coordinate axes are "larger" with respect to the ℓ_1-norm than with respect to p-norms with $p > 1$. For example, consider the vectors $(1,0)$ and $(1/\sqrt{2}, 1/\sqrt{2})$. For the two norm, $\|(1,0)\|_2 = \|(1/\sqrt{2}, 1/\sqrt{2})\|_2 = 1$, but for the ℓ_1-norm, $1 = \|(1,0)\|_1 < \|(1/\sqrt{2}, 1/\sqrt{2})\|_2 = \sqrt{2}$. Note that using the ℓ_1-norm as regularizer the optimal solution is always a vertex solution (or can be expressed as such) and tends to be very sparse. It can easily be shown that independent of the size of a (finite) hypothesis space $\mathcal{H}$, the optimal number of hypotheses in the ensemble is not greater than the number of samples. See [RDB00, Rät01] for more details.

The constraint $\boldsymbol{\alpha} \geq 0$ causes some complications with the assumptions on the base learning algorithm. Suppose one would use a base learner that satisfies condition 1 in Definition 5. Then one would do updates only in positive directions (since the edge is large, say greater than γ, and the gradient is therefore small, say negative; cf. (16)). We therefore need to check the coefficients of previously selected hypothesis whether *not* selecting them, could violate the δ-optimality condition 1 of Proposition 2 for the negative direction. If this is the case, the algorithm selects such hypothesis for the next iteration instead of using the hypothesis returned by the base learning algorithm (cf. Algorithm 2, step 3d).

Furthermore, one has to ensure that all hypothesis coefficients α_j, $j = 1, \ldots, J$, stay positive over all iterations. Note that simply adding a constraint $\hat{\alpha}_t \geq 0$ in step 3b of Algorithm 1 is undue, since then one could never reduce a potentially too large coefficient (cf. discussion above). Note that $\hat{\alpha}_t$ might need to be negative, since it is just the *update* of some $\boldsymbol{\alpha}_j$. We therefore add the constraint $\alpha_t \geq \underline{\alpha}_t$ to the minimization with respect to α_t (cf. Algorithm 1, step 3e). The lower bound $\underline{\alpha}_t$ is zero for all hypotheses directly returned by the base learning algorithm and can be negative if the wrapper selects a previous hypothesis. This makes updates reducing some α_j possible. A hypothesis coefficient can only be too large, if the corresponding hypothesis has been selected in a previous iteration and it has not been set to zero, say in later iterations (cf. definition of $\mathcal{J}$ in step 3c of Algorithm 2). This discussion leads to Algorithm 2, where steps 3a–3d implement a *wrapper* around the original base learning algorithm.

Algorithm 2 – A Leveraging algorithm for ℓ_1-norm regularized loss G.

1. **Input:** $\mathrm{S} = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \rangle$, Number of Iterations T
   Regularization constant $\gamma > 0$, Loss function $\mathrm{G} : \mathbb{R}^N \to \mathbb{R}$
2. **Initialize:** $f_0 \equiv 0$, $d_n^1 = \mathrm{g}'(y_n, f_0(\mathbf{x}_n))$ for all $n = 1 \ldots N$
3. **Do for** $t = 1, \ldots, T$,
   (a) Train classifier on $\{S, \mathbf{d}^t\}$ and obtain hypothesis $\hat{h}_t : \mathcal{X} \to \mathcal{Y}$
   (b) Let $\hat{\gamma}_r = \sum_{n=1}^N d_n^t \hat{h}_r(\mathbf{x}_n)$ and $\underline{\alpha}_r = \sum_{s=1}^t \hat{\alpha}_r \mathbf{I}(\hat{h}_s = \hat{h}_r)$ for $r = 1, \ldots, t$
   (c) $r^* = \mathrm{argmin}_{i \in \mathcal{J}} \hat{\gamma}_i$, where $\mathcal{J} = \{i \mid i \in \{1, \ldots, t-1\} \text{ and } \underline{\alpha}_i > 0\}$.
   (d) **if** $\hat{\gamma}_t - \gamma < \gamma - \gamma_{r^*}$ **then** $\hat{h}_t = \hat{h}_{r^*}$ and $\underline{\alpha}_t = \underline{\alpha}_{r^*}$
   else $\underline{\alpha}_t = 0$
   (e) Set $\hat{\alpha}_t = \mathrm{argmin}_{\alpha \geq -\underline{\alpha}_t} G[f_t + \alpha \hat{h}_t] + \gamma \alpha$
   (f) Update $f_{t+1} = f_t + \hat{\alpha}_t \hat{h}_t$ and $d_n^{t+1} = \mathrm{g}'(y_n, f_{t+1}(\mathbf{x}_n))$, $n = 1, \ldots, N$
4. **Output:** f_T

We now show the convergence of Algorithm 2. We are faced with the following problem: Let r be the index of the hypothesis that has been selected in step 3d instead of the hypothesis returned by the base learning algorithm.

