BOOSTING FOR HIGH-DIMENSIONAL LINEAR MODELS

By Peter Bühlmann

ETH Zürich

We prove that boosting with the squared error loss, L_2Boosting, is consistent for very high-dimensional linear models, where the number of predictor variables is allowed to grow essentially as fast as $O(\exp(\text{sample size}))$, assuming that the true underlying regression function is sparse in terms of the ℓ_1-norm of the regression coefficients. In the language of signal processing, this means consistency for de-noising using a strongly overcomplete dictionary if the underlying signal is sparse in terms of the ℓ_1-norm. We also propose here an *AIC*-based method for tuning, namely for choosing the number of boosting iterations. This makes L_2Boosting computationally attractive since it is not required to run the algorithm multiple times for cross-validation as commonly used so far. We demonstrate L_2Boosting for simulated data, in particular where the predictor dimension is large in comparison to sample size, and for a difficult tumor-classification problem with gene expression microarray data.

1. Introduction. Freund and Schapire's (1996) AdaBoost algorithm for classification has attracted much attention in the machine learning community (cf. Schapire, 2002, and the references therein) as well as in related areas in statistics (Breiman, 1998; Friedman et al., 2000), mainly because of its good empirical performance in a variety of data sets. Boosting methods have been originally introduced as multiple prediction schemes, averaging estimated predictions from re-weighted data. Later, Breiman (1998, 1999) noted that the AdaBoost algorithm can be viewed as a gradient descent optimization technique in function space. This important insight opened a new perspective, namely to use boosting methods in other contexts than classification. For example, Friedman (2001) developed boosting methods for regression which are implemented as an optimization using the squared error loss function: this is what we call L_2Boosting. It is essentially the same as Mallat and Zhang's (1993) matching pursuit algorithm in signal processing.

Recently, Efron et al. (2004) made a connection for linear models between forward stagewise linear regression (FSLR), which seems closely related to

AMS 2000 subject classifications: Primary 62J05,62J07; secondary 49M15,62P10,68Q32

Keywords and phrases: Binary classification, Gene expression, Lasso, Matching pursuit, Overcomplete dictionary, Sparsity, Variable selection, Weak greedy algorithm

L_2Boosting, and the ℓ_1-penalized Lasso (Tibshirani, 1996) or basis pursuit (Chen et al., 1999). Roughly speaking: under some restrictive assumptions on the design matrix of a linear model, FSLR approximately yields the set of all Lasso solutions (when varying over the penalty parameter). This intriguing insight may be useful to get a rough picture about L_2Boosting via its relatedness to FSLR: it does variable selection and shrinkage, similar to the Lasso. However, it should be stated clearly that the methods are not the same: an example showing a distinct difference between L_2Boosting and the Lasso is presented in section 4.3. Moreover, we point out in section 2.1 that FSLR and L_2Boosting are different algorithms as well.

As the main result, we prove here that L_2Boosting for linear models yields consistent estimates in the very high-dimensional context, where the number of predictor variables is allowed to grow essentially as fast as $O(\exp(\text{sample size}))$, assuming that the true underlying regression function is sparse in terms of the ℓ_1-norm of the regression coefficients. This result is, to our knowledge, the first about boosting in the presence of (fast) growing dimension of the predictor. Some consistency results for boosting with fixed predictor dimension include Mannor et al. (2002), Jiang (2004), Lugosi and Vayatis (2004) as well as Zhang and Yu (2003). Except Jiang's (2004) result, these authors consider versions of boosting with either ℓ_1-constraints for the boosting aggregation coefficients or, as in Zhang and Yu (2003), with a relaxed version of boosting which we found very difficult to use in practice due to the non-obvious tuning of the relaxation, i.e. how fast the boosting aggregation coefficients should decay. The result by Zhang and Yu (2003) may be generalized without too much effort to a setting with increasing dimension of the predictor variable, but their theoretical work includes only a rigorous treatment of the classification problem (besides the above mentioned disadvantage of their relaxed boosting algorithm). We believe that it is mainly for the case of high-dimensional predictors where boosting, among other methods, has a substantial advantage over more classical approaches. Some evidence for this will be given in section 4.1, and other supporting empirical results have been reported in Bühlmann and Yu (2003) in the different context of low- or high-dimensional additive models for comparing L_2Boosting with more traditional methods such as backfitting or MARS (restricted to additive function estimates). Notably, many real data-sets nowadays are of high-dimensional nature. Besides the well-documented good empirical performance of boosting, we identify it here as a method which can consistently recover very high-dimensional, sparse functions.

We may also view our result as a consistency property for de-noising using L_2Boosting with a strongly overcomplete dictionary. In contrast to a

complete dictionary, e.g. Fourier- or wavelet-basis, the strongly overcomplete noisy case is not well understood. Our result yields at least the basic property of consistency.

Besides the theoretical consistency result, we propose here a computationally efficient approach for the tuning parameter in boosting, i.e. the number of boosting iterations. We give some easily computable definition of degrees of freedom for L_2Boosting, and we then propose its use in the corrected AIC criterion. Unlike cross-validation, our AIC-tuning does not require boosting to be run multiple times. This makes the AIC-type data-driven boosting computationally attractive: depending on the data, it is sometimes as fast as the very efficient LARS algorithm for the Lasso with tuning by its default 10-fold cross-validation (Efron et al. (2004); `lars` package in `R` CRAN (1997 ff.).

We demonstrate on some simulated examples how our L_2Boosting performs for (low- and) mainly high-dimensional linear models, in comparison to the Lasso, forward variable selection, Ridge regression, ordinary least squares and a method which has been designed for high-dimensional regression (Goldenshluger and Tsybakov, 2001). We also consider a difficult tumor-classification problem with gene expression microarray data: the predictive accuracy of L_2Boosting is compared with four other, commonly used classifiers for microarray data, and we briefly indicate the interpretation of the L_2Boosting-fit along the lines of a linear model fit.

2. L_2Boosting with componentwise linear least squares. To explain boosting for linear models, consider a regression model

$$Y_i = \sum_{j=1}^{p} \beta_j X_i^{(j)} + \varepsilon_i,\ i = 1, \ldots, n,$$

with p predictor variables (the jth component of a p-dimensional vector x is denoted by $x^{(j)}$) and a random, mean zero error term ε. More precise assumptions for the model are given in section 3.

We first specify a base procedure: given some input data $\{(X_i, U_i);\ i = 1, \ldots, n\}$, where $U_1, \ldots, U_n$ denote some (pseudo-)response variables which are not necessarily the original $Y_1, \ldots, Y_n$, the base procedure yields an estimated function

$$\hat{g}(\cdot) = \hat{g}_{(\mathbf{X}, \mathbf{U})}(\cdot),$$

based on $\mathbf{X} = [X_i^{(j)}]_{i=1,\ldots,n;j=1,\ldots,p}$, $\mathbf{U} = (U_1, \ldots, U_n)^T$. Here, we will exclu-

sively consider the componentwise linear least squares base procedure:

$$\hat{g}_{(\mathbf{X},\mathbf{U})}(x) = \hat{\beta}_{\hat{\mathcal{S}}} x^{(\hat{\mathcal{S}})},\ \hat{\beta}_j = \frac{\sum_{i=1}^n U_i X_i^{(j)}}{\sum_{i=1}^n (X_i^{(j)})^2}\ (j = 1, \ldots, p),$$

$$\hat{\mathcal{S}} = \operatorname*{arg\,min}_{1 \le j \le p} \sum_{i=1}^n (U_i - \hat{\beta}_j X_i^{(j)})^2. \tag{2.1}$$

Thus, the componentwise linear least squares base procedure performs a linear least squares regression against the one selected predictor variable which reduces residual sum of squares most.

Boosting using the squared error loss, L_2Boosting, has a simple structure. Boosting algorithms using other loss functions are described in Friedman (2001).

L_2Boosting algorithm

Step 1 (initialization). Given data $\{(X_i, Y_i); i = 1, \ldots, n\}$, apply the base procedure yielding the function estimate

$$\hat{F}^{(1)}(\cdot) = \hat{g}(\cdot),$$

where $\hat{g} = \hat{g}_{(\mathbf{X},\mathbf{Y})}$ is estimated from the original data. Set $m = 1$.

Step 2. Compute residuals $U_i = Y_i - \hat{F}^{(m)}(X_i)$ $(i = 1, \ldots, n)$ and fit the real-valued base procedure to the current residuals. The fit is denoted by $\hat{g}^{(m+1)}(\cdot) = \hat{g}_{(\mathbf{X},\mathbf{U})}(\cdot)$ which is an estimate based on the original predictor variables and the current residuals.
Update

$$\hat{F}^{(m+1)}(\cdot) = \hat{F}^{(m)}(\cdot) + \hat{g}^{(m+1)}(\cdot).$$

Step 3 (iteration). Increase the iteration index m by one and repeat Step 2 until a stopping iteration M is achieved.

$\hat{F}^{(M)}(\cdot)$ is an estimator of the regression function $\mathbb{E}[Y|X = \cdot]$. L_2Boosting is nothing else than repeated least squares fitting of residuals (cf. Friedman (2001), Bühlmann and Yu (2003)). With $m = 2$ (one boosting step), it has already been proposed by Tukey (1977) under the name "twicing". In the non-stochastic context, the L_2Boosting algorithm is known as "Matching

Pursuit" (Mallat and Zhang, 1993) which is popular in signal processing for fitting overcomplete dictionaries.

