On the Interconnections between Gradient Boosting and Gradient Optimization Methods

Karl Rickmer Schulte [1 2] **David Rügamer** [1 2]

Abstract

In this work, we investigate the interconnections between several gradient boosting variants and related gradient optimization methods. Building on recent theoretical results on greedy optimization, we derive novel convergence results for numerous boosting approaches. This includes convergence rates for block- and component-wise boosting for linear, ridge, and regression spline base learners and considering a variety of loss functions. Under very general assumptions on the underlying optimization problem, we derive a global linear convergence rate for a general greedy block coordinate descent (GBCD) procedure with the recently proposed Gauss-Southwell-Quadratic (GSQ) update scheme. We show that this scheme is underlying numerous boosting variants, offering new insights into the basic principles of boosting procedures. While the derived interconnection sheds light on boosting with penalized base learners in general, the derived numerical convergence results for GBCD further give rise to a first explicit convergence rate for optimization procedures solving regression spline problems.

1. Introduction

Boosting methods such as gradient boosting have demonstrated remarkable effectiveness across a wide range of domains. While boosting originated from the machine learning community (Freund & Schapire, 1996; 1997; Breiman, 1998; 1999), it has also emerged as a prominent "stage-wise additive" model fitting routine in other domains such as statistics (Friedman et al., 2000; Bühlmann & Yu, 2003; Hofner et al., 2016; Brockhaus et al., 2020). The modularity of gradient boosting to include different loss functions and base procedures renders it exceptionally adaptable and broadly applicable (Bühlmann & Hothorn, 2007). Especially in high-dimensional setups where most other methods usually break down, boosting remains applicable by utilizing block- or component-wise updates, making it one of the most prominent stage-wise additive model fitting methods for such settings (see, e.g., Rügamer et al., 2017). However, in settings such as high-dimensional linear models, alternative model fitting procedures such as the Lasso (Tibshirani, 1996) are usually still preferred since certain theoretical properties of these methods are better understood (Bühlmann et al., 2014).

Related Findings. The fact that gradient boosting can not only be seen as functional gradient descent (Friedman, 2001) but also as greedy coordinate descent yields a connection to common gradient optimization methods and inspired several investigations on the numerical convergence of boosting methods (Mason et al., 1999; Rätsch et al., 2001; Collins et al., 2002; Meir & Rätsch, 2003; Zhang & Yu, 2005). Most of these require rather strong assumptions about the objective function or consider modified versions of boosting by imposing certain restrictions on the step size and considered function space. More recent convergence results (Karimi et al., 2016; Freund et al., 2017; Locatello et al., 2018) exploit new theoretical insights about greedy coordinate descent (GCD), which also inspired our work.

Despite extensive research on various theoretical aspects of boosting over the past decades, certain areas, particularly numerical convergence for various boosting variants, are still not well understood. Through our current investigation, we aim to bridge these gaps in the theoretical understanding of boosting methods.

Our Contributions. We first derive a general convergence result for greedy block coordinate descent (GBCD). Using this result, we obtain global linear convergence rates for component- and block-wise boosting with linear, ridge and regression spline base learners, providing the first explicit numerical convergence rate for regression spline optimization. While previous research usually considered convergence in function space, we particularly focus on convergence on the parameter level, as the latter is of special interest in many domains. Further, we examine recently proposed selection and update schemes for G(B)CD, and show for the first time that these underlie the routines of

[1]Department of Statistics, LMU Munich, Munich, Germany [2]Munich Center for Machine Learning, Munich, Germany. Correspondence to: David Rügamer <david.ruegamer@lmu.de>.

Preliminary work under review. Do not distribute.

several component/block-wise gradient boosting methods. Our findings have important practical implications for applications of boosting in regularized estimation problems, that we establish both theoretically and empirically. This further allows us to derive results for boosting with other loss functions and for boosting of generalized linear models.

2. Background

2.1. Gradient Boosting

Gradient boosting (cf. Algorithm A in the Appendix) can be seen as a greedy function approximation method (Friedman, 2001) in which the functional gradient at the current function estimate is approximated by a given base procedure (Bühlmann & Hothorn, 2007). When considering the L_2 loss, the functional gradient corresponds to the current residuals, and thus boosting becomes an iterative fitting of the residuals (Bühlmann & Yu, 2003). The base procedure usually considers base learners such as linear or ridge estimators, regression splines, or trees. We will focus on the former three in our investigation as all of them share similar theoretical properties and can also be theoretically analyzed on the parameter level. Instead of using all base learners simultaneously for model updates, the base procedure can also be defined to greedily select the best-performing base learner or set of base learners at each step. This corresponds to greedy component- or block-wise selection and updates, respectively. In combination with early stopping, this can enable variable selection and result in sparse models.

2.2. Ingredients for Convergence

Throughout this paper, we are considering unconstrained problems of the form

$$\underset{x\in\mathbb{R}^d}{\operatorname{argmin}} f(x), \tag{1}$$

that are minimized with respect to the parameter $x \in \mathbb{R}^d$. To derive theoretical convergence results for gradient optimization methods, conditions such as strong convexity and L-smoothness have been considered (Boyd & Vandenberghe, 2004; Nesterov, 2004; 2012).

Definition 2.1. A function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth with $L > 0$, if $\forall x, \tilde{x} \in \mathbb{R}^d$:

$$\|\nabla f(x) - \nabla f(\tilde{x})\| \leq L\|x - \tilde{x}\|.$$

L-smoothness is sometimes also referred to as L-Lipschitz continuous gradient ∇f. From Definition 2.1 we can deduce that an L-smooth function f fulfills $\forall x, \tilde{x} \in \mathbb{R}^d$

$$f(\tilde{x}) \leq f(x) + \langle \nabla f(x), \tilde{x} - x\rangle + \frac{L}{2}\|\tilde{x} - x\|^2. \tag{2}$$

By contrast, a function that is strongly convex, or more precisely μ-strongly convex with $\mu > 0$, fulfills $\forall x, \tilde{x} \in \mathbb{R}^d$

$$f(\tilde{x}) \geq f(x) + \langle \nabla f(x), \tilde{x} - x\rangle + \frac{\mu}{2}\|\tilde{x} - x\|^2. \tag{3}$$

For twice-differentiable objective functions, the conditions in (2) and (3) provide lower and upper bounds on eigenvalues of the Hessian, $\mu I \preceq \nabla^2 f(x) \preceq LI \, \forall x \in \mathbb{R}^d$, where I is the identity matrix. More recently, Karimi et al. (2016) showed that it is sufficient to consider the PL-inequality instead of strong convexity to derive convergence rates for iterative gradient methods.

Definition 2.2. (Karimi et al., 2016) A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-PL with some $\mu > 0$, if $\forall x \in \mathbb{R}^d$:

$$\frac{1}{2}\|\nabla f(x)\|^2 \;\geq\; \mu\,(f(x) - f^*). \tag{4}$$

In Definition 2.2, f^* denotes the optimal function value of the optimization problem in (1). The notation f^* instead of $f(x^*)$ is used as the optimal function value is unique, whereas the solution x^* to the problem $f(x)$ with $f^* = f(x^*)$ does not have to be unique in case f is μ-PL. This is in contrast to strong convexity, where the solution x^* is guaranteed to be unique. Definition 2.2 is more general than strong-convexity and several important problems do not fulfill strong convexity but the more general PL-inequality.

2.3. Component-wise vs. Block-wise Greedy Selection and Updates

In the subsequent convergence analyses, we will consider both component-wise as well as block-wise gradient methods, which are related to GCD and GBCD, respectively.