It has the smallest edge among all previously selected hypotheses. Suppose $\underline{\alpha}_t$ is greater than zero, but very small. Then the update $|\hat{\alpha}_t|$ is also very small and one cannot lower bound $|\hat{\alpha}_t|$: since there might exist another direction with smaller gradient, but larger $\underline{\alpha}$, condition (11) can not be satisfied for some fixed $\beta > 0$. One can therefore not guarantee that the selection implemented by Algorithm 2 *always* satisfies (11).

However, we will show that this cannot happen very often, since in the next iterations the coefficient is already zero and would not be selected again. This will be exploited in the following analysis of Algorithm 2. We start with:

Lemma 9. *Given a regularization constant $\gamma > 0$, a training sample $\mathrm{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subset \mathcal{X} \times \mathcal{Y}$, a finite hypothesis space $\mathcal{H}$ and the matrix H as defined in* (8)*. Assume that each element of H and y is finite. If the loss function* G *is defined on $\mathbb{R}^N$, convex on $\mathbb{R}^N$ and is lower bounded, then any solution to* (17) *is finite.*

Proof. Suppose there would exists an infinite solution $\boldsymbol{\alpha}^\infty$, then the objective of (17) must be $+\infty$ (we assumed $\gamma > 0$). Since the labels y_n, $n = 1, \ldots, N$, are finite, also $\mathrm{G}(\mathbf{0})$ is finite. Furthermore, $\boldsymbol{\alpha} = \mathbf{0}$ is feasible and $\boldsymbol{\alpha}^\infty$ cannot be optimal. □

By the same arguments we can also conclude that the $\hat{\alpha}_t$'s computed in Algorithm 2 and the $\alpha_j^{t,j}$'s as in Theorem 1 are bounded.

Extending the results of Section 3.1 for the GS scheme, we have the following result for coordinate descent on the positive quadrant:

Proposition 10 (GS scheme on $\mathbb{R}^J_+$). *Let $\mathcal{S} = \mathbb{R}^J_+ := \{\boldsymbol{\alpha} \mid \mathbf{0} \leq \boldsymbol{\alpha} \in \mathbb{R}^J\}$ and $\mathcal{S}^b$ be a convex subset of $\mathcal{S}$. Then a coordinate selection $j_1, j_2, \ldots$ starting at $\boldsymbol{\alpha}^0$ satisfies condition* (11) *of Theorem 1 for all iterations t, where (i) conditions 2–3 of Proposition 2 hold, (ii) $\alpha^t_{j_t} \geq \frac{1}{\eta_u} \frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_{j_t}}$ and (iii) the selection satisfies $j_t \in \mathcal{J}_t$ as well as the following inequality:*

$$\left| \frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_{j_t}} \right| \quad \geq \quad \delta \max_{j \in \mathcal{J}_t} \left| \frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_j} \right| \tag{18}$$

for some fixed $\delta \in (0, 1]$, where

$$\mathcal{J}_t = \left\{ j \;\middle|\; \left(\frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_j} < 0 \right) \;\vee\; (\alpha_j^t > 0) \right\}.$$

Proof. The only difference to Proposition 2 is the additional constraint $\boldsymbol{\alpha} \geq \mathbf{0}$. Thus, there can exist a dimension j with *strictly positive gradient* that is already optimal. We therefore must only take the positive gradients of those coefficients into account that are strictly positive (and therefore have been selected before – since we start at $\mathbf{0}$). Thus, in Proposition 2 we restrict the indices in (18) and can follow the proof of Proposition 2

We need to consider three cases: (a) positive update, (b) negative update and (c) negative update of a variable that violates (ii):

(a) Assume a coordinate with negative gradient, i.e. with edge greater than γ has been selected. Then the update is positive ($\hat{\alpha}_t > 0$). Since for all $j \notin \mathcal{J}_t$ holds $\alpha_j = 0$ and $\frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_j} \geq 0$, we can ensure that there does not exist an index j with a (much) larger change in the coordinate j than in j_t. This can be seen by using (18) and then following the proof of Proposition 2 using Lemma 9. This shows that (11) is satisfied for $\beta = \frac{\delta \eta_l}{\eta_u}$.