It is often better to use small step sizes: we advocate here to use the step-size ν in the update of $\hat{F}^{(m+1)}$ in step 2 which then becomes

$$\hat{F}^{(m+1)}(\cdot) = \hat{F}^{(m)}(\cdot) + \nu \hat{g}^{(m+1)}(\cdot),\ 0 < \nu \leq 1, \tag{2.2}$$

where ν is constant during boosting iterations and small, e.g. $\nu = 0.1$. The parameter ν can be seen as a shrinkage parameter or alternatively, describing the step-size when up-dating $\hat{F}^{(m+1)}(\cdot)$ along the function $\hat{g}^{(m+1)}(\cdot)$. Small step-sizes (or shrinkage) make the boosting algorithm slower and require a larger number M of iterations. However, the computational slow-down often turns out to be advantageous for better out-of-sample empirical prediction performance, cf. Friedman (2001), Bühlmann and Yu (2003).

2.1. *Forward stagewise linear regression.* L_2Boosting with componentwise linear least squares is related to forward stagewise linear regression (FSLR), as pointed out by Efron et al. (2004). FSLR differs from L_2Boosting with componentwise linear least squares in the update of the new estimate $\hat{F}_m$: instead of using (2.2) which becomes

$$\hat{F}_m(x) = \hat{F}_{m-1}(x) + \nu \hat{\beta}_{\hat{\mathcal{S}}_m} x^{(\hat{\mathcal{S}}_m)},$$

where $\hat{\beta}_{\hat{\mathcal{S}}_m}$ is the least squares estimate when fitting the current residuals against the best predictor variable $x^{(\hat{\mathcal{S}}_m)}$, FSLR updates

$$\hat{F}_{m;FSLR}(x) = \hat{F}_{m-1;FSLR}(x) + \nu\, sign(\hat{\beta}_{\hat{\mathcal{S}}_m}) x^{(\hat{\mathcal{S}}_m)}.$$

Note that this description of FSLR is equivalent to the one in Efron et al. (2004). In our limited experience, FSLR has about the same prediction accuracy as L_2Boosting with componentwise linear least squares. However, we give here two reasons to favor boosting over FSLR. First, the update in FSLR is not scale-invariant whereas the boosting update is on the scale of the current residuals via the magnitude of the least squares estimate $\hat{\beta}_{\hat{\mathcal{S}}_m}$. It implies that FSLR is often more sensitive to the choice of ν than boosting. In particular, in case of an orthogonal linear model, L_2Boosting has a uniform approximation property for the soft-threshold estimator over all values of the threshold parameter, whereas this nice property does not hold anymore for FSLR (Bühlmann and Yu, 2005). Second, the number of boosting iterations can be reasonably well estimated via degrees of freedom defined as the trace of a boosting hat matrix, as to be described in section 2.2.

Defining reasonable degrees of freedom which are simple to compute seems not easily possible for FSLR. This has also been pointed out by Efron et al. (2004, comment after formula (4.11)), and they suggest the computationally intensive bootstrap to cope with this problem.

We emphasize that Efron et al. (2004) do not advocate to use FSLR in practice. They rather focus on the more interesting LARS algorithm.

2.2. *Stopping the boosting iterations.* Boosting needs to be stopped at a suitable number of iterations, to avoid overfitting. The computationally efficient AIC_c criterion in (2.3) below can be used in our context where the base procedure has linear components.

Our goal here is to assign degrees of freedom for boosting. Denote by

$$\mathcal{H}^{(j)} = \mathbf{X}^{(j)}(\mathbf{X}^{(j)})^T/\|\mathbf{X}^{(j)}\|^2,\ j = 1, \ldots, p,$$

the $n \times n$ hat-matrix for the linear least squares fitting operator using the jth predictor variable $\mathbf{X}^{(j)} = (X_1^{(j)}, \ldots, X_n^{(j)})^T$ only; $\|x\|^2 = x^T x$ denotes the Euclidean norm for a vector $x \in \mathbb{R}^n$. It is then straightforward to show (Bühlmann and Yu, 2003) that the L_2Boosting hat-matrix, when using the step size $0 < \nu \leq 1$, equals,

$$\mathcal{B}_m = I - (I - \nu\mathcal{H}^{(\hat{\mathcal{S}}_m)})(I - \nu\mathcal{H}^{(\hat{\mathcal{S}}_{m-1})}) \ldots (I - \nu\mathcal{H}^{(\hat{\mathcal{S}}_1)}),$$

where $\hat{\mathcal{S}}_i \in \{1, \ldots, p\}$ denotes the component which is selected in the componentwise least squares base procedure in the ith boosting iteration.

Using the trace of $\mathcal{B}_m$ as degrees of freedom, we employ a corrected version of AIC (Hurvich et al., 1998, cf.) to define a stopping rule for boosting:

$$AIC_c(m) = \log(\hat{\sigma}^2) + \frac{1 + \text{trace}(\mathcal{B}_m)/n}{1 - (\text{trace}(\mathcal{B}_m) + 2)/n},$$

$$\hat{\sigma}^2 = n^{-1}\sum_{i=1}^{n}(Y_i - (\mathcal{B}_m\mathbf{Y})_i)^2,\ \mathbf{Y} = (Y_1, \ldots, Y_n)^T. \tag{2.3}$$

An estimate for the number of boosting iterations is then

$$\hat{M} = \operatorname*{arg\,min}_{1 \leq m \leq m_{upp}} AIC_c(m),$$

where m_{upp} is a large, upper bound for the candidate number of boosting iterations.

3. Consistency of L_2Boosting in high dimensions. We present here a consistency result for L_2Boosting in linear models where the number of predictors is allowed to grow very fast as the sample size n increases. Consider the model

$$\begin{aligned} &Y_i = f_n(X_i) + \varepsilon_i,\ i = 1, \dots, n, \\ &f_n(x) = \sum_{j=1}^{p_n} \beta_{j,n} x^{(j)},\ x \in \mathbb{R}^{p_n}, \end{aligned} \tag{3.1}$$

where $X_1, \dots, X_n$ are i.i.d. with $\mathbb{E}|X^{(j)}|^2 \equiv 1$ for all $j = 1, \dots, p_n$ and $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d., independent from $\{X_s; 1 \le s \le n\}$, with $\mathbb{E}[\varepsilon] = 0$. The case with heteroscedastic ε_i's and potential dependence between ε_i and X_i is discussed in Remark 3 below. The number of predictors p_n is allowed to grow with sample size n. Therefore, also the predictor $X_i = X_{i,n}$ and the response $Y_i = Y_{i,n}$ depend on n, but we usually ignore this in the notation. The scaling of the predictor variables $\mathbb{E}|X^{(j)}|^2 = 1$ is not necessary for running L_2Boosting, but it allows to identify the magnitude of the coefficients $\beta_{j,n}$ (see also assumption (A1) below).

We make the following assumptions.

(A1) The dimension of the predictor in model (3.1) satisfies $p_n = O(\exp(Cn^{1-\xi}))$ $(n \to \infty)$, for some $0 < \xi < 1,\ 0 < C < \infty$.
(A2) $\sup_{n \in \mathbb{N}} \sum_{j=1}^{p_n} |\beta_{j,n}| < \infty$.
(A3) $\sup_{1 \le j \le p_n, n \in \mathbb{N}} \|X^{(j)}\|_\infty < \infty$, where $\|X\|_\infty = \sup_{\omega \in \Omega} |X(\omega)|$ (Ω denotes the underlying probability space).
(A4) $\mathbb{E}|\varepsilon|^s < \infty$ for some $s > 4/\xi$ with ξ from (A1).

Assumption (A1) allows for a very large predictor dimension relative to the sample size n. Assumption (A2) is a ℓ_1-norm sparseness condition (it could be generalized to $\sum_{j=1}^{p_n} |\beta_{j,n}| \to \infty$ sufficiently slowly as $n \to \infty$, at the expense of additional restrictions on p_n). Even if p_n grows, all predictors may be relevant but most of them contribute only with small magnitudes (small $|\beta_{j,n}|$). Assumption (A2) holds for regressions where the number of effective predictors is finite and fixed: that is, the number of $\beta_{j,n} \neq 0$ is independent from n and finite. Assumption (A3) about the boundedness of the predictor variables can be relaxed at the price of a more restrictive growth of $p = p_n$, see Remark 1 below.

THEOREM 1. *Consider the model (3.1) satisfying (A1)-(A4). Then, the boosting estimate $\hat{F}^{(m)}(\cdot) = \hat{F}_n^{(m)}(\cdot)$ with the componentwise linear base procedure from (2.1) satisfies: for some sequence $(m_n)_{n \in \mathbb{N}}$ with $m_n \to \infty$*

($n \to \infty$) sufficiently slowly,

$$\mathbb{E}_X |\hat{F}_n^{(m_n)}(X) - f_n(X)|^2 = o_P(1) \ (n \to \infty),$$

where X denotes a new predictor variable, independent of and with the same distribution as the X-component of the data (X_i, Y_i) $(i = 1, \ldots, n)$.

A proof is given in section 6. Theorem 1 says that L_2Boosting recovers the true sparse regression function even if the number of predictor variables is essentially exponentially increasing with sample size n. Notably, no assumptions are needed on the correlation structure of the predictor variables.

REMARK 1. *Assumption (A3) requires boundedness of the predictor variables. Theorem 1 also holds under the assumption*

$$\sup_{1 \le j \le p_n} \mathbb{E}|X^{(j)}|^s < \infty \text{ for some } s \ge 4$$

if the growth of dimension is restricted to $p_n = O(n^\alpha)$ where $\alpha = \alpha(s) > 0$ is a number, depending on the number of existing moments s, which converges monotonically to ∞ as s increases, i.e. any polynomial growth of p_n is allowed if the number of moments s is sufficiently large.