2.3.1. Greedy Coordinate Descent

GCD with constant step size $\varepsilon \in (0, 1]$ is an iterative method in which the update steps are performed component-wise as

$$x^{k+1} = x^k - \varepsilon \nabla_{i_k} f\left(x^k\right) e_{i_k}, \tag{5}$$

where e_{i_k} is the unit vector corresponding to the variable $i_k \in \{1, \ldots, d\}$ selected to be updated at step k. A common selection strategy is to use the Gauss-Southwell (GS) selection rule

$$i_k = \underset{i}{\operatorname{argmax}} |\nabla_i f(x^k)|. \tag{6}$$

A more elaborated update routine, the so called Gauss-Southwell-Lipschitz (GSL) rule, was proposed by Nutini et al. (2015) and is given by

$$i_k = \underset{i}{\operatorname{argmax}} \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}}. \tag{7}$$

The GSL rule not only takes the gradient into account but also the curvature along each component by scaling with respect to the Lipschitz constant L_i of the i-th component. This second-order information in the GSL rule can be incorporated into the update step (5) as well, yielding

$$x^{k+1} = x^k - \varepsilon \frac{1}{L_{i_k}} \nabla_{i_k} f\left(x^k\right) e_{i_k}. \tag{8}$$

2.3.2. Greedy Block Coordinate Descent

GBCD works similarly to GCD but considers blocks of variables instead of single variables to be updated in each step. Thus, the d variables are partitioned into disjoint blocks, where each block is indexed by $b \in \mathcal{B}$. Overall, one obtains a total number of $|\mathcal{B}|$ blocks, where $|\cdot|$ denotes the carnality. Note that by considering a single coordinate per block, we recover GCD, which is why the latter can be seen as a special instance of GBCD. The block-wise updates in GBCD are of the form $x^{k+1} = x^k + \varepsilon U_{b_k} \upsilon_{b_k}$, with step size $\varepsilon \in (0, 1]$ and U_{b_k} a block-wise matrix with an identity matrix block for the selected block b_k at the current step k and else zeros. With υ_{b_k} we denote the direction in which the selected block will be updated. This direction can be chosen to correspond to the block-wise steepest descent direction and thus to the negative gradient with respect to the variables of the selected block $\upsilon_{b_k} = -\nabla_{b_k} f\left(x^k\right)$, or can be extended to a matrix update by scaling with matrix H_{b_k} to yield

$$\upsilon_{b_k} = -(H_{b_k})^{-1} \nabla_{b_k} f\left(x^k\right), \tag{9}$$

where H_{b_k} could correspond to the respective block of the Hessian or an upper bound of the latter (Nutini et al., 2022). Analogously to the block update in (9), a greedy block selection rule called Gauss-Southwell-Quadratic (GSQ) was defined by Nutini et al. (2022)

$$b_k \quad = \quad \underset{b \in \mathcal{B}}{\operatorname{argmax}} \quad \{\|\nabla_b f(x^k)\|_{H_b^{-1}}\}, \tag{10}$$

where, $\|\cdot\|_H = \sqrt{\langle H\cdot, \cdot\rangle}$ denotes a general quadratic norm (a proper norm as long as H is a positive definite matrix).

3. Convergence Analysis

Given that the two previously defined conditions of L-smoothness (2.1) and μ-PL (2.2) are fulfilled, convergence results for GCD and GBCD with greedy selection rules and updates as defined in Section 2.3 can be derived (Nutini et al., 2015; Karimi et al., 2016; Nutini et al., 2022). To transfer these results and derive convergence results for different gradient boosting variants, we first check whether Definition 2.1 and Definition 2.2 are fulfilled for the respective problems and then relate the specific boosting methods to their G(B)CD counterpart. We begin by examining the convergence of L_2-Boosting for linear models.

3.1. Convergence Results of L_2-Boosting for Linear Models

A popular variant of gradient boosting is L_2-Boosting (Bühlmann & Yu, 2003), which uses the L_2 loss as loss function. When considering (component-wise) linear models as base learners, this boils down to iteratively solving the least squares (LS) problem

$$\min_{x \in \mathbb{R}^d} f(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} \|y - Ax\|^2, \tag{11}$$

where $y \in \mathbb{R}^n$ denotes the outcome variables and A the $n \times d$ feature matrix with n observations and d predictor variables. Noting that the Hessian of the problem is $\nabla^2 f(x) = A^\top A$, one can deduce that (11) is L-smooth with some Lipschitz constant L_{LS} as we can upper bound the eigenvalues of the Hessian by $L_{LS} \leq \lambda_{max}(A^\top A)$, where $\lambda_{max}(A^\top A)$ denotes the largest eigenvalue of the Hessian. Further, the LS problem is also μ-PL with $\mu_{LS} = \lambda_{pmin}(A^\top A)$, where $\lambda_{pmin}(\cdot)$ denotes the smallest non-zero eigenvalue, as shown by Karimi et al. (2016). More generally, it was discovered by Karimi et al. (2016) and Freund et al. (2017) that any quadratic convex problem of the form

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{2} x^\top Q x + q_1^\top x + q_0, \tag{12}$$

with a symmetric positive semi-definite (p.s.d.) matrix Q, is μ-PL with $\mu = \lambda_{pmin}(Q)$. As (11) can be written in the form of (12) with $Q = A^\top A$, one obtains the previously mentioned PL constant μ_{LS} of the LS problem as a special case. Noting that all eigenvalues of a symmetric positive definite (p.d.) matrix are positive, we get that $\mu = \lambda_{min}(Q) > 0$ for p.d. Q. In this case, we even recover the stronger condition of strong convexity. This indeed makes a difference when considering the LS problem (11) in the overdetermined $n > d$-setting ($A^\top A$ p.d.) compared to the underdetermined setting with $n < d$ ($A^\top A$ only p.s.d.) as frequently encountered in high-dimensional statistics.

3.1.1. Greedy Coordinate Descent for L_2 Loss and Component-wise L_2-Boosting

In the coordinate-wise setting, instead of the stronger condition of L-smoothness (2), we only require the LS problem to obtain a component-wise L-Lipschitz continuous gradient ∇f with Lipschitz constant L_{CLS}, such that for all $\forall x \in \mathbb{R}^d, \beta \in \mathbb{R}$, it holds $\forall j \in \{1, \ldots, d\}$

$$|\nabla_j f(x + \beta e_j) - \nabla_j f(x)| \leq L_{CLS} |\beta|. \tag{13}$$

Thus, (13) immediately follows with $L_{CLS} \leq L_{LS}$. As the LS problem is μ-PL with μ_{LS}, we can derive a global linear convergence rate following Karimi et al. (2016). For simplicity, we assume standardized predictors, making our derivation independent of whether we use the GS or GSL

scheme. Given a constant step size $\varepsilon \in (0, 1]$, we obtain the following convergence result:

$$f(x^k) - f^* \leq \left(1 - \varepsilon(2-\varepsilon)\frac{\mu_{LS}}{d}\right)^k \left(f(x^0) - f^*\right). \quad (14)$$

The derivation of (14) is given in Section 3.2 and in line with results by Freund et al. (2017). It shows the linear convergence of GCD to the optimal function value and hence in the parameter space the convergence to an ordinary least squares (OLS) solution. The latter is unique as long as $A^\top A$ is p.d. and thus in case the problem is strongly convex. However, even in the underdetermined case of $d > n$, the procedure shows linear convergence towards the nearest point of the solution set.

To transfer the convergence result in (14) to boosting for linear models we need to show the equivalence of the boosting method to GCD. In contrast to previous results from Freund et al. (2017), we derive a more general result that particularly shows the equivalence to GCD with GSL selection and update rule, which also holds beyond standardized predictors.