(b) Consider the other case and the update is negative ($\hat{\alpha}_t < 0$). Since $j_t \in \mathcal{J}_t$, holds $\alpha^t_{j_t} > 0$. Only if $\alpha^t_{j_t} \geq \frac{1}{\eta_u} \frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_j}$ the minimization with respect to α_{j_t} can reach the minimum without hitting the constraint $\alpha_{j_t} \geq 0$. However, assuming $\alpha_{j_t} \geq \frac{1}{\eta_u} \frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_{j_t}}$, we can follow the proof of Proposition 2 as above to show that (11) is satisfied for $\beta = \frac{\delta \eta_l}{\eta_u}$. (Roughly speaking, hitting the constraint $\alpha_{j_t} \geq 0$ does not hurt, as long as the value of α_t is large enough.)

(c) If $\alpha^t_{j_t} < \frac{1}{\eta_u} \frac{\partial \mathrm{F}(\boldsymbol{\alpha}^t)}{\partial \alpha_{j_t}}$, then $\alpha^{t+1}_{j_t} = 0$ and (11) might not be satisfied. But this case is excluded by assumption (ii).

This proves the proposition. □

Using this proposition and Lemma 9, we can conclude with the following convergence result:

Corollary 11. *Assume the conditions on* $\gamma, \mathrm{S}, \mathcal{H}, H$ *and* G *as in Lemma 9. Furthermore, assume the loss function* G *is strongly convex on any bounded subset of* $\mathbb{R}^N$. *Suppose Algorithm 2 generates a sequence of combined hypotheses* $f_{\boldsymbol{\alpha}^0}, f_{\boldsymbol{\alpha}^1}, \ldots$ *using a base learner that satisfies condition 1 of Definition 5 and never returns the zero hypothesis (as in Definition 5). Then the sequence of coefficients* $\{\boldsymbol{\alpha}^t\}$ *of the combined hypotheses* (cf. (9)) *converges* linearly *to the solution of* (17).

Proof. Using the same arguments as in Lemma 9, one can show that the sequences $\{\boldsymbol{\alpha}^{t,1}\}, \ldots, \{\boldsymbol{\alpha}^{t,J}\}$, $t = 1, 2, \ldots$, stay bounded. One may therefore define the bounded convex set $\mathcal{S}^o = \mathrm{conv}\{H\boldsymbol{\alpha}^{t,j} \mid j = 1, \ldots, J, t = 1, 2, \ldots\}$. Thus, using that G is assumed to be strongly convex on $\mathcal{S}^o$, conditions 2–3 are satisfied.

Using the assumption on the base learning algorithm, we can apply Proposition 10. Theorem 4 bounds the progress in the objective function in each individual iteration (and at any starting point). Thus, if we can find a subsequence $t_1, t_2, \ldots$ of iterations in which (11) is satisfied, the algorithm converges to the optimal solution.

By Proposition 10, if case (a) or (b) apply, (11) is satisfied for $\beta = \frac{\delta \eta_l}{\eta_u}$ and (15) guarantees the convergence rate. If case (c) applies, this is not guaranteed. However, the maximal number of subsequent iterations where (c) applies is bounded by J. Hence, at least every J-th iteration applies (15). This proves the linear convergence (in the sense of [Lue84, p. 128]; see also [LT92, p. 26]). □

5 Conclusion

We gave a unifying convergence analysis for a fairly general family of leveraging methods. These convergence results were obtained under rather mild assumptions on the base learner and, additionally, led to linear convergence rates. This was achieved by relating leveraging algorithms to the Gauss-Southwell method known from numerical optimization. Our analysis includes ℓ_1 regularized cost functions leading to a clean and general way to regularize ensemble learning.

While the main theorem used here was already proven in [LT92], its applications closes a central gap between existing algorithms and their theoretical understanding in terms of convergence.

Future investigations include the generalization to infinite hypotheses spaces, an improvement of the convergence rate η and a further analysis of regularized leveraging algorithms. Furthermore, we conjecture that our results can be extended to many other variants of boosting type algorithms proposed recently in the literature (cf. `http://www.boosting.org`).

References

[BB97] H.H. Bauschke and J.M. Borwein. Legendre functions and the method of random bregman projections. *Journal of Convex Analysis*, 4:27–67, 1997.