REMARK 2. *For the Lasso, a consistency for high-dimensional regression has been given by Greenshtein and Ritov (2004). Their result covers the case where $p_n = O(n^\alpha)$ for any $\alpha > 0$ and the ℓ_1-norm of the coefficients $\sum_{j=1}^{p_n} |\beta_{j,n}|$ is allowed to grow with n, see also Remark 1 above. We should keep in mind however, that the Lasso is a different estimator than L_2Boosting, as will be demonstrated on an empirical example in section 4.3.*

REMARK 3. *Theorem 1 also holds for possibly heteroscedastic errors ε_i which are potentially dependent of X_i, by assuming $(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d. and suitable moment conditions for Y_i. For the case with bounded Y_i, a proof follows as for Corollary 1 below.*

3.1. *Binary classification.* The case of binary classification with $Y_i \in \{0, 1\}$ can be essentially deduced from squared error regression. Bühlmann and Yu (2003) argue why L_2Boosting is also a reasonable procedure for binary classification. We can always write

$$\begin{aligned} &Y_i = f_n(X_i) + \varepsilon_i, \\ &f_n(x) = \mathbb{E}[Y|X = x] = \mathbb{P}[Y = 1|X = x], \ \varepsilon_i = Y_i - f_n(X_i), \end{aligned} \tag{3.2}$$

where the $\varepsilon_1, \ldots, \varepsilon_n$ are independent but heteroscedastic with $\mathbb{E}[\varepsilon_i] = 0$ and $Var(\varepsilon_i) = f_n(X_i)(1 - f_n(X_i))$. When using L_2Boosting, we get an estimate for the conditional probability function $\mathbb{P}[Y = 1|X = x]$, and the L_2Boosting plug-in classifier (for equal misclassification costs) is given by $\hat{C}_n^{(m)}(x) = \mathbb{1}_{[\hat{F}_n^{(m)}(x) > 1/2]}$.

The proof of Theorem 1 essentially goes through and we get the following.

COROLLARY 1. *Consider a binary classification problem with* $(X_1, Y_1), \ldots, (X_n, Y_n)$ *independent and* $Y_i \in \{0, 1\}$ *for all* $i = 1, \ldots, n$. *Denote by* $f_n(x) = \mathbb{P}_n[Y = 1|X = x]$ *and assume (A1)-(A3). Then, for the* L_2*Boosting estimate as in Theorem 1: for some sequence* $(m_n)_{n \in \mathbb{N}}$ *with* $m_n \to \infty$ $(n \to \infty)$ *sufficiently slowly,*

$$\mathbb{E}_X |\hat{F}_n^{(m_n)}(X) - f_n(X)|^2 = o_P(1) \ (n \to \infty),$$
$$\mathbb{P}_{X,Y}[\hat{C}_n^{m_n}(X) \neq Y] - L_{n,Bayes} = o_P(1) \ (n \to \infty),$$

where $L_{n,Bayes}$ *denotes the Bayes risk* $\mathbb{E}_X[\min\{f_n(X), 1 - f_n(X)\}]$ *and* X, Y *denote new response and predictor variables, independent of and with the same distribution as the data* (X_i, Y_i) $(i = 1, \ldots, n)$.

A proof is given in section 6. □

4. Numerical results.

4.1. *Low-dimensional regression surface within low- or high-dimensional predictor space.* We consider the model

$$X \sim \mathcal{N}_p(0, V), \ Y = f(X) + \varepsilon, \ p \in \{3, 10, 100\},$$
$$f(X) = a(V)(1 + 5X^{(1)} + 2X^{(2)} + X^{(3)}), \ \varepsilon \sim \mathcal{N}(0, 2^2), \tag{4.1}$$

where $a(V)$ is a scaling factor. The covariance matrix for the predictor variable X and the factor $a(V)$ are chosen as:

$$V = I_p, \ a(V) = 1 \tag{4.2}$$

for uncorrelated predictors; or for block-correlated predictors,

$$V = \begin{pmatrix} 1 & b & c & 0 & \ldots & \ldots & \ldots & 0 \\ b & 1 & b & c & 0 & \ldots & \ldots & 0 \\ c & b & 1 & b & c & 0 & \ldots & 0 \\ 0 & c & b & 1 & b & c & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & c & b & 1 & b & c \\ 0 & \ldots & \ldots & 0 & c & b & 1 & b \\ 0 & \ldots & \ldots & \ldots & 0 & c & b & 1 \end{pmatrix},$$

$$b = 0.677, c = 0.323,\ a(V) = 0.779. \tag{4.3}$$

The constant $a(V)$ is such that the signal to noise ratio $\mathbb{E}|f(X)|^2/\sigma_\varepsilon^2$ is about the same for both model specifications. The model (4.1) with either specification (4.2) or (4.3) has only 3 effective predictors plus an intercept, all of them contributing to the regression function with different magnitudes (different coefficients). We choose sample size $n = 20$, i.e. we generate 20 i.i.d. realizations $(X_i, Y_i),\ i = 1, \ldots 20$ from the model. The case with $p = 3$ represents a low-dimensional model; for $p \in \{10, 100\}$, relative to the number of observations n, the problem is high-dimensional with a low-dimensional (effective $p_{eff} + 1 = 4$) true underlying structure.

We use L_2Boosting, using shrinkage factor $\nu = 0.1$ (see (2.2)) and the corrected AIC criterion for stopping the boosting iterations (see (2.3)). We compare it with the Lasso using 10-fold cross-validation for selecting the penalty parameter (i.e. using the default-setting from the `lars` package in `R` with 10-fold cross-validation (CRAN, 1997 ff.), with forward variable selection for optimizing the classical AIC criterion, with ordinary least squares (OLS) without variable selection and with Ridge regression using the oracle Ridge-penalty parameter which minimizes the squared error loss over the simulations; the latter cannot be used in practice but serves as an optimistic value for the performance of Ridge regression. Table 4.1 reports in detail the

method	(4.2), $p = 3$	(4.2), $p = 10$	(4.2), $p = 100$
L_2Boost	1.658 (0.192)	2.318 (0.238)	8.792 (0.640)
Lasso	1.597 (0.240)	3.385 (0.447)	8.557 (0.751)
fwd.var.sel.	1.499 (0.215)	3.648 (0.421)	13.551 (1.275)
Ridge*	1.079 (0.117)	4.436 (0.392)	25.748 (0.637)
OLS	1.103 (0.127)	5.674 (0.556)	–
	(4.3), $p = 3$	(4.3), $p = 10$	(4.3), $p = 100$
L_2Boost	1.054 (0.104)	1.649 (0.181)	4.643 (0.239)
Lasso	1.727 (0.269)	3.105 (0.473)	3.770 (0.402)
fwd.var.sel.	1.206 (0.104)	2.893 (0.373)	12.685 (0.911)
Ridge*	0.777 (0.079)	2.442 (0.226)	20.799 (0.538)
OLS	1.103 (0.127)	5.674 (0.556)	–

Table 4.1: Mean squared error $\mathbb{E}[(\hat{f}(X) - f(X))^2]$ for L_2Boosting, Lasso, forward variable selection (fwd.var.sel.), Ridge regression with the oracle penalty (Ridge*) and ordinary least squares (OLS) in model (4.1) with specifications (4.2) and (4.3). Sample size $n = 20$. Estimated standard errors from independent model simulations are given in parentheses.

mean squared error MSE $= \mathbb{E}[(\hat{f}(X) - f(X))^2]$ where X is a new test observation, independent from but with the same distribution as the training data. Figure 4.1 summarizes one of the settings. All results are based on 50 model simulations.

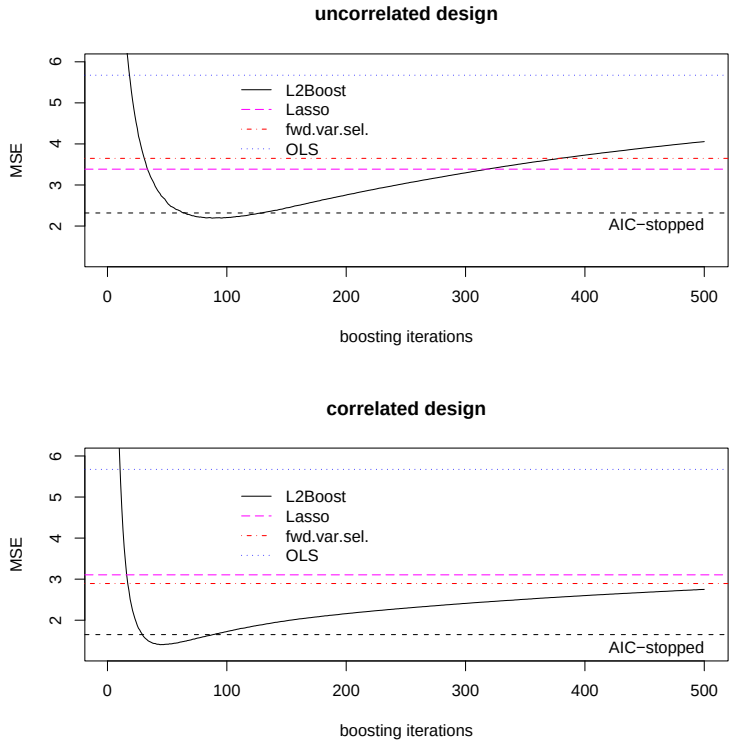

Figure 4.1: Mean squared error $\mathbb{E}[(\hat{f}(X) - f(X))^2]$ for L_2Boosting as a function of boosting iterations (solid line), for L_2Boosting with AIC_c-stopping (dashed line denoted by AIC-stopped), Lasso (long-dashed line), forward variable selection (dashed-dotted line) and ordinary least squares (dotted line) in model (4.1) with $p = 10$ and specifications (4.2) (top panel) and (4.3) (bottom panel). Sample size $n = 20$.