GSL Rule. The GSL rule in component-wise L_2-Boosting for linear models can be recovered by examining the greedy selection of components at iteration k (given the current residuals u^k) in the boosting method:

$$\begin{aligned} \hat{\jmath}_k &= \underset{1\leq j\leq d}{\arg\min} (u^k - A_j \hat{x}_j)^2 = \underset{1\leq j\leq d}{\arg\min} \; - \frac{({u^k}^\top A_j)^2}{A_j^\top A_j} \\ &= \underset{1\leq j\leq d}{\arg\max} \quad \frac{(\nabla_j f(x^k))^2}{L_j} \quad \text{(GSL rule).} \end{aligned}$$

Here, A_j corresponds to the column of A belonging to the single predictor with index j. After plugging in the OLS estimate for $\hat{x}_j$, we obtain the result by using the gradient and Lipschitz constant of the LS problem along the component j. Apart from recovering the GSL rule (7), we further notice that the update step for this L_2-Boosting variant is

$$x^{k+1} = x^k + \varepsilon e_{\hat{\jmath}_k} \hat{x}_{\hat{\jmath}_k}, \quad (15)$$

with step size $\varepsilon \in (0, 1]$ and

$$\hat{x}_{\hat{\jmath}_k} = \frac{{u^k}^\top A_{\hat{\jmath}_k}}{A_{\hat{\jmath}_k}^\top A_{\hat{\jmath}_k}} = -\frac{\nabla_{\hat{\jmath}_k} f(x^k)}{L_{\hat{\jmath}_k}}. \quad (16)$$

Thus, the update is identical to the GSL update step as defined in (8). Therefore, we have established the equivalence of component-wise L_2-Boosting for linear models and GCD with GSL-type selection and updates. For the latter, Nutini et al. (2015) showed that the convergence is at least as fast as the one of GCD with GS-type selection and updates. Thus, component-wise boosting inherits this comparably faster convergence.

3.1.2. Greedy Block Coordinate Descent and Block-wise L_2-Boosting

We now extend the previous analysis by considering blocks of coordinates instead of single coordinate updates. In general, block-wise structures often occur naturally in modeling applications, e.g., when considering grouped variables that range over several columns of the feature matrix A. Classical examples are dummy-encoded categorical variables or regression splines as we will see in Section 3.3.2.

Considering block-wise L_2-Boosting for linear models, the block $\hat{b}_k$ to be updated at step k, is greedily chosen by

$$\begin{aligned} \hat{b}_k &= \underset{b\in\mathcal{B}}{\arg\min} \left(u^k - A_b \hat{x}_b\right)^2 \\ &= \underset{b\in\mathcal{B}}{\arg\min} \quad - {u^k}^\top A_b \left(A_b^\top A_b\right)^{-1} A_b^\top u^k \\ &= \underset{b\in\mathcal{B}}{\arg\max} \quad \sqrt{(\nabla_b f(x^k))^\top \left(H_b\right)^{-1} \nabla_b f(x^k)} \\ &= \underset{b\in\mathcal{B}}{\arg\max} \quad \|\nabla_b f(x^k)\|_{H_b^{-1}} \quad \text{(GSQ rule).} \end{aligned} \quad (17)$$

As for the component-wise selection, we plugged in the OLS estimate for $\hat{x}_b$ that minimizes the LS problem for each block. Using again our knowledge about the block-wise gradient and Hessian in case of the L_2 loss, we can recover the GSQ selection rule. Similar to (15) and (16), it is also easy to show that the update step in the block-wise setting corresponds to the GSQ related update (9) as well.

3.2. General Convergence Results for Block-wise Coordinate Descent with GSQ Rule

After having established the equivalence between block-wise L_2-Boosting for linear models and GBCD with GSQ rule, we will now derive a general convergence result, including a convergence rate for these. To this end, we adopt some of the techniques used by Nutini et al. (2022), who showed convergence for the GSQ rule but without deriving an explicit rate. Dealing with matrix updates and not just single coordinates, we state the L-smoothness condition in the form of

$$\|\nabla_b f(x + U_b v) - \nabla_b f(x)\|_{H_b^{-1}} \leq \|v\|_{H_b}, \quad (18)$$

where $x \in \mathbb{R}^d$, $H_b \in \mathbb{R}^{|b|\times|b|}$ and $v \in \mathbb{R}^{|b|}$. When considering twice-differentiable functions for f, (18) essentially states that H_b must provide an upper bound with respect to the block of the Hessian belonging to the coordinates of block b, i.e., $\nabla^2_{bb} f(x) \preceq H_b$. Note that (18) contains the usual L-smoothness condition as described in (2) as a special case by choosing $H_b = L_b I$, with L_b being the Lipschitz constant for block b (Nutini et al., 2022). This alternative formulation allows us to derive the following upper bound, which is tighter than the one we would get

when considering the formulation via L_b in the block-wise version of (2):

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \langle \nabla_{b_k} f(x^k), x^{k+1} - x^k \rangle \\ &\quad + \frac{1}{2} \|x^{k+1} - x^k\|^2_{H_{b_k}} \\ &= f(x^k) + \varepsilon(1 - \frac{\varepsilon}{2}) \|\nabla_{b_k} f(x^k)\|^2_{H^{-1}_{b_k}} . \end{aligned} \tag{19}$$

The first inequality follows from the L-smoothness in (18), whereas for the second equality, we used the GSQ related update (16) for x^{k+1}. The inequality in (19) defines the so called progress bound. For quadratic objective functions f, one can obtain a tight progress bound by choosing $H_b = \nabla^2_{bb} f(x)$. This makes the GSQ rule optimal when considering quadratic objectives. The optimality also transfers to the GSL rule in the component-wise setup. Thus L_2-Boosting makes use of optimal greedy selection and updates by using the GSQ and GSL rule in the block- and component-wise setting, respectively.

Continuing our proof of convergence, we next introduce a norm that particularly matches the GSQ rule, defined by

$$\|\vartheta\|_{\mathcal{B}} = \max_{b \in \mathcal{B}} \|\vartheta_b\|_{H_b^{-1}} \tag{20}$$

for some $\vartheta \in \mathbb{R}^d, \vartheta_b \in \mathbb{R}^{|b|}$, and some p.d. $H_b \in \mathbb{R}^{|b| \times |b|}$. This norm is a block-wise generalization of the L_∞-norm, which can be seen by choosing $H_b = I_b$ and a block for each coordinate. The close connection to GSQ updates and selection was also noted by Nutini et al. (2022), as we have $\|\nabla f(x^k)\|_{\mathcal{B}} = \|\nabla_{b_k} f(x^k)\|_{H^{-1}_{b_k}}$. Thus we can write the progress bound in (19) as

$$f(x^{k+1}) \leq f(x^k) + \varepsilon(1 - \frac{\varepsilon}{2}) \|\nabla f(x^k)\|_{\mathcal{B}}. \tag{21}$$

As a second ingredient, we use the PL-inequality from Definition 2.2, however using the norm in (20) and not the L_2-norm. To relate the two norms, we use that $\|\vartheta\|_2^2 \leq L_{\mathcal{B}} |\mathcal{B}| \|\vartheta\|^2_{\mathcal{B}}$ with $L_{\mathcal{B}} = \max_{b \in \mathcal{B}} \lambda_{max}(H_b)$ (see Appendix B for a proof). Thus, we obtain the PL-inequality

$$\frac{1}{2} \|\nabla f(x)\|^2_{\mathcal{B}} \geq \frac{\mu}{L_{\mathcal{B}} |\mathcal{B}|} (f(x) - f^*), \tag{22}$$

where μ corresponds to the PL parameter of (4) using the L_2-norm. Lastly, by connecting the inequalities (19) and (22), and iterating over k iterations, we get our final convergence result

$$f(x^k) - f^* \leq \left(1 - \varepsilon(2 - \varepsilon) \frac{\mu}{L_{\mathcal{B}} |\mathcal{B}|}\right)^k \left(f(x^0) - f^*\right). \tag{23}$$

For quadratic functions in particular, $\mu = \lambda_{pmin}(Q)$ and we can use that $L_{\mathcal{B}} \leq \lambda_{max}(Q)$ to get

$$f(x^k) - f^* \leq \underbrace{\left(1 - \varepsilon(2 - \varepsilon) \frac{1}{|\mathcal{B}|} \frac{\lambda_{pmin}(Q)}{\lambda_{max}(Q)}\right)}_{=:\gamma}^k \left(f(x^0) - f^*\right). \tag{24}$$

We can obtain several insights related to the convergence speed by looking at the terms determining the convergence rate γ. First, γ is monotonically decreasing in the step size $\varepsilon \in (0, 1]$ and increasing in the number of blocks $|\mathcal{B}|$. Looking at the terms in γ, it is evident that $\gamma \in [0, 1)$, so progress is guaranteed in each step. Due to the optimality of the GSQ rule for quadratic problems and the related tightness of the progress bound (19), we get optimal convergence with a step size of $\varepsilon = 1$ and updating all coordinates simultaneously, thereby recovering the classic Newton's method. Further, the last term in γ can be identified as the reciprocal of the condition number of the Hessian Q. This is to no surprise, as it is a well-known result that gradient methods converge faster for well-posed problems with condition numbers close to one (Boyd & Vandenberghe, 2004). Lastly, one can note that the result in (14) is just a special case of (23) when considering the component-wise case with $|\mathcal{B}| = p$ and standardized predictors with $L_{\mathcal{B}} = L_{CLS} = 1$.