[BDST00] K.P. Bennett, A. Demiriz, and J. Shawe-Taylor. A column generation algorithm for boosting. In P. Langley, editor, *Proceedings, 17th ICML*, pages 65–72, San Francisco, 2000. Morgan Kaufmann.

[Ber95] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

[BMR98] P. Bradley, O. Mangasarian, and J. Rosen. Parsimonious least norm approximation. *Computational Optimization and Applications*, 11(1):5–21, 1998.

[Bre97] L. Breiman. Prediction games and arcing algorithms. Technical Report 504, Statistics Department, University of California, December 1997.

[CDS95] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. Technical Report 479, Department of Statistics, Stanford University, 1995.

[Cop83] J. Copas. Regression, prediction and shrinkage. *J.R. Statist. Soc. B*, 45:311–354, 1983.

[CSS00] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. In *Proc. COLT*, pages 158–169, San Francisco, 2000. Morgan Kaufmann.

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

[DH99] N. Duffy and D.P. Helmbold. A geometric approach to leveraging weak learners. In P. Fischer and H. U. Simon, editors, *Computational Learning Theory: 4th European Conference (EuroCOLT '99)*, pages 18–33, March 1999. Long version to appear in TCS.

[DH00] N. Duffy and D.P. Helmbold. Potential boosters? In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 258–264. MIT Press, 2000.

[FHT98] J. Friedman, T. Hastie, and R.J. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Department of Statistics, Sequoia Hall, Stanford Univerity, July 1998.

[Fri99] J.H. Friedman. Greedy function approximation. Technical report, Department of Statistics, Stanford University, February 1999.

[FS97] 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, August 1997.

[Hof52] A.J. Hoffmann. On approximate solutions of systems of linear inequalities. *Journal of Research of the National Bureau of Standards*, 49(4):263–265, October 1952.

[LT92] Z.-Q. Luo and P. Tseng. On the convergence of coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, January 1992.

[Lue84] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley Publishing Co., Reading, second edition, May 1984. Reprinted with corrections in May, 1989.

[MBBF00] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean. Functional gradient techniques for combining hypotheses. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 221–247. MIT Press, Cambridge, MA, 2000.

[MRM01] S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the Kernel Fisher algorithm. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 591–597. MIT Press, 2001.

[MS87] O.L. Mangasarian and T.H. Shiau. Lipschitz continuity of solutions of linear inequalities: Programs and complementarity problems. *SIAM Journal on Control and Optimization*, 25:583–595, 1987.

[ORM98] T. Onoda, G. Rätsch, and K.-R. Müller. An asymptotic analysis of AdaBoost in the binary classification case. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proc. of the Int. Conf. on Artificial Neural Networks (ICANN'98)*, pages 195–200, March 1998.

[Rät01] G. Rätsch. *Sparse ensemble learning.* PhD thesis, University of Potsdam, Neues Palais 10, 14469 Potsdam, Germany, August 2001. in preparation.

[RDB00] G. Rätsch, A. Demiriz, and K. Bennett. Sparse regression ensembles in infinite and finite hypothesis spaces. NeuroCOLT2 Technical Report 85, Royal Holloway College, London, September 2000. Machine Learning, to appear.

[Rob73] S.M. Robinson. Bounds for errors in the solution set of a perturbed linear programm. *Linear Algebra and its applications*, 6:69–81, 1973.

[Roc70] R.T. Rockafellar. *Convex Analysis.* Princeton Landmarks in Mathematics. Princeton University Press, New Jersey, 1970.

[ROM98] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. Technical Report NC-TR-1998-021, Royal Holloway College, University of London, UK, 1998. to appear in Machine Learning.

[Sin00] Y. Singer. Leveraged vector machines. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 610–616. MIT Press, 2000.

[Smo98] A.J. Smola. *Learning with Kernels.* PhD thesis, Technische Universität Berlin, 1998.

[SS98] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proc. COLT'98*, pages 80–91, 1998.

[Tib94] R.J. Tibshirani. Regression selection and shrinkage via the lasso. Technical report, Department of Statistics, University of Toronto, June 1994. ftp://utstat.toronto.edu/pub/tibs/lasso.ps.

[Vap95] V.N. Vapnik. *The nature of statistical learning theory.* Springer Verlag, New York, 1995.

[Zha01] T. Zhang. A general greedy approximation algorithm with applications. private communication, submitted to NIPS*01, July 2001.