For the high-dimensional settings with $p \in \{10, 100\}$, L_2Boosting and the Lasso are clearly best for this model with very few effective predictors (see Table 4.1). Figure 4.1 displays the good performance of the corrected AIC_c criterion in (2.3) for stopping the boosting iterations. A detailed comparison of the "oracle"-stopping rule of L_2Boosting which stops at the boosting

iteration minimizing the mean squared error (see Table 4.2) can be made to the results in Table 4.1. Obviously, the "oracle"-rule can only be applied for simulated data. We also include in Table 4.2 the performance of the Lasso with the "oracle" penalty parameter minimizing the mean squared error. L_2Boosting and the Lasso perform similarly when using the "oracle" tuning parameters (see Table 4.2), while the differences are somewhat more pronounced when comparing AIC_c-stopped L_2Boosting with Lasso using 10-fold CV tuning (see Table 4.1).

model	L_2Boost*	Lasso*
(4.2), $p = 3$	1.103 (0.127)	1.103 (0.127)
(4.3), $p = 3$	0.891 (0.100)	1.075 (0.117)
(4.2), $p = 10$	2.193 (0.230)	2.208 (0.262)
(4.3), $p = 10$	1.404 (0.114)	1.378 (0.116)
(4.2), $p = 100$	7.583 (0.593)	7.116 (0.603)
(4.3), $p = 100$	2.995 (0.208)	2.730 (0.234)

Table 4.2: Mean squared error $\mathbb{E}[(\hat{f}(X) - f(X))^2]$ (MSE) for L_2Boosting (L_2Boost*) and the Lasso (Lasso*), both with their "oracle"-tuning parameter minimizing the MSE. The model is as in (4.1) with specifications (4.2) and (4.3). Sample size $n = 20$. Estimated standard errors from independent model simulations are given in parentheses.

Finally, we also include a small study when p increases exponentially while n grows only linearly. We focus on the model (4.1) with (4.2) for $(n, p) = (20, 3), (40, 30), (60, 300)$. The results for L_2Boosting and the Lasso are given in Table 4.3. For both L_2Boosting and the Lasso, the mean squared error exhibits only a slow increase as n grows linearly and p exponentially; compare also with the results from Table 4.1 with fixed $n = 20$. For this par-

(n, p)	L_2Boost	Lasso	L_2Boost*	Lasso*
$(20, 3)$	1.658 (0.192)	1.597 (0.240)	1.103 (0.127)	1.103 (0.127)
$(40, 30)$	2.090 (0.199)	2.800 (0.289)	1.730 (0.169)	1.438 (0.120)
$(60, 300)$	3.652 (0.186)	2.430 (0.169)	2.372 (0.135)	1.855 (0.122)

Table 4.3: Mean squared error $\mathbb{E}[(\hat{f}(X) - f(X))^2]$ (MSE). L_2Boosting with AIC_c-stopping (L_2Boost) and with the "oracle"-stopping (L_2Boost*); Lasso with 10-fold CV tuning (Lasso) and with "oracle" tuning (Lasso*). The model is as in (4.1) with specification (4.2). Estimated standard errors from independent model simulations are given in parentheses.

ticular example, the "oracle"-tuned Lasso is slightly better for large $p = 300$ than the "oracle"-stopped L_2Boosting (but this is not a general superiority).

4.2. *High-dimensional regression surface with ℓ_1-coefficients.* We consider here a regression model which fits into the theory of an adaptive estimation procedure for high-dimensional linear regression, presented by Goldenshluger and Tsybakov (2001).

The model is

$$X \sim \mathcal{N}_p(0, I),\ Y = \sum_{j=1}^{p} \beta_j X^{(j)} + \varepsilon,$$
$$\beta_j \sim \mathcal{N}(0, \sigma_j^2)\ (j = 1, \ldots, p),\ \varepsilon \sim \mathcal{N}(0, 1), \tag{4.4}$$

where ε, X and $\beta_1, \ldots, \beta_p$ are independent of each other. The values σ_j^2 are decreasing as j increases. Thus, absolute values of the regression coefficients $|\beta_j|$ have a tendency to become small for large j. A precise description of the model is given in the Appendix. To summarize, the model is such that $p = p_n$ and $\beta_j = \beta_{j,n}$ $(j = 1, \ldots, p_n)$ depend on n, satisfying with high probability $\sup_{n\in\mathbb{N}} \sum_{j=1}^{p_n} |\beta_{j,n}| < \infty$, which is our assumption (A2) from section 3. Sample size is chosen as $n = 100$ and the resulting dimension of the predictor then equals $p = 23$.

We use L_2Boosting, using shrinkage $\nu = 0.1$ (see (2.2)) and with estimated number of boosting iterations via the corrected AIC criterion as in (2.3), and we compare it with the Lasso (using the default-setting from the `lars` package in `R` with 10-fold cross-validation (CRAN, 1997 ff.)), forward variable selection for optimizing the classical AIC criterion, with Ridge regression using 10-fold cross-validation for selecting the Ridge parameter, with ordinary least squares and with the procedure from (Goldenshluger

L_2Boost	Lasso	G&T
0.132 (0.006)	0.159 (0.011)	0.195 (0.047)
fwd.var.sel.	Ridge	OLS
0.279 (0.019)	0.116 (0.008)	0.313 (0.017)

Table 4.4: Mean squared error $\mathbb{E}[(\hat{f}(X) - f(X))^2]$ for L_2Boosting, Lasso, the method from Goldenshluger and Tsybakov (G&T), forward variable selection (fwd.var.sel.), Ridge regression (Ridge) and ordinary least squares (OLS) in model (4.4). Sample size $n = 100$. Estimated standard errors from independent model simulations are given in parentheses.

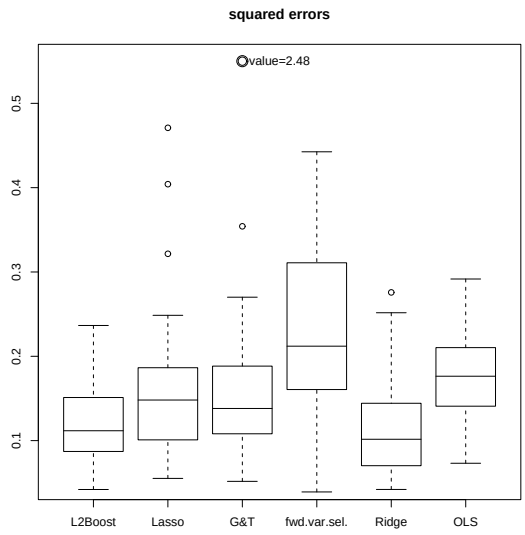

Figure 4.2: Boxplots of squared errors $(\hat{f}(X)-f(X))^2$ for L_2Boosting, Lasso, the method from Goldenshluger and Tsybakov (G&T), forward variable selection (fwd.var.sel.), Ridge regression (Ridge) and ordinary least squares (OLS) in model (4.4). One outlier with value=2.48 occurred using the G&T method. Sample size $n = 100$.

and Tsybakov, 2001). Table 4.4 and Figure 4.2 display the results which are based on 50 independent model simulations. The method from (Goldenshluger and Tsybakov, 2001) produced one outlier with very large squared error. L_2Boosting and Ridge regression perform best for this model.

Moreover, the method from (Goldenshluger and Tsybakov, 2001) depends on the indexing of the predictor variables and is tailored for regression problems where the coefficients β_j have a tendency to decay as j increases (e.g. in time series where j indicates the jth lagged variable). All other methods are not depending on indexing the predictor variables. We also ran the method from (Goldenshluger and Tsybakov, 2001) on the same model but with index-reversed regression coefficients

$$\beta_1, \dots \beta_{23} = \tilde{\beta}_{23}, \dots, \tilde{\beta}_1, \ \tilde{\beta}_j \text{ as in (4.4)}. \tag{4.5}$$

The mean squared error was then

MSE for G&T method with (4.5): 0.224 (0.025)

which shows very clearly the sensitivity of indexing the variables.

4.3. *L_2Boosting is different from Lasso.* Consider a model with predictors as in (4.1) and (4.2) with $p = 100$ but with regression function

$$f(X) = 0.2 + 0.2 \sum_{j=1}^{100} X^{(j)} \tag{4.6}$$

and noise $\varepsilon \sim \mathcal{N}(0, 0.5^2)$. Sample size is chosen as $n = 20$. This model is high-dimensional and non-sparse, and it has a high signal to noise ratio.

Since all the predictors contribute equally, we may want to keep many of the variables in the model and shrink their corresponding coefficient estimates to zero. However, the Lasso will only allow to select at most $\min(n, p + 1) = 20$ predictor variables (including an intercept), cf. Zou and Hastie (2003). When generating one realization of the model (4.6), L_2Boosting with the AIC_c-stopping rule selected 42 predictor variables (including the intercept), whereas the corresponding number of selected variables with Lasso, tuned by 10-fold cross-validation, is 19 only. Thus, we have here an example which demonstrates a feature of L_2Boosting which is qualitatively different to the Lasso.

A comparison in terms of performances is given in Table 4.5. The methods are described in section 4.2. It is no surprise that Ridge regression (using 10-

L_2Boost	Lasso	Ridge
9.468 (0.251)	11.519 (0.322)	5.548 (0.229)

Table 4.5: Mean squared error $\mathbb{E}[(\hat{f}(X) - f(X))^2]$ for L_2Boosting, Lasso and Ridge regression in model (4.6). Sample size $n = 20$. Estimated standard errors from independent model simulations are given in parentheses.

fold cross-validation for tuning) performs clearly best. It keeps all variables in the model and shrinks the corresponding estimates towards zero: this is tailored for the structure of the model (4.6) where all the variables contribute equally. We also see from the mean squared error, that L_2Boosting is quite different (in fact better) than the Lasso.