3.3. Convergence Results for Penalized L_2 Loss

After having analyzed the convergence of G(B)CD and component/block-wise gradient boosting for the L_2 loss, we will now consider penalized L_2 losses by examining ridge regression and regression splines.

3.3.1. Ridge Regression

The loss of the ridge-penalized linear model is defined by

$$\min_{x \in \mathbb{R}^d} f_R(x) = \min_{x \in \mathbb{R}^d} \frac{1}{2} \|y - Ax\|^2 + \frac{\alpha}{2} \|x\|^2, \tag{25}$$

where $\alpha > 0$ is a penalization parameter that determines the degree of the coefficient shrinkage. The problem can be written in the form of the quadratic problem (12) with Hessian $Q_R = A^\top A + \alpha I$. A well-known property of the ridge regression is that even in the underdetermined setting with $d > n$ and thus $A^\top A$ only being p.s.d., Q_R will still be p.d. and thus (25) is strongly convex with $\mu = \lambda_{min}(Q_R)$. Being a quadratic and μ-PL problem, we know from Section 3.2 that greedy component- and block-wise CD with GSQ selection and updates will converge to the unique minimizer of (25), with rate stated in (24). The unique solution is the ridge estimator $x^*_R = \left(A^\top A + \alpha I\right)^{-1} A^\top y$.

3.3.2. Regression Splines

We now consider the problem of regression splines – a commonly used non-linear modeling approach due to its ability

to fit flexible functional relationships between the dependent and independent variables. As an example for regression splines, we use P-Splines (Eilers & Marx, 1996) which use B-Splines (de Boor, 1978) as basis expansion of the predictor variables and penalize (higher-order) differences between adjacent regression coefficients of the B-Spline. Further details about P-Splines are given in Appendix E. For our purpose, it is sufficient to recognize that the regression spline problem is given by

$$\min_{x\in\mathbb{R}^d} f_P(x) = \min_{x\in\mathbb{R}^d} \frac{1}{2}\|y - Zx\|^2 + \frac{\alpha}{2}x^\top Px, \qquad (26)$$

where Z is a matrix containing the evaluated basis functions and P is a penalty matrix. A common choice for P is a second-order-difference penalty matrix (Wahba, 1990; Green & Silverman, 1994). In Z, multiple columns belong to a single predictor variable, each corresponding to a piecewise polynomial in the B-Spline case. For multiple predictor variables, this naturally induces a block-wise structure for x, Z, and P (see Appendix E for details).

As the block structure in (26) requires the blocks to be updated simultaneously, only GBCD but not GCD will be applicable. To show convergence for GBCD for the regression spline problem, we again check whether the problem in (26) fulfills certain characteristics. Similar to ridge regression, we can write (26) as a quadratic problem of the form (12) with $Q_P = Z^\top Z + \alpha P$. However, in contrast to the ridge case, the penalization matrix P is usually rank deficient, and thus αP is only p.s.d. Although $Z^\top Z$ might also only be p.s.d., Q_P can be shown to be p.d. (Eilers et al., 2015, see, e.g.,). For the general convergence result in (24) to apply, it does not matter whether Q_P is p.s.d. or p.d., as this only determines whether the minimizer of (26) is unique. However, to make the progress bound (19) for GBCD with GSQ rule tight, one needs to choose $H^{PLS} := Q_P$. Thus, for the matrix inversion in (10) and (9), the parts of the Hessian belonging to block b, $H_b^{PLS} = \nabla^2_{bb} f_P(x)$, need to be of full rank. As many regression spline problems fulfill the requirements for the convergence result in Section 3.2, we get convergence of GBCD with GSQ selection and updates with the rate stated in (24) when applied to these problems.

3.3.3. Component-/Block-wise Boosting

Next, we examine the convergence of boosting using component-wise regression splines, again using the example of P-spline boosting (Schmid & Hothorn, 2008), and the convergence of boosting ridge regression (Tutz & Binder, 2007). Results for the latter can directly be established from boosting with regression splines by defining the blocks of basis functions as blocks in the original data, i.e., $Z_b = A_b$, and setting the respective penalty matrix blocks to $P_b = I_b$.

A key insight is that compared to G(B)CD, these boosting algorithms neglect the penalization accumulated in previous boosting iterations. This can be seen by reformulating the selection and update at step k of boosting using component-wise regression splines as

$$\begin{aligned}
\hat{b}_k &= \arg\min_{b\in\mathcal{B}} \left(u^k - Z_b\hat{x}_b\right)^2 + \alpha\hat{x}_b^\top P_b\hat{x}_b \\
&= \arg\min_{b\in\mathcal{B}} - {u^k}^\top Z_b \left(Z_b^\top Z_b + \alpha P_b\right)^{-1} Z_b^\top u^k \\
&= \arg\max_{b\in\mathcal{B}} \|\nabla_b f_{LS}(x^k)\|_{(H_b^{PLS})^{-1}} \quad \text{(GSQ rule)}.
\end{aligned} \qquad (27)$$

In the second line, we plugged in the P-Spline estimator for $\hat{x}_b$ and subsequently simplified terms. Importantly, in the fourth line, we recover the gradient of the unpenalized LS problem $\nabla_b f_{LS}$ as the algorithm does not account for penalization from previous steps. The gradients of the penalized and unpenalized problem are only equivalent in the first iteration when no component is penalized yet. Thus, the procedure still applies a GSQ rule, however, not using the Hessian of the unpenalized but the penalized problem. The same can be observed for the update step which is $x^{k+1} = x^k + \varepsilon U_{\hat{b}_k}\hat{x}_{\hat{b}_k}$ with

$$\hat{x}_{\hat{b}_k} = -\left(H_{\hat{b}_k}^{PLS}\right)^{-1} \nabla_{\hat{b}_k} f^{LS}(x^k). \qquad (28)$$

Note, that (27) and (28) scale the gradient with H_b^{PLS}, which is a block from the Hessian of the penalized problem. Hence, (27) is not equivalent to the update step that we have seen for L_2-Boosting (17). Nonetheless, as H_b^{PLS} still provides an upper bound to the Hessian of the unpenalized problem ($Z_b^\top Z_b \preceq Z_b^\top Z_b + \alpha P_b$ as αP_b p.s.d. $\forall b \in \mathcal{B}$), we still apply a GSQ selection and update. Thus, the condition in (18) is still fulfilled even though the progress bound in (19) is not tight. Thus, boosting using regression splines and boosting ridge regression are simply G(B)CD routines with GSQ rule related to the unpenalized problems. Choosing H_b not to correspond to the Hessian of the unpenalized problem (progress bound not tight), makes the selection and updates no longer optimal. However, as the unpenalized problems are quadratic and μ-PL, convergence from (24) still applies. Hence, boosting using regression splines and boosting ridge regression converge towards the unpenalized fit. For the case of boosting ridge regression, this is therefore not the ridge solution x_R^* but the OLS solution. Our previous result has unexpected practical implications for commonly used boosting implementations that fit (large-scale) penalized spline problems (e.g., Hothorn et al., 2010; Brockhaus et al., 2020). This is demonstrated in Fig. 1 and 2, and further discussed in Section 4.