It is not difficult to extend this example such that Ridge regression becomes worse than L_2Boosting. Take p large such as $p = 1000$ and use the same function $f(X) = 0.2 + 0.2 \sum_{j=1}^{100} X^{(j)}$ which depends only on the first 100 components of X. Ridge is expected to perform poorly, because it uses all p (e.g. $= 1000$) predictor variables, while L_2Boosting remains to be a bit better than the Lasso.

4.4. *Gene expression microarray data.* We consider a dataset which monitors $p = 7129$ gene expressions in 49 breast tumor samples using the Affymetrix technology, see West et al. (2001). After thresholding to a floor of 100 and a ceiling of 16,000 expression units, we applied a base 10 log-transformation and standardized each experiment to zero mean and unit variance. For each sample, a binary response variable is available, describing the status of lymph node involvement in breast cancer. The data are available at `http://mgm.duke.edu/genome/dna_micro/work/`.

We use L_2Boosting although the data has the structure of a binary classification problem; section 3.1 and Corollary 1 yield a justification for it, and e.g. Zou and Hastie (2003) also use some penalized squared error regression for binary classification with microarray gene expression predictors. The only modification is the AIC stopping criterion: instead of (2.3), we use

$$AIC(m) = -2 \cdot \text{log-likelihood} + 2 \cdot \text{trace}(\mathcal{B}_m),$$

with the Bernoulli log-likelihood. Instead of L_2Boosting, we could also use the LogitBoost algorithm (Friedman et al., 2000): for stopping, the penalty-term in the AIC criterion above then needs some modification since LogitBoost involves another operator than $\mathcal{B}_m$.

We estimate the classification performance by a cross-validation scheme where we randomly divide the 49 samples into balanced training- and test-data of sizes $2n/3$ and $n/3$, respectively, and we repeat this 50 times. We

	L_2Boost	FPLR	1-NN	DLDA	SVM
misclassifications	30.50%	35.25%	43.25%	36.12%	36.88%

Table 4.6: Cross-validated misclassification rates for lymph node breast cancer data. L_2Boosting is with linear least squares and AIC-stopping (L_2Boost), forward variable selection penalized logistic regression (FPLR), 1-nearest-neighbor rule (1-NN), diagonal linear discriminant analysis (DLDA) and a support vector machine (SVM); the latter three are based on 200 best genes (on each training dataset) according to a Wilcoxon score.

compare L_2Boosting with AIC-stopping (as described above) with four other classification methods: 1-nearest neighbors, diagonal linear discriminant analysis, support vector machine with radial basis kernel (from the R-package `e1071` and using its default values), and a forward selection penalized logistic regression model (using some reasonable penalty parameter and number of selected genes). For 1-nearest neighbors, diagonal linear discriminant analysis and support vector machine, we pre-select the 200 genes

which have the best Wilcoxon score in a two-sample problem (estimated from the training dataset only), which is recommended to improve the classification performance, see Dudoit et al. (2002). Our L_2Boosting and the forward variable selection penalized regression are run without pre-selection of genes. The results are given in Table 4.6.

For this difficult classification problem, our L_2Boosting with componentwise linear least squares performs well. It is also interesting to note that the minimal cross-validated misclassification rate as a function of boosting iterations is 29.25%. It shows that the AIC-stopping rule is very accurate for this example. The only method which we found to perform better for this dataset is the recently proposed Pelora algorithm (Dettling and Bühlmann, 2004) which does supervised gene grouping: its misclassification rate is 27.88%.

We also show in Figure 4.3 the estimated regression coefficients for the 42 genes which have been selected during the boosting iterations until AIC-stopping; the AIC-curve is also shown in Figure 4.3. For comparing the influence of different genes, we display scaled coefficients $\hat{\beta}_j \sqrt{Var(X^{(j)})}$ which correspond to the estimated coefficients when standardizing the genes to unit variance. There is one gene whose positive expression strongly points towards the class with $Y = 0$ (having negative scaled regression coefficient) and there are 5 genes whose positive expressions point towards the class

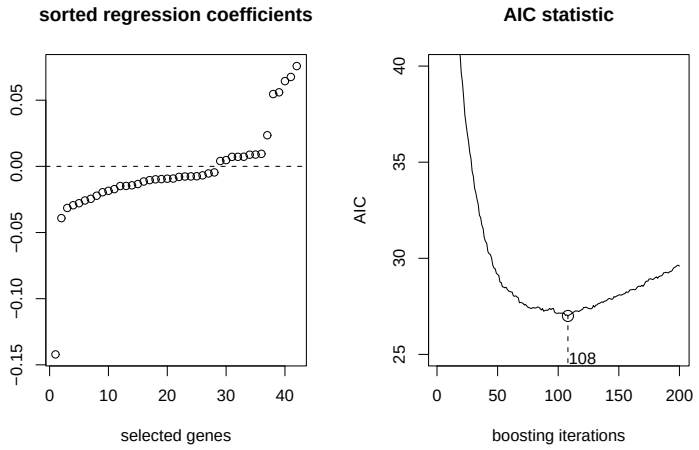

Figure 4.3: Lymph node breast cancer data. Left: scaled regression coefficients $\hat{\beta}_j \sqrt{Var(X^{(j)})}$ (plotted in increasing order) from L_2Boosting for the selected 42 genes. Right: AIC-statistic as a function of L_2Boosting iterations with minimum at 108.

with $Y = 1$. The smallest scaled regression coefficient corresponds to a gene which appears as the second best when ranking all the genes with the score of a two-sample Wilcoxon test; the five largest scaled coefficients correspond to the Wilcoxon-based ranks 7, 6, 1, 121, 3 among all the genes. But it should be emphasized that, as usual, our estimated regression model takes partial correlations between gene expressions (given all other remaining genes) into account which goes well beyond describing the effects of single genes only.

5. Conclusions. We consider L_2Boosting for fitting linear models. The method does variable selection and shrinkage, a property which is very useful in practical applications. This indicates that L_2Boosting is related to the ℓ_1-penalized Lasso, but the methods are not the same.

As a useful device, we propose a simple estimate for the number of boosting iterations, which is the tuning parameter of the method, by using a corrected AIC_c criterion. This makes boosting computationally attractive, since we do not have to run it multiple times in a cross-validation set-up.

We then present some theory for very high-dimensional regression (or for de-noising with strongly overcomplete dictionaries), saying that if the underlying true regression function is sparse in terms of the ℓ_1-norm of the regression coefficients, L_2Boosting consistently estimates the true regression function, even when the number of predictor variables grows like $p_n = O(\exp(n^{1-\xi}))$ for some (small) $\xi > 0$. Notably, no assumptions are made on the correlation structure of the predictors. Thus, we identify L_2Boosting as a method which is able, under mild assumptions, to consistently recover very high-dimensional, sparse functions.

6. Proofs. We first consider the regression case where the step-size in (2.2) equals $\nu = 1$. In section 6.3, we give the argument for arbitrary, fixed $0 < \nu \leq 1$. Finally, we present the case for binary classification in section 6.4.

6.1. *A population version.* The L_2Boosting algorithm has a population version which is known as "matching pursuit" (Mallat and Zhang, 1993) or "weak greedy algorithm" (Temlyakov, 2000).

Consider the Hilbert space $L_2(P) = \{f; \|f\|^2 = \int f(x)^2 dP(x) < \infty\}$ with inner product $\langle f, g\rangle = \int f(x)g(x)dP(x)$. Here, the probability measure P is generating the predictor X in model (3.1). To be precise, the probability measure $P = P_n$ depends on n since the dimensionality of X is growing with n: we are actually looking at a sequence of Hilbert spaces $L_2(P_n)$ but we often ignore this notationally (a uniform bound in (6.5) will be a key result to deal with such sequences of Hilbert spaces).

Denote the components of X by

$$g_j(x) = x^{(j)},\ j = 1, \ldots, p_n.$$

Note that by assumption, $\|g_j\| = 1$ for all j. Define the following sequence of remainder functions, called matching pursuit or weak greedy algorithm:

$$\begin{aligned} &R^0 f = f, \\ &R^m f = R^{m-1} f - \langle R^{m-1} f, g_{\mathcal{S}_m} \rangle\, g_{\mathcal{S}_m},\ m = 1, 2, \ldots \end{aligned} \tag{6.1}$$

where $\mathcal{S}_m$ would be ideally chosen as

$$\mathcal{S}_m = \operatorname{argmax}_{1 \le j \le p_n} |\langle R^{m-1} f, g_j \rangle|.$$

The choice function $\mathcal{S}_m$ is sometimes infeasible to realize in practice. A weaker criterion is: for every m (under consideration), choose any $\mathcal{S}_m$, which satisfies

$$|\langle R^{m-1} f, g_{\mathcal{S}_m} \rangle| \ge b \cdot \sup_{1 \le j \le p_n} |\langle R^{m-1} f, g_j \rangle| \text{ for some } 0 < b \le 1. \tag{6.2}$$

Of course, the sequence $R^m f = R^{m,\mathcal{S}} f$ depends on $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m$ how we actually make the choice in (6.2). Again, we will ignore this notationally.

It easily follows that

$$f = \sum_{j=0}^{m-1} \langle R^j f, g_{\mathcal{S}_{j+1}} \rangle\, g_{\mathcal{S}_{j+1}} + R^m f,$$

and

$$\|R^m f\|^2 = \|R^{m-1} f\|^2 - |\langle R^{m-1} f, g_{\mathcal{S}_m} \rangle|^2 \tag{6.3}$$

6.1.1. *Temlyakov's result.* Temlyakov (2000) gives a uniform bound for the algorithm in (6.1) with (6.2).