3.4. Other Loss Functions

Exponential family losses. Boosting as defined in Algorithm in Appendix A can also incorporate exponential family loss functions (Bühlmann & Hothorn, 2007). This includes

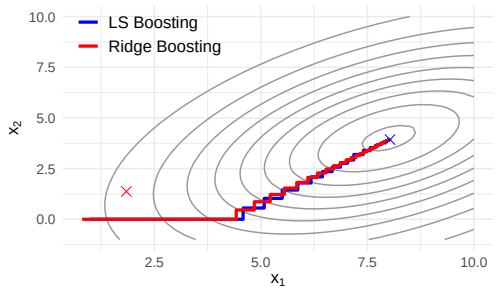

Figure 1. Coefficient paths for LS (blue line) and Ridge (red line) boosting as well as their actual solutions (crosses in resp. colors).

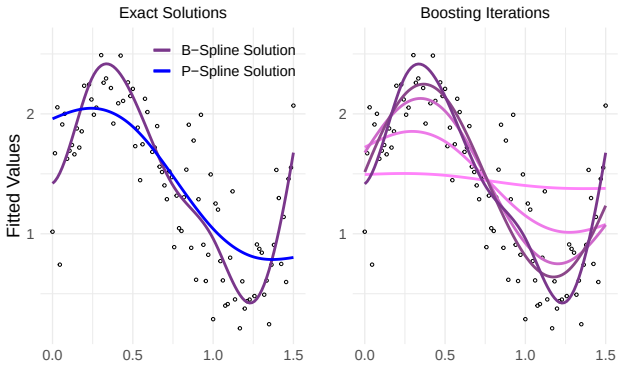

Figure 2. Left: Exact B-Spline (purple) and P-Spline solution (blue). Right: P-Spline boosting iterates converging to the unpenalized (B-Spline) solution (darkest color).

L_2-Boosting, as well as boosting for count (Poisson) or binary (Bernoulli) data. Although these loss functions are not necessarily strongly convex, they are strictly convex in case the canonical link is chosen (Appendix D.1). As strictly convex functions are μ-PL over any compact set (Karimi et al., 2016), the problem qualifies for the convergence result of GBCD in Section 3.2. For generalized linear models where the distribution of the response belongs to the exponential family class, a common approach is to use boosting with component-wise LS base procedure to fit the model (Bühlmann & Hothorn, 2007). Remarkably, we can show that even with exponential family loss, this procedure can be interpreted as GBCD with GSQ rule (Appendix D.2). Due to the LS base procedure, one uses the upper bound $A_b^\top A_b$, instead of the actual Hessian $A_b^\top W_b A_b$. In the case of the logistic loss (BinomialBoosting; Bühlmann & Hothorn, 2007), we have that $W_b \preceq I_b$, so that boosting uses a valid upper bound. In case of the Poisson loss (PoissonBoosting), this upper-boundedness cannot be guaranteed in general, making a sufficiently small step size ε necessary to provide a valid global upper bound of the Hessian. Hence, as long as a valid upper bound is used, we get guaranteed convergence of these boosting methods by the convergence result in Section 3.2.

Robust losses. Robust loss functions such as the L_1 and Huber loss are frequently used alternatives to the L_2 loss for gradient boosting methods (Bühlmann & Hothorn, 2007). While the idea of greedy selection and updates would still be applicable in this case, we cannot obtain convergence results as stated in (23). The reason for this is that the gradient of the L_1-loss does not decrease in size as we reach the minimizer of the problem, which is why it cannot be μ-PL. The Huber loss is μ-PL only in a δ neighborhood of the minimum. Thus, the convergence rate for the Huber loss cannot be global linear as in (23) but only hold locally.

3.5. Optimization for Regression and Smoothing Splines

Cubic smoothing splines. L_2-Boosting with cubic smoothing splines (CSS) was proposed in the seminal work of Bühlmann & Yu (2003). As CSS penalize the second differences of the fitted function, this can be seen as a continuous generalization of boosting with regression spline using second-order difference matrix. While Bühlmann & Yu (2003) considered component-wise CSS in their experiments, they did not derive convergence results for the component-wise case. Using the eigenvalue properties of CSS, we show convergence to the fully saturated model for component-wise L_2-Boosting with multiple CSS (Appendix C). Interestingly, this result holds independently of the selection rule.

Regression splines. Backfitting (Friedman & Stuetzle, 1981) is one of the most common methods to fit models with regression splines or additive models. It is a Gauss-Seidel iterative method that cyclically proceeds through each component of the additive model, fitting each component while holding the others fixed at their current estimate. The key difference to the GBCD is that updates are cyclic instead of greedy, and estimates for each component are not incrementally added up, but estimated independently from previous estimates of the respective component. The convergence and existence of solutions for backfitting has been analyzed by Buja et al. (1989) and Ansley & Kohn (1994). They showed that backfitting converges, even in the "degenerated" case, where multiple solutions exist. However, no explicit convergence rate has been derived so far. More recently, Beck & Tetruashvili (2013) derived convergence rates for related methods with cyclic updates. However, to achieve a linear rate of convergence, their analysis either assumes strong convexity or a maximum of two blocks.

4. Numerical Experiments

In the following, we numerically demonstrate our theoretical findings on convergence rates and derived conclusions.

Convergence rates. We start by empirically exploring the impact of different factors on our derived convergence rates. In Figure 3, we investigate the influence of different degrees

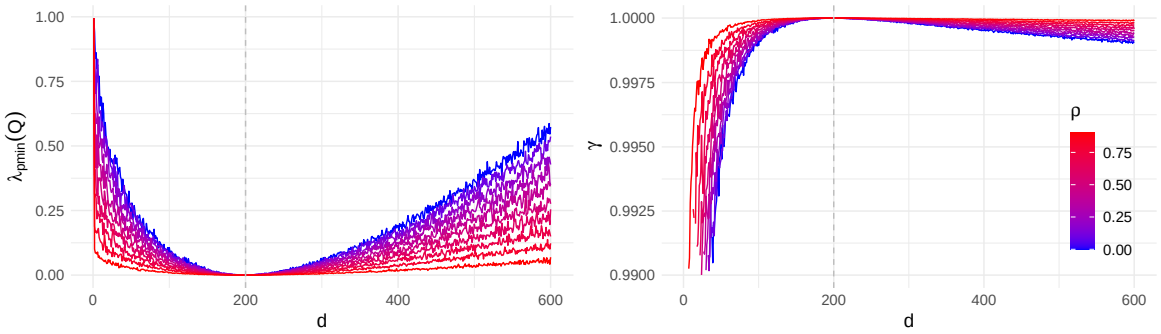

Figure 3. Smallest non-zero eigenvalue of Q (left) and convergence rate γ as given in (14) (right) for component-wise boosting/GCD for a linear model with varying pairwise correlation ρ between predictor variables (color) for fixed $n = 200$, $\varepsilon = 1$, and varying d (x-axis).

of pairwise correlations between predictors on the convergence rate γ and the μ-PL constant ($\lambda_{pmin}(Q)$) for a linear model with an increasing number of predictors. Generally, the convergence rate is close to one, indicating relatively slow convergence behavior. An observation that matches the general notion of slow (over)fitting behavior of boosting (Bühlmann & Hothorn, 2007). Convergence is generally faster (small γ values) for low pairwise correlation. For a growing number of predictors, γ steeply increases to a rate close to one until $d = n$. In the underdetermined case ($n < d$), the rate slowly decreases again while remaining close to one. This phenomenon is due to the smallest eigenvalue decreasing with the number of predictors while remaining above zero for full-rank tall matrices ($n > d$). By contrast, for flat matrices ($n < d$), more and more eigenvalues become zero, thus the smallest non-zero eigenvalue grows, resulting in an increase in convergence speed. We further investigated the impact of different condition numbers on the linear convergence rate in Figure 4 (in Appendix F). In line with the theoretical results obtained in Section 3.2, the convergence is slower for higher condition numbers of the problem.

Boosting paths and solutions. Another implication of our analysis is the solution commonly used boosting algorithms are converging to. As discussed in the previous section, this corresponds to the unpenalized solution for linear, ridge, and regression spline boosting. Further, several implementations consider the unpenalized instead of the penalized LS criterion even for the selection of penalized base learners, which can induce differences in the selection compared to (27). While this can lead to different boosting paths, the fact that these implementations use the gradients of the unpenalized problem still implies convergence to the unpenalized fit. We now demonstrate this empirically. In our first experiment, we perform least-squares (LS) and Ridge boosting on a two-dimensional problem $x \in \mathbb{R}^2$ to check whether both are converging against the same solution. As depicted in Figure 1, this is in fact the case, despite the exact Ridge solution (without boosting) being far away from the LS solution. As a second example, we fit a univariate spline problem using unpenalized and penalized B-Splines (B-/P-splines, resp.) using boosting. Analogous to the result of Ridge boosting, we can observe (Figure 2) that boosting converges to the unpenalized B-spline despite iteratively solving a penalized LS criterion. Interestingly, due to the stage-wise fitting nature of boosting, the coefficient and function path do not have to visit the penalized fit. Thus, stopping boosting iterations early does not guarantee that we can recover the penalized fit. As a consequence, the common notion of boosting performing an implicit smoothing parameter selection might be flawed and the early stopped model might belong to a completely different function class. This, in turn, can have detrimental consequences for the statistical inference, likely providing false uncertainty statements.

5. Discussion

In this paper, we derived novel convergence rates for various boosting variants, including linear, ridge, and regression splines. A key achievement is establishing a global linear convergence rate for the greedy block coordinate descent procedure under broad assumptions. This insight not only deepens the understanding of boosting principles but also provides the first explicit convergence rate for optimization in regression spline problems.

Future research. The rate of convergence in (23) shows that the speed of convergence is monotonically decreasing in the step size. However, choosing a small step size might lead to greater model exploration over the boosting iterations, thus enabling better generalization performance when coupled with early stopping. This resembles a common notion in boosting research, that the procedure usually exhibits better out-of-sample performance for incremental step sizes (Friedman, 2001). While fast convergence rates are vital in optimization-related research, it might not be of primary interest from a statistical point of view (Tibshirani, 2015). A logical next step is thus to analyze the effect of early stopping and examine whether updates can be structured in a way that the penalized solution is in fact included in the boosting path.

Broader Impact

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ansley, C. F. and Kohn, R. Convergence of the backfitting algorithm for additive models. *Journal of the Australian Mathematical Society*, 57(3):316–329, 1994.

Beck, A. and Tetruashvili, L. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013.

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

Breiman, L. Arcing classifiers. *The Annals of Statistics*, 26 (3):801–824, 1998.

Breiman, L. Prediction games and arcing algorithms. *Neural Comput.*, 11(7):1493–1517, oct 1999.

Brockhaus, S., Rügamer, D., and Greven, S. Boosting functional regression models with fdboost. *Journal of Statistical Software*, 94(10):1–50, 2020.

Bühlmann, P. and Hothorn, T. Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science*, 22(4):477 – 505, 2007.

Buja, A., Hastie, T., and Tibshirani, R. Linear smoothers and additive models. *The Annals of Statistics*, pp. 453–510, 1989.

Bühlmann, P. and Yu, B. Boosting with the l2 loss: Regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

Bühlmann, P., Gertheiss, J., Hieke, S., Kneib, T., Ma, S., Schumacher, M., Tutz, G., Wang, C.-Y., Wang, Z., and Ziegler, A. Discussion of "the evolution of boosting algorithms" and "extending statistical boosting". *Methods of information in medicine*, 53:436–445, 11 2014.

Collins, M., Schapire, R. E., and Singer, Y. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48:253–285, 2002.

de Boor, C. *A Practical Guide to Spline*, volume 27. Springer, 01 1978.

Eilers, P. H., Marx, B. D., and Durbán, M. Twenty years of p-splines. *SORT: statistics and operations research transactions*, 39(2):0149–186, 2015.

Eilers, P. H. C. and Marx, B. D. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89 – 121, 1996.

Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. *Regression: Models, Methods and Applications*. Springer, 01 2013.

Freund, R. M., Grigas, P., and Mazumder, R. A new perspective on boosting in linear regression via subgradient optimization and relatives. *The Annals of Statistics*, 45 (6):2328–2364, 2017.

Freund, Y. and Schapire, R. E. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, pp. 148–156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Friedman, J., Hastie, T., and Tibshirani, R. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337 – 407, 2000.

Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Friedman, J. H. and Stuetzle, W. Projection pursuit regression. *Journal of the American statistical Association*, 76 (376):817–823, 1981.

Green, P. J. and Silverman, B. W. *Nonparametric regression and generalized linear models*. Chapman & Hall/CRC Press, 1994.

Hofner, B., Mayr, A., and Schmid, M. gamboostLSS: An R Package for Model Building and Variable Selection in the GAMLSS Framework. *Journal of Statistical Software*, 74(1):1–31, 2016.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., and Hofner, B. Model-based boosting 2.0. *Journal of Machine Learning Research*, 11(71):2109–2113, 2010.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pp. 795–811. Springer, 2016.

Locatello, F., Raj, A., Karimireddy, S. P., Rätsch, G., Schölkopf, B., Stich, S., and Jaggi, M. On matching pursuit and coordinate descent. In *International Conference on Machine Learning*, pp. 3198–3207. PMLR, 2018.

Mason, L., Baxter, J., Bartlett, P., and Frean, M. Boosting algorithms as gradient descent. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

Meir, R. and Rätsch, G. *An Introduction to Boosting and Leveraging*, volume 2600, pp. 119–184. Springer, 01 2003.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Nutini, J., Schmidt, M., Laradji, I., Friedlander, M., and Koepke, H. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pp. 1632–1641. PMLR, 2015.

Nutini, J., Laradji, I., and Schmidt, M. Let's make block coordinate descent converge faster: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *Journal of Machine Learning Research*, 23 (131):1–74, 2022.

Rätsch, G., Mika, S., and Warmuth, M. K. K. On the convergence of leveraging. In Dietterich, T., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

Rügamer, D., Brockhaus, S., Gentsch, K., Scherer, K., and Greven, S. Boosting Factor-Specific Functional Historical Models for the Detection of Synchronization in Bioelectrical Signals. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 67(3):621–642, 2017.

Schmid, M. and Hothorn, T. Boosting additive models using component-wise p-splines. *Computational Statistics & Data Analysis*, 53(2):298–311, 2008.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Tibshirani, R. J. A general framework for fast stagewise algorithms. *Journal of Machine Learning Research*, 16 (78):2543–2588, 2015.

Tutz, G. and Binder, H. Boosting ridge regression. *Computational Statistics & Data Analysis*, 51(12):6044–6059, 2007.

Wahba, G. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, 1990.

Wood, S. N. *Generalized additive models: an introduction with R*. CRC press, 2017.

Zhang, T. and Yu, B. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33(4): 1538–1579, 2005.

A. Background

The following algorithm, as given in Bühlmann & Hothorn (2007), is a generalization of the original version from Friedman (2001). Besides being compatible with different loss functions, Appendix A gains its generality by incorporating a generic base procedure. The latter can include various base learners as well as block- and component-wise selection and updates, thereby resembling a wide variety of gradient boosting variants. Further, note that the algorithm replaces a line search as proposed in the original gradient boosting version (Friedman, 2001) with updates of constant step size in step 4. The latter was also discussed in Friedman (2001) and is usually implemented in software applications due to its computational advantages. Below, we use the notation $\widetilde{y}_i$ to denote the functional derivative, also known as the Gâteaux derivative, of the loss function at the current function estimate. Notice, that this is only defined at the data points a_i with $i \in \{1, \dots, n\}$, where a_i denotes the i-th row of the $n \times d$ feature matrix A. The updates $g^k(\,\cdot\,; \hat{x})$ of the function estimate f^k are parameterized by a parameter x. This is to highlight the fact that gradient boosting can not only be seen as a method performing steps in function but also in the parameter space.