If the function f is representable as

$$f(x) = \sum_j \beta_j g_j(x),\ \sum_j |\beta_j| \le B < \infty, \tag{6.4}$$

which is true by our assumption (A2), then

$$\|R^m f\| \le B(1 + mb^2)^{-b/(2(2+b))},\ 0 < b \le 1 \text{ as in (6.2)}. \tag{6.5}$$

By construction, $R^m f$ depends on the selectors $\mathcal{S}_1, \ldots, \mathcal{S}_m$ in (6.2). The mathematical power of the bound in (6.5) is, that it holds for *any* selectors $\mathcal{S}_1, \ldots, \mathcal{S}_m$ which satisfy (6.2). In particular, the bound also holds for sequences $R^m f$ which depend on the sample size n (since $X \sim P = P_n$ and also the function of interest $f = f_n$ depend on n).

6.2. *Asymptotic analysis as sample size increases.* The L_2Boosting algorithm can be represented analogously to (6.1). We introduce the following notation:

$$\langle f, g\rangle_{(n)} = n^{-1}\sum_{i=1}^n f(X_i)g(X_i), \text{ and } \|f\|^2_{(n)} = n^{-1}\sum_{i=1}^n f(X_i)^2$$

for functions $f, g : \mathbb{R}^{p_n} \to \mathbb{R}$. As before, we denote by $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ the vector of response variables.

Define

$$\begin{aligned}
&\hat{R}_n^0 f = f, \quad \hat{R}_n^1 f = f - \left\langle \mathbf{Y}, g_{\hat{\mathcal{S}}_1}\right\rangle_{(n)} g_{\hat{\mathcal{S}}_1}, \\
&\hat{R}_n^m f = \hat{R}_n^{m-1} f - \left\langle \hat{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m}\right\rangle_{(n)} g_{\hat{\mathcal{S}}_m}, \ m = 2, 3, \ldots,
\end{aligned}$$

where

$$\begin{aligned}
&\hat{\mathcal{S}}_1 = \operatorname*{arg\,max}_{1\le j\le p_n} |\langle \mathbf{Y}, g_j\rangle_{(n)}|, \\
&\hat{\mathcal{S}}_m = \operatorname*{arg\,max}_{1\le j\le p_n} \left|\left\langle \hat{R}_n^{m-1} f, g_j\right\rangle_{(n)}\right|, \ m = 2, 3, \ldots
\end{aligned}$$

By definition, $\hat{R}_n^m f = f - \hat{F}_n^m$ is the difference of the function f and its L_2Boosting estimate $\hat{F}_n^m$. Note that we emphasize here the dependence of $\hat{R}_n^m$ on n since finite-sample estimates $\left\langle \hat{R}_n^{m-1} f, g_j\right\rangle_{(n)}$ are involved.

6.2.1. *A semi-population version.* For analyzing $\hat{R}_n^m f$, we want to use Temlyakov's (2000) result from (6.5). We will apply it to a semi-population version $\tilde{R}_n^m f$, as defined below (since it seems difficult to establish (6.2) for $\hat{R}_n^m f$ directly).

Consider

$$\begin{aligned}
&\tilde{R}_n^0 f = f, \\
&\tilde{R}_n^m f = \tilde{R}_n^{m-1} f - \left\langle \tilde{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m}\right\rangle g_{\hat{\mathcal{S}}_m}, \ m = 1, 2, \ldots
\end{aligned}$$

where $\hat{\mathcal{S}}_m$ is the selector from the sample version above.

The strategy will be as follows. First, we want to establish a finite-sample analogue of (6.2) for the estimated selectors $\hat{\mathcal{S}}_m$: this will then allow us to use Temlyakov's (2000) result from (6.5) for $\tilde{R}_n^m f$. Finally, we need to analyze the difference $\hat{R}_n^m f - \tilde{R}_n^m f$.

6.2.2. *Uniform laws of large numbers.*

LEMMA 1. *Under the assumptions (A1)-(A4), with* $0 < \xi < 1$ *as in (A1),*

(i) $\sup_{1\le j,k\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)g_k(X_i) - \mathbb{E}[g_j(X)g_k(X)]| = \zeta_{n,1} = O_P(n^{-\xi/2})$,
(ii) $\sup_{1\le j\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)\varepsilon_i| = \zeta_{n,2} = O_P(n^{-\xi/2})$,
(iii) $\sup_{1\le j\le p_n} |n^{-1}\sum_{i=1}^n f(X_i)g_j(X_i) - \mathbb{E}[f(X)g_j(X)]| = \zeta_{n,3} = O_P(n^{-\xi/2})$,
(iv) $\sup_{1\le j\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)Y_i - \mathbb{E}[g_j(X)Y]| = \zeta_{n,4} = O_P(n^{-\xi/2})$,

Proof: For assertion (i), denote by $M = \sup_j \|g_j(X)\|_\infty$, see assumption (A3). Then, Bernstein's inequality yields for every $\gamma > 0$,

$$\begin{aligned} & \mathbb{P}[n^{\xi/2} \sup_{1\le j,k\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)g_k(X_i) - \mathbb{E}[g_j(X)g_k(X)]| > \gamma] \\ \le\ & p_n^2 2\exp\left(-\frac{\gamma^2 n^{1-\xi}}{2(\sigma_g^2 + M^2\gamma n^{-\xi/2})}\right), \end{aligned}$$

where σ_g^2 is an upper bound for $Var(g_j(X)g_k(X))$ for all j,k (e.g. $\sigma_g^2 = M^4$). Since $p_n^2 = O(\exp(2C(n^{1-\xi})))$, the right-hand side of the inequality above becomes arbitrarily small for n sufficiently large and $\gamma > 0$ large.

For proving assertion (ii), we have to deal with the unboundedness of the ε_i's in order to apply Bernstein's inequality. Define the truncated variables

$$\varepsilon_i^{tr} = \begin{cases} \varepsilon_i, & \text{if } |\varepsilon_i| \le M_n \\ \text{sign}(\varepsilon_i)M_n, & \text{if } |\varepsilon_i| > M_n. \end{cases}$$

Then, for $\gamma > 0$,

$$\begin{aligned} & \mathbb{P}[n^{\xi/2} \sup_{1\le j\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)\varepsilon_i| > \gamma] \\ \le\ & \mathbb{P}[n^{\xi/2} \sup_{1\le j\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)\varepsilon_i^{tr} - \mathbb{E}[g_j(X)\varepsilon^{tr}]| > \gamma/3] \\ +\ & \mathbb{P}[n^{\xi/2} \sup_{1\le j\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)(\varepsilon_i - \varepsilon_i^{tr})| > \gamma/3] \\ +\ & \mathbb{P}[n^{\xi/2} \sup_{1\le j\le p_n} |n^{-1}\sum_{i=1}^n \mathbb{E}[g_j(X_i)(\varepsilon_i - \varepsilon_i^{tr})]| > \gamma/3] \\ =\ & I + II + III, \end{aligned}$$

since $\mathbb{E}[g_j(X)\varepsilon] = \mathbb{E}[g_j(X)]\mathbb{E}[\varepsilon] = 0$ which we use for III. We can bound I again by using Bernstein's inequality:

$$I \le p_n 2 \exp\left(-\frac{(\gamma^2/9)n^{1-\xi}}{2(\sigma_g^2 + M_n^2(\gamma/3)n^{-\xi/2})}\right), \tag{6.6}$$

where σ_g^2 is an upper bound for $Var(g_j(X)\varepsilon^{tr})$ (e.g. $\sup_j \|g_j(X)\|_\infty^2 \mathbb{E}|\varepsilon|^2$). When using

$$M_n = n^{\xi/4},$$

we can make the right hand since in (6.6) arbitrarily small since $p_n = O(\exp(Cn^{1-\xi}))$: thus, for every $\delta > 0$,

$$I \le \delta \text{ for } n \text{ sufficiently large, } \gamma \text{ sufficiently large.} \tag{6.7}$$

A bound for II can be obtained as follows:

$$\begin{aligned} II &\le \mathbb{P}[\text{some } |\varepsilon_i| > M_n] \le n\mathbb{P}[|\varepsilon| > M_n] \le nM_n^{-s}\mathbb{E}|\varepsilon|^s \\ &= O(n^{1-s\xi/4}) = o(1) \ (n \to \infty) \end{aligned} \tag{6.8}$$

since $s > 4/\xi$ by assumption (A4).

For III we use the bound

$$III \le \mathbb{1}_{[n^{\xi/2}\sup_j |\mathbb{E}[g_j(X)(\varepsilon - \varepsilon^{tr})]| > \gamma/3]}. \tag{6.9}$$

Note that by the independence of ε (and ε^{tr}) from $g_j(X)$,

$$\mathbb{E}[g_j(X)(\varepsilon - \varepsilon^{tr})] = \mathbb{E}[g_j(X)]\mathbb{E}[\varepsilon - \varepsilon^{tr}].$$

Hence, an upper bound is

$$|\mathbb{E}[g_j(X)(\varepsilon - \varepsilon^{tr})]| \le M|\mathbb{E}[\varepsilon - \varepsilon^{tr}]|.$$

The latter can be bounded as

$$\begin{aligned} &|\mathbb{E}[\varepsilon - \varepsilon^{tr}]| \le |\int_{|x|>M_n} (\text{sign}(x)M_n - x)dP_\varepsilon(x)| \\ \le\ & \int \mathbb{1}_{[|x|>M_n]}(M_n + |x|)dP_\varepsilon(x) \\ =\ & M_n\mathbb{P}[|\varepsilon| > M_n] + \int |x|\mathbb{1}_{[|x|>M_n]}dP_\varepsilon(x) \\ \le\ & M_n^{1-s}\mathbb{E}|\varepsilon|^s + (\mathbb{E}|\varepsilon|^2)^{1/2}(\mathbb{P}[|\varepsilon| > M_n])^{1/2} \\ =\ & O(M_n^{1-s}) + O(M_n^{-s/2}) = o(M_n^{-2}) = o(n^{-\xi/2}) \end{aligned}$$

since $s > 4/\xi > 4$ $(0 < \xi < 1)$. Hence, by using (6.9):

$$III = 0 \text{ for } n \text{ sufficiently large, } \gamma > 0 \text{ sufficiently large,}$$

and together with (6.7) and (6.8), this proves assertion (ii).