Algorithm 1 Gradient Boosting (Bühlmann & Hothorn, 2007)

1: Initialize $f^0(A)$. Set $k = 0$.
2: Given loss function $\ell(\cdot)$, compute the functional derivative evaluated at the current estimate f^k, i.e.,

$$\widetilde{y}_i := -\frac{\partial}{\partial f}\ell(y_i, f)\Big|_{f=f^k(a_i)} \quad i = 1, \dots, n \tag{29}$$

Set k = k+1.
3: Fit base procedure to the negative gradient $\{\widetilde{y}_i\}_{i=1}^n$:

$$\{\widetilde{y}_i, a_i\}_{i=1}^n \xrightarrow{\text{base procedure}} g^k(\,\cdot\,; \hat{x}). \tag{30}$$

4: Update $f^k(\cdot) = f^{k-1}(\cdot) + \varepsilon g^k(\,\cdot\,; \hat{x})$ with $\varepsilon \in (0, 1]$
5: Repeat steps 2 – 4 until convergence or until a specified stopping criterion is met.

B. Proof: Norm upper bound

Lemma B.1. *Define the following norm* $\|\vartheta\|_{\mathcal{B}} =: \max_{b\in\mathcal{B}} \|\vartheta_b\|_{H_b^{-1}}$ *with* $\vartheta_b \in \mathbb{R}^{|b|}$ *and* $H_b \in \mathbb{R}^{|b|x|b|}$ *for* $b \in \mathcal{B}$*. The vector* ϑ *denotes the stacked vector of all* ϑ_b *for* $b \in \mathcal{B}$*. Using* $L_{\mathcal{B}} =: \max_{b\in\mathcal{B}} \lambda_{max}(H_b)$*, the* L_2 *norm can be upper bounded by this norm as*

$$\|\vartheta\|_2^2 \leq L_{\mathcal{B}}\ |\mathcal{B}| \|\vartheta\|_{\mathcal{B}}^2. \tag{31}$$

Proof.

$$\|\vartheta\|_2^2 = \sum_b^{|\mathcal{B}|} \|\vartheta_b\|_2^2 \leq \sum_b^{|\mathcal{B}|} \lambda_{max}(H_b)\ \vartheta_b^\top H_b^{-1} \vartheta_b \leq L_{\mathcal{B}} \sum_b^{|\mathcal{B}|} \vartheta_b^\top H_b^{-1} \vartheta_b \leq L_{\mathcal{B}}\ |\mathcal{B}| \|\vartheta\|_{\mathcal{B}}^2 \tag{32}$$

□

For quadratic problems, we have seen that the tightest progress bound (19) is achieved by choosing $H_b = Q_b$, where Q corresponds to the Hessian of the quadratic problem. In that case, it holds that $L_{\mathcal{B}} = \max_{b\in\mathcal{B}} \lambda_{max}(Q_b) \leq \lambda_{max}(Q)$.

C. Proof: Boosting with Multiple Cubic Smoothing Splines

PRELIMINARIES

Consider a linear operator T. In the Lemma below, we will consider the case of $T : \mathbb{R}^n \to \mathbb{R}^n$ and $z, c \in \mathbb{R}^n$. For the last property we consider two linear operators $T_1 : \mathbb{R}^n \to \mathbb{R}^n$ and $T_2 : \mathbb{R}^n \to \mathbb{R}^n$.

1. Operator Norm: $\|T\|^* = \sup_{\|z\|=1} \|Tz\| = \sup_{\|z\|\neq 0} \|T\frac{z}{\|z\|}\|$

2. $\|T\|\|c\| = \|c\|\|T\frac{c}{\|c\|}\| \leq \|c\| \sup_{\|c\|\neq 0} \|T\frac{c}{\|c\|}\| = \|c\|\|T\|^*$

3. $\|T_1T_2\|^* = \sup_{\|z\|\neq 0} \frac{\|T_1T_2z\|}{\|z\|} = \sup_{\|z\|\neq 0} (\frac{\|T_1T_2z\|}{\|T_2z\|}\frac{\|T_2z\|}{\|z\|}) \leq \sup_{\|T_2z\|\neq 0} \frac{\|T_1T_2z\|}{\|T_2z\|} \sup_{\|z\|\neq 0} \frac{\|T_2z\|}{\|z\|} = \|T_1\|^*\|T_2\|^*$

Lemma C.1. *L_2-Boosting with d cubic smoothing splines yields the saturated model (perfect fit) for $k \to \infty$.*

Proof. We have d cubic smoothing splines $S^{(m)}$ $(m \in \{1, ..., d\})$ with n eigenvalues $\lambda_i^{(m)}$ $(i \in \{1, ..., n\})$ and $0 < \lambda_i^{(m)} \leq 1$. Define the boosting operator in step k as $T_k := \left(I - S^{(m_k)}\right)$, where m_k denotes the index of the selected cubic smoothing spline learner at step k. The boosting operator $T_k : \mathbb{R}^n \to \mathbb{R}^n$ with $k \in \mathbb{N}$ has n eigenvalues $\tilde{\lambda}_i^{(m)}$ with $0 \leqslant \tilde{\lambda}_i^{(m)} < 1$ as considered by (Bühlmann & Yu, 2003).
Then:

$$\begin{aligned}
\left\|y - f^k\right\| = \left\|u^k\right\| &= \|T_k \cdot \ldots \cdot T_1 y\| \\
&\leq \|T_k \cdot \ldots \cdot T_1\|^* \|y\| && \text{by (2.)} \\
&\leq \|T_k\|^* \cdot \ldots \cdot \|T_1\|^* \|y\| && \text{by (3.)} \\
&\leq \|y\| \underbrace{\prod_{j=1}^{k} \underbrace{q_j^{(m)}}_{\in[0,1)}}_{\overset{k\to\infty}{\to} 0} && \text{with } q_j^{(m)} = \max\{|\tilde{\lambda}_i^{(m)}| : i \in \{1, \ldots, n\}\}
\end{aligned}$$

Hence, we get $\left\|u^k\right\| \overset{k\to\infty}{\to} 0$. □

D. Exponential Family Models

This section establishes the link between boosting with exponential family losses via component/block-wise LS and GBCD with GSQ selection and updates.

D.1. Exponential Family

For any exponential family model, the density of the response can be written in the following form

$$f_\theta(y) = \exp\left[\{y\theta - b(\theta)\}/\phi + c(\phi, y)\right], \tag{33}$$

where the terms θ, $b(\theta)$, ϕ and $c(\phi, y)$ depend on the exponential family distribution considered (Fahrmeir et al., 2013; Wood, 2017). The parameter θ is called the canonical parameter and is often written as $\theta(\mu)$, due to its dependence on the expectation of the response variable, $\mathbb{E}[Y] = \mu$. In this setup, one aims to estimate a function η, given some response function $h(\cdot)$, such that $h(\eta_i) = \mathbb{E}[Y_i] = \mu_i$. For generalized linear models (GLMs) η is linear in the estimated parameter x, as we assume $\eta = Ax$. For each exponential family, there exists a unique canonical link function $g = h^{-1}$, such that $\theta_i = \eta_i$ (Fahrmeir et al., 2013). Choosing the canonical link has several theoretical benefits. First, the log-likelihood $\log(f_\theta(y))$ used to estimate the model, can be written both in terms of θ or η. Thus, the loss function is defined by

$$\ell(\eta) = \ell(\theta) = \{y\theta - b(\theta)\}/\phi + c(\phi, y). \tag{34}$$

Another key feature of the canonical link is that (34) is strictly convex in η. For other link functions this property cannot be guaranteed. We can make this notion more explicit by looking at the Hessian of (34). We do so by considering GLMs, for which (34) becomes a function of the parameter x. Using the canonical link, the Hessian of the log-likelihood simplifies and coincides with the Fisher information matrix (Wood, 2017). The latter (derived, e.g., in Fahrmeir et al., 2013) corresponds to

$$\nabla^2 \ell(x) = A^\top W A \tag{35}$$

with $W = \text{diag}(\ldots, \tilde{w}_i, \ldots)$ and

$$\tilde{w}_i = \frac{(h'(\eta_i))^2}{b''(\theta_i)\phi}. \tag{36}$$

By looking at the definition of the respective terms for different exponential family distributions, it becomes clear that $b''(\theta) > 0$ and $\phi > 0$ (Fahrmeir et al., 2013; Wood, 2017). With the canonical response function being strictly monotonic, the numerator must be greater than zero as well. Therefore, given the canonical link, the weights are positive and the Hessian positive definite. Thus, as long as A is full rank, the log-likelihood is strictly convex.