Assertion (iii) follows from (i):

$$\begin{aligned}
&\sup_{1\le j\le p_n, n\in\mathbb{N}} |n^{-1}\sum_{i=1}^n f(X_i)g_j(X_i) - \mathbb{E}[f(X)g_j(X)]| \\
\le\ & \sum_{r=1}^{p_n} |\beta_{r,n}| \sup_{1\le j,k\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)g_k(X_i) - \mathbb{E}[g_j(X)g_k(X)]| \\
\le\ & \sum_{r=1}^{p_n} |\beta_{r,n}| \sup_{1\le j,k\le p_n} |n^{-1}\sum_{i=1}^n g_j(X_i)g_k(X_i) - \mathbb{E}[g_j(X)g_k(X)]| \\
\le\ & \sum_{r=1}^{p_n} |\beta_{r,n}|\zeta_{n,1} = O_P(n^{-\xi/2}).
\end{aligned}$$

Assertion (iv) follows from (ii) and (iii). □

6.2.3. *Recursive analysis of L_2Boosting.* Denote by

$$\zeta_n = \max\{\zeta_{n,1}, \zeta_{n,2}, \zeta_{n,3}, \zeta_{n,4}\} = O_P(n^{-\xi/2})$$

which is a bound for all assertions (i)-(iv) in Lemma 1. Also, we denote by ω a realization of all n data-points.

LEMMA 2. *Under the assumptions of Lemma 1, there exists a constant $0 < C_* < \infty$, independent from n and m, such that*

$$\begin{aligned}
&\sup_{1\le j\le p_n} |\left\langle \hat{R}_n^m f, g_j\right\rangle_{(n)} - \left\langle \tilde{R}_n^m f, g_j\right\rangle| \le (5/2)^m \zeta_n C_* \\
&\text{on the set } A_n = \{\omega;\ |\zeta_n(\omega)| < 1/2\}.
\end{aligned}$$

Note that Lemma 1 implies that $\mathbb{P}[A_n] \to 1$ $(n \to \infty)$. The constant C_* is depending on $\sup_{n\in\mathbb{N}} \sum_{j=1}^{p_n} |\beta_{j,n}|$.

Proof: We proceed recursively. For $m = 0$, the statement follows directly from Lemma 1(iv). Denote by $A_n(m,j) = \left\langle \hat{R}_n^m f, g_j\right\rangle_{(n)} - \left\langle \tilde{R}_n^m f, g_j\right\rangle$. Then, by definition,

$$\begin{aligned}
A_n(m,j) &= A_n(m-1,j) - \left\langle \tilde{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m}\right\rangle \left(\left\langle g_{\hat{\mathcal{S}}_m}, g_j\right\rangle_{(n)} - \left\langle g_{\hat{\mathcal{S}}_m}, g_j\right\rangle\right) \\
&\quad - \left\langle g_{\hat{\mathcal{S}}_m}, g_j\right\rangle_{(n)} \left(\left\langle \hat{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m}\right\rangle_{(n)} - \left\langle \tilde{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m}\right\rangle\right) \\
&= A_n(m-1,j) - I_{n,m}(j) - II_{n,m}(j). \qquad (6.10)
\end{aligned}$$

From Lemma 1(i) we get,

$$\sup_{1\le j\le p_n} |I_{n,m}(j)| \le \|\tilde{R}_n^{m-1} f\| \|g_{\hat{\mathcal{S}}_m}\| \zeta_n \le \|f\| \zeta_n, \tag{6.11}$$

where we have used the norm-reducing property in (6.3) for $\tilde{R}_n^m f$.
For the second term we proceed recursively,

$$\begin{aligned} \sup_{1\le j\le p_n} |II_{n,m}(j)| &\le \sup_{1\le j\le p_n} |\left\langle g_{\hat{\mathcal{S}}_m}, g_j \right\rangle_{(n)}| \sup_{1\le j\le p_n} |A_n(m-1,j)| \\ &\le (1+\zeta_n) \sup_{1\le j\le p_n} |A_n(m-1,j)|. \end{aligned} \tag{6.12}$$

For the last inequality, we have used again Lemma 1(i) and the Cauchy-Schwarz inequality $|\left\langle g_{\hat{\mathcal{S}}_m}, g_j \right\rangle| \le \|g_{\hat{\mathcal{S}}_m}\| \|g_j\| = 1$.
Using the notation $B_n(m) = \sup_{1\le j\le p_n} |A_n(m,j)|$, we get the following recursion from (6.10)-(6.12):

$$\begin{aligned} &B_n(0) \le \zeta_n, \\ &B_n(m) \le B_n(m-1) + \zeta_n \|f\| + (1+\zeta_n) B_n(m-1) \\ &\qquad \le (5/2) B_n(m-1) + \zeta_n \|f\| \text{ on the set } A_n. \end{aligned}$$

Therefore,

$$\begin{aligned} B_n(m) &\le (5/2)^m \zeta_n + \zeta_n \|f\| \sum_{j=0}^{m-1} (5/2)^j \le (5/2)^m \zeta_n (1 + \|f\| \sum_{j=0}^{m-1} (5/2)^{j-m}) \\ &\le (5/2)^m \zeta_n (1 + \sup_{n\in\mathbb{N}} \sum_{j=1}^{p_n} |\beta_{j,n}| \sum_{k=1}^{\infty} (5/2)^{-k}), \end{aligned}$$

which completes the proof by setting $C_* = 1 + \sup_{n\in\mathbb{N}} \sum_{j=1}^{p_n} |\beta_{j,n}| \sum_{k=1}^{\infty} (5/2)^{-k}$.
□

Analysing $\tilde{R}_n^m f$.
We are now ready to establish a finite-sample analogue of (6.2) for $\tilde{R}_n^m f$.
We have

$$\left\langle \hat{R}_n^m f, g_j \right\rangle_{(n)} = \left\langle \tilde{R}_n^m f, g_j \right\rangle + \left(\left\langle \hat{R}_n^m f, g_j \right\rangle_{(n)} - \left\langle \tilde{R}_n^m f, g_j \right\rangle \right).$$

Hence, by invoking Lemma 2 (and denoting by A_n the set as there) we get

$$\begin{aligned} |\left\langle \hat{R}_n^m f, g_{\hat{\mathcal{S}}_m} \right\rangle_{(n)}| &= \sup_{1\le j\le p_n} |\left\langle \hat{R}_n^m f, g_j \right\rangle_{(n)}| \\ &\ge \sup_{1\le j\le p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| - (5/2)^m \zeta_n C_* \text{ on the set } A_n. \end{aligned}$$

Therefore, again by Lemma 2 for the first inequality to follow,

$$\begin{aligned} |\left\langle \tilde{R}_n^m f, g_{\hat{\mathcal{S}}_m} \right\rangle| &\geq |\left\langle \hat{R}_n^m f, g_{\hat{\mathcal{S}}_m} \right\rangle_{(n)}| - (5/2)^m \zeta_n C_* \text{ on the set } A_n \\ &\geq \sup_{1\leq j\leq p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| - 2(5/2)^m \zeta_n C_* \text{ on the set } A_n. \end{aligned} \tag{6.13}$$

Consider the set $B_n = \{\omega;\ \sup_{1\leq j\leq p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| > 4(5/2)^m \zeta_n C_*\}$. Then, by (6.13),

$$|\left\langle \tilde{R}_n^m f, g_{\hat{\mathcal{S}}_m} \right\rangle| \geq 0.5 \sup_{1\leq j\leq p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| \text{ on the set } A_n \cap B_n. \tag{6.14}$$

Formula (6.14) says that the selectors $\hat{\mathcal{S}}_m$ satisfy the condition (6.2) for $\tilde{R}_n^m f$ on the set $A_n \cap B_n$. We can now invoke Temlyakov's result in (6.5), since the condition (6.2) holds on the set $A_n \cap B_n$ (as established in (6.14)),

$$\|\tilde{R}_n^m f\| \leq B(1 + m/4)^{-1/10} = o(1) \text{ on the set } A_n \cap B_n \tag{6.15}$$

by choosing $m = m_n \to \infty$ $(n \to \infty)$ (slow enough), where $B = \sup_{n\in\mathbb{N}} \sum_{j=1}^{p_n} |\beta_{j,n}| < \infty$, cf. (6.4) and assumption (A2).

For $\omega \in B_n^C = \{\omega;\ \sup_{1\leq j\leq p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| \leq 4(5/2)^m \zeta_n C_*\}$, by using formula (5.2) from Temlyakov (2000) with b_m as defined there (i.e. $b_m = b_{m-1} + |\left\langle \tilde{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m} \right\rangle|, b_0 = 1$),

$$\begin{aligned} \|\tilde{R}_n^m f\|^2 &\leq \sup_{1\leq j\leq p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| b_m \leq \sup_{1\leq j\leq p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| (1 + m\|f\|) \\ &\leq 4(5/2)^m \zeta_n C_* (1 + m\|f\|) \text{ on the set } B_n^C. \end{aligned} \tag{6.16}$$

For bounding the number b_m, we have used the norm reducing property in (6.3) applied to $\tilde{R}_n^m f$. Therefore, using (6.15), (6.16) and $\zeta_n = O_P(n^{-\xi/2})$ from Lemma 1, we have for $m = m_n \to \infty (n \to \infty)$ slow enough (e.g. $m_n = o(\log(n))$),

$$\begin{aligned} \|\tilde{R}_n^m f\| &\leq B(1 + m_n/4)^{-1/10} + 4(5/2)^{m_n} \zeta_n C_* (1 + m\|f\|) \\ &\quad \text{on the set } (A_n \cap B_n) \cup B_n^C \\ &= o_P(1), \end{aligned} \tag{6.17}$$

since $\mathbb{P}[(A_n \cap B_n) \cup B_n^C] \geq \mathbb{P}[A_n] \to 1$ $(n \to \infty)$ due to Lemma 1.