D.2. Relating Boosting for Exponential Family Models to GBCD with GSQ

As described in Algorithm Appendix A, boosting fits in each iteration a certain base procedure against the negative functional derivative of the loss function $\ell(\cdot)$ at the current function estimate f^k. Note the function estimate f as defined in (29), corresponds to the term η in exponential family notation. In order to relate boosting for exponential families to GBCD, one needs to establish the link between the negative functional derivative $\{\widetilde{y}_i\}_{i=1}^n$ and the gradient of the loss function with respect to the parameter x. Considering the negative log-likelihood instead of the log-likelihood in (34) and using that for GLMs we have $\eta = Ax$, one can do so by

$$-\frac{\partial}{\partial x}\ell(x) = \frac{\partial}{\partial x}\eta(x)\ \widetilde{y} = A^\top \widetilde{y}. \tag{37}$$

The block-wise LS base procedure of boosting for exponential family models can now be written as GBCD with GSQ. We derive this for the block-wise procedure as the component-wise procedure can be recovered as a special case of it. First, the block selection for this kind of boosting method is based on

$$\begin{aligned}
\hat{b}_k &= \arg\min_{b\in\mathcal{B}} \left(\widetilde{y}^k - A_b \hat{x}_b\right)^2 \\
&= \arg\min_{b\in\mathcal{B}} \quad -\widetilde{y}^{k\top} A_b \left(A_b^\top A_b\right)^{-1} A_b^\top \widetilde{y}^k \\
&= \arg\max_{b\in\mathcal{B}} \quad \sqrt{(\nabla_b \ell(x^k))^\top \left(A_b^\top A_b\right)^{-1} \nabla_b \ell(x^k)} \\
&= \arg\max_{b\in\mathcal{B}} \quad \|\nabla_b \ell(x^k)\|_{(A_b^\top A_b)^{-1}} \quad \text{(GSQ rule)},
\end{aligned} \tag{38}$$

which is analogous to the derivation in (17) and corresponds to the GSQ rule. Similarly, the update is

$$\hat{x}^{k+1} = \hat{x}^k + \varepsilon U_{\hat{b}_k} \hat{x}_{\hat{b}_k} \tag{39}$$

with

$$\hat{x}_{\hat{b}_k} = \left(A_b^\top A_b\right)^{-1} A_{\hat{b}_k}^\top \widetilde{y}^k = -\left(A_{\hat{b}_k}^\top A_{\hat{b}_k}\right)^{-1} \nabla_{\hat{b}_k} \ell(x_k). \tag{40}$$

So as long as $A_b^\top A_b$ provides an upper bound to the respective blocks of the Hessian in (35) for all $b \in \mathcal{B}$, the selection and updates correspond to the GSQ rule and we get linear convergence of boosting with component/block-wise LS for exponential family models due to the result from Section 3.2.

E. Cubic Smoothing Splines and P-Splines

Cubic Smoothing Splines. A *cubic smoothing spline* as considered in Section 3.5 is defined for a with a twice differentiable function f, by

$$\sum_{i=1}^{n} (y_i - f(a_i))^2 + \alpha \int \left(\frac{\partial^2 f}{\partial a^2}\right)^2 da, \tag{41}$$

where a_i denotes the i-th row of the $n \times d$ feature matrix A. Cubic smoothing splines aim to find an optimal function that best adjusts to the least squares problem while at the same time penalizing the squared second derivative of the function to induce smoothness on the solution. The trade-off between the two is controlled by the smoothness parameter α. Cubic smoothing splines are considered by (Bühlmann & Yu, 2003) for componentwise L_2-Boosting.

P-Splines. Although cubic smoothing splines are especially interesting from a theoretical point of view, the integral for penalization imposes major computational disadvantages compared to other smooth base learners that could be used. A natural alternative is to consider a discretized version of it. As such, Eilers & Marx (1996) proposed a specific type of penalized regression splines, which uses B-Splines (de Boor, 1978) as basis expansions of the predictor variables and

penalize the higher-order differences between adjacent regression coefficients of the B-Spline. B-Splines, also called basis splines, are a collection of piece-wise polynomials which are connected at specific points, called knots. For notational convenience, we elaborate on the basis expansion of a single variable via B-Splines. The extension to multiple variables is straightforward. Consider $\{B_i^l(\cdot)\}_{i=1}^{\kappa}$ B-Spline of order l and with κ knots. When considering a P-Spline on a certain variable A_j, this will induce an expanded feature matrix Z (Fahrmeir et al., 2013), which is given by

$$Z = \begin{pmatrix} B_1^l(A_{1,j}) & \dots & B_\kappa^l(A_{1,j}) \\ \vdots & & \vdots \\ B_1^l(A_{n,j}) & \dots & B_\kappa^l(A_{n,j}) \end{pmatrix}.$$

With this expanded feature matrix, we can take a look at an approximation of the cubic smoothing spline by a P-Spline as proposed by (Eilers & Marx, 1996):

$$\|y - Zx\|^2 + \alpha x^\top \underbrace{D_2^\top D_2}_{:=P} x, \tag{42}$$

where x now denotes an extended coefficient vector corresponding to each piecewise polynomial in the B-Spline, respectively. The matrix P corresponds to the second-order difference penalty matrix. The $d-2 \times d$ matrix D_2 and $d \times d$ matrix P as given in (Fahrmeir et al., 2013) are defined as follows:

$$D_2 = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix} \qquad P = \begin{pmatrix} 1 & -2 & 1 & & & & \\ -2 & 5 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 5 & -2 \\ & & & & 1 & -2 & 1 \end{pmatrix}.$$

Notice, that the penalty matrix P is clearly rank deficient. A key property of the P-Spline modeling approach in (42) is that the dimensionality of the penalization term is greatly reduced compared to the penalty in (41), as we consider a discretized version of it (Schmid & Hothorn, 2008). This is what gives P-Spline a great computational advantage over cubic smoothing splines and thus makes it particularly interesting for applications such as L_2-Boosting, where (42) needs to be solved repeatedly.

F. Further Experimental Results

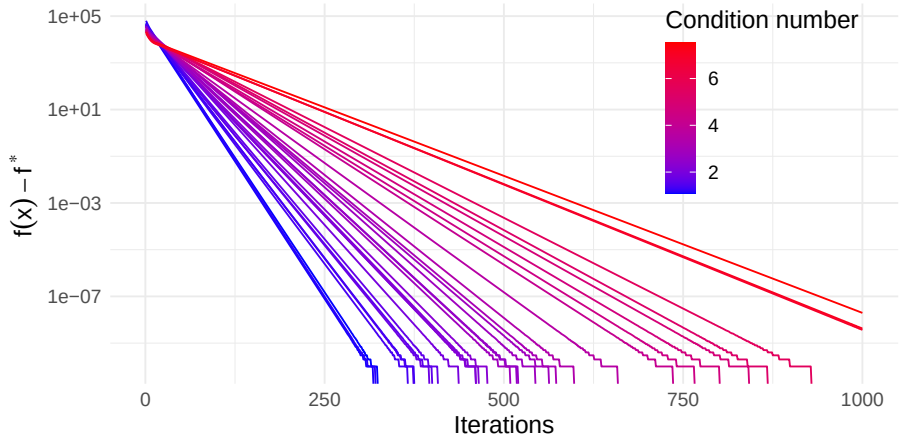

Figure 4. Linear convergence for different condition numbers (indicated by the color) for a linear model with two predictor variables. The condition numbers are induced by pairwise correlation of the predictor variables. Y axis on a logarithmic scale.