Analysing $\hat{R}_n^m f$.
By definition and using the triangle inequality

$$\|\hat{F}_n^m - f\| = \|\hat{R}_n^m f\| \leq \|\tilde{R}_n^m f\| + \|\hat{R}_n^m f - \tilde{R}_n^m f\|. \tag{6.18}$$

A recursive analysis can be developed for the second term on the right hand side

$$A_n(m) = \|\hat{R}_n^m f - \tilde{R}_n^m f\|.$$

By definition,

$$\begin{aligned} A_n(m) &= \|\hat{R}_n^{m-1} f - \tilde{R}_n^{m-1} f - \left(\left\langle \hat{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m} \right\rangle_{(n)} - \left\langle \tilde{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m} \right\rangle \right) g_{\hat{\mathcal{S}}_m} \| \\ &\leq A_n(m-1) + |\left\langle \hat{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m} \right\rangle_{(n)} - \left\langle \tilde{R}_n^{m-1} f, g_{\hat{\mathcal{S}}_m} \right\rangle| \|g_{\hat{\mathcal{S}}_m}\| \\ &\leq A_n(m-1) + (5/2)^{m-1} \zeta_n C_* \text{ on the set } A_n, \end{aligned}$$

where the last inequality follows from Lemma 2. Therefore, for some constant $C > 0$,

$$\|\hat{R}_n^m f - \tilde{R}_n^m f\| \leq 3^m \zeta_n C = o_P(1) \tag{6.19}$$

by choosing $m = m_n \to \infty$ sufficiently slowly such that $3^{m_n} \zeta_n = o_P(1)$.

By (6.18), (6.17) and (6.19) we get (e.g. by using the choice $m_n \to \infty, m_n = o(\log(n))$),

$$\mathbb{E}_X |\hat{F}_n^{m_n}(X) - f(X)|^2 = \|\hat{R}_n^{m_n} f\|^2 = o_P(1)$$

which completes the proof of Theorem 1. □

6.3. *Arbitrary step-size ν.* For arbitrary, fixed step-size $0 < \nu \leq 1$ in (2.2), we need to make a few modifications of the proof.

Temlyakov's result in (6.5) becomes

$$\|R^m f\| \leq B(1 + \nu(2-\nu)mb^2)^{-b/(2(2+b))}, \ 0 < b \leq 1 \text{ as in (6.2)}.$$

Proof: The claim follows as in Temlyakov (2000). Using his notation, we use $a_m = \|R^m f\|^2$, $y_m = |\left\langle R^{m-1} f, g_{\mathcal{S}_m} \right\rangle|$, $b_m = b_{m-1} + \nu y_m$, $b_0 = 1$ and $t_m \equiv b$ from (6.2). □

We can then use exactly the same reasoning as in section 6.2. At some obvious places, a factor ν occurs in addition which can be trivially bounded by 1. The only slightly non-trivial reasoning occurs in (6.16): but using b_m as defined above (applied now to $\tilde{R}_n^m f$ instead of $R^m f$) yields the bound

$$\|\tilde{R}_n^m f\|^2 \leq \sup_{1 \leq j \leq p_n} |\left\langle \tilde{R}_n^m f, g_j \right\rangle| b_m,$$

which then allows to proceed as in section 6.2.

6.4. *Binary classification.* The first assertion of Corollary 1 follows exactly as in the proof of Theorem 1 by using the representation in 3.2. There is no crucial place where we make use of homoscedastic errors ε_i: the uniform laws of large numbers from Lemma 1 look formally a bit different (e.g. for (ii) we need to subtract a term $\mathbb{E}[g_j(X_i)\varepsilon_i] = \mathbb{E}[g_j(X_i)(Y_i - f(X_i))]$), but the i.i.d. structure of the pairs (X_i, Y_i) suffices to get through. The moment assumption for $\varepsilon_i = Y_i - f(X_i)$ trivially holds since $|Y_i| \leq 1$ and $\sup_x |f(x)| \leq 1$.

For the second assertion, it is well-known that 2 times the L_1-norm bounds from above the difference between the generalization error of a plug-in classifier (expected 0-1 loss error for classifying a new observation) and the Bayes risk (Devroye et al., 1996, Theorem 2.3). Furthermore, the L_1-norm is upper bounded by the L_2-norm. □

Appendix: The model (4.4). The model (4.4) is as follows. Define $a_j = j^{0.51}$. Let the parameter κ be the solution of the equation $\sigma_\varepsilon^2 n^{-1} \sum_{j=1}^{\infty} a_j \lambda_j = \kappa$, where we denote by $\lambda_j = (1 - \kappa a_j)_+$. For $n = 100$, the solution is $\kappa = 0.199$. Determine the predictor dimension $p = \max_j\{a_j \leq \kappa^{-1}\} = 23$. The variances are

$$\sigma_j^2 = \lambda_j (n\kappa a_j)^{-1},\ j = 1, \ldots, 23,\ n = 100.$$

It can be shown that such regression coefficients belong with high probability to $\{(\beta_{j,n})_j;\ \sum_{j=1}^{p_n} a_j^2 \beta_{j,n}^2 \leq 1\}$ (note that $p = p_n$ depends on n via the parameter $\kappa = \kappa_n$).

Acknowledgments: I thank two referees and the editor M. Eaton for constructive comments.

References.

Breiman, L. (1998). Arcing classifiers (with discussion). *Annals of Statistics* **26** 801–849.

Breiman, L. (1999). Prediction games & arcing algorithms. *Neural Computation* **11** 1493–1517.

Bühlmann and Yu, B. (2005). Boosting, model, selection, lasso and nonnegative garrote. Tech. rep., `ftp://ftp.stat.math.ethz.ch/Research-Reports/Other-Manuscripts/buhlmann/boost-garrote-otherf.pdf`.

Bühlmann, P. and Yu, B. (2003). Boosting with the $l_2$loss: regression and classification. *Journal of the American Statistical Association* **98** 324–339.

CHEN, S., DONOHO, D. and SAUNDERS, M. (1999). Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing* **20** 33–61.

CRAN (1997 ff.). *The Comprehensive R Archive Network*. `http://cran.R-project`.

DETTLING, M. and BÜHLMANN, P. (2004). Finding predictive gene groups from microarray data. *Journal of Multivariate Analysis* **90** 106–131.

DEVROYE, L., GYÖRFI, L. and LUGOSI, G. (1996). *A probabilistic theory of pattern recognition*. Springer.

DUDOIT, S., FRIDLYAND, J. and SPEED, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* **97** 77–87.

EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression (with discussion). *Annals of Statistics* **32** 407–451.

FREUND, Y. and SCHAPIRE, R. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proc. Thirteenth International Conference*. Morgan Kauffman.

FRIEDMAN, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics* **29** 1189–1232.

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion). *Annals of Statistics* **28** 337–407.

GOLDENSHLUGER, A. and TSYBAKOV, A. (2001). Adaptive prediction and estimation in linear regression with infinitely many parameters. *Annals of Statistics* **29** 1601–1619.

GREENSHTEIN, E. and RITOV, Y. (2004). Persistence in high-dimensional predictor selection and the virtue of over-parametrization. *Bernoulli* **10** 971–988.

HURVICH, C., SIMONOFF, J. and TSAI, C.-L. (1998). Smoothing parameter selection in nonparametric regression using an improved akaike information criterion. *Journal of the Royal Statistical Society, Series B* **60** 271–293.

JIANG, W. (2004). Process consistency for adaboost (with discussion). *Annals of Statistics* **32** 13–29 (disc. pp. 85–134).

LUGOSI, G. and VAYATIS, N. (2004). On the bayes-risk consistency of regularized boosting methods (with discussion). *Annals of Statistics* **32** 30–55 (disc. pp. 85–134).

MALLAT, S. and ZHANG, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing* **41** 3397–3415.

MANNOR, S., MEIR, R. and ZHANG, T. (2002). The consistency of greedy algorithms for classification. In *Computational Learning Theory: 15th Annual Conference on Computational Learning Theory, COLT 2002, Proceedings* (J. Kivinen and R. Sloan, eds.), vol. 2375 of *Lecture Notes in Computer Science*. Springer.

SCHAPIRE, R. (2002). The boosting approach to machine learning: an overview. In *MSRI Workshop on Nonlinear Estimation and Classification* (D. Denison, M. Hansen, C. Holmes, B. Mallick and B. Yu, eds.). Springer.

TEMLYAKOV, V. (2000). Weak greedy algorithms. *Advances in Computational Mathematics* **12** 213–227.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistica;l Society, Series B* **58** 267–288.

TUKEY, J. (1977). *Exploratory data analysis*. Addison-Wesley.

WEST, M., BLANCHETTE, C., DRESSMAN, H., HUANG, E., ISHIDA, S., SPANG, R., ZUZAN, H., OLSON, J., MARKS, J. and NEVINS, J. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences, USA* **98** 11462–11467.

ZHANG, T. and YU, B. (2003). Boosting with early stopping: convergence and consistency. *To appear in the Annals of Statistics* .

ZOU, H. and HASTIE, T. (2003). Regularization and variable selection via the elastic net. *To appear in the Journal of the Royal Statistical Society, Series B* .

SEMINAR FÜR STATISTIK, ETH ZÜRICH
CH-8092 ZÜRICH
SWITZERLAND

E-MAIL: buhlmann@stat.math.ethz.ch