Gerhard Tutz & Jan Gertheiss

Feature Extraction in Signal Regression: A Boosting Technique for Functional Data Regression

Technical Report Number 011, 2007
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Feature Extraction in Signal Regression: A Boosting Technique for Functional Data Regression

Gerhard Tutz & Jan Gertheiss
Ludwig-Maximilians-Universität München
Akademiestraße 1, 80799 München
{tutz,jan.gertheiss}@stat.uni-muenchen.de

December 20, 2007

Abstract

Main objectives of feature extraction in signal regression are the improvement of accuracy of prediction on future data and identification of relevant parts of the signal. A feature extraction procedure is proposed that uses boosting techniques to select the relevant parts of the signal. The proposed blockwise boosting procedure simultaneously selects intervals in the signal's domain and estimates the effect on the response. The blocks that are defined explicitly use the underlying metric of the signal. It is demonstrated in simulation studies and for real-world data that the proposed approach competes well with procedures like PLS, P-spline signal regression and functional data regression.

Keywords: Signal Regression, Boosting techniques, Generalized Ridge Regression, P-Splines, Partial Least Squares

1 Introduction

Signal regression has been extensively studied in the chemometrics community. An excellent summary of tools has been given by Frank and Friedman (1993). With the recent surge of interest in functional data, signal regression may be embedded into the framework of functional data, nicely outlined by Ramsay and Silverman (2005). If functional data like signals are used as regressors the main problem is the large number of predictors which makes common least-squares techniques inapplicable. Each experimental unit usually generates a number of regressors which far exceeds the number of units collected in the study. Figure 1 shows signal regressors from near infrared spectroscopy applied to a compositional analysis of 32 marzipan samples (Christensen et al., 2004). Each "signal" consists of 600 digitizations along the wavelength axis. The objective of the analysis is to determine moisture and sugar content from these signals (for details see subsection 4.2).

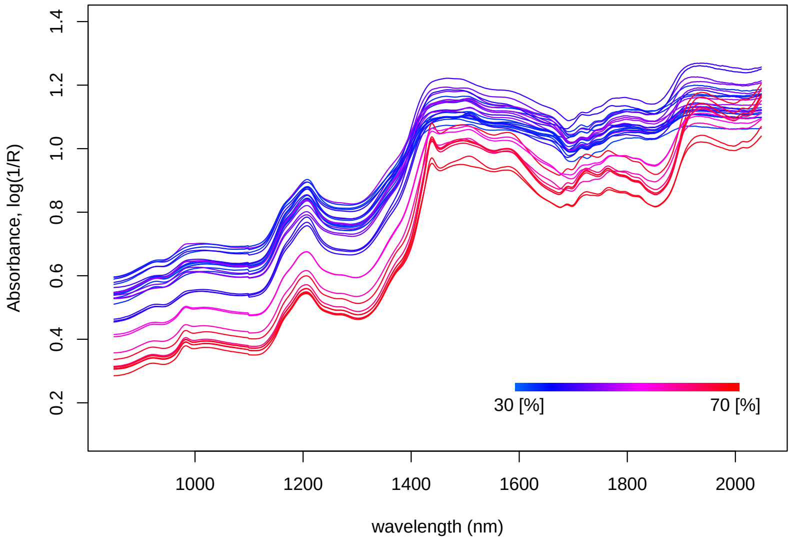


Figure 1: NIR spectra of 32 marzipan samples; colors corresponding to sugar content.

Objectives in functional regression are manifold, our specific concern is on two aspects, accuracy of prediction on future data and feature extraction. When the main concern is prediction, feature extraction is secondary but may serve the purpose to obtain better prediction performance. In other cases feature extraction is of interest from the viewpoint of interpretability. One wants to know which predictors effect upon the response, for the spectroscopy data that means which areas of wavelength are relevant. As a second example we will use the rainfall data from Ramsay and Silverman (2005). Figure 2 shows the temperature profiles (in degrees celcius) of Canadian weather stations across the year - averaged over 1960 to 1994. As Ramsay and Silverman we will consider the base 10 logarithm of the total annual precipitation as response variable.

In this article, we propose a method of feature extraction which focuses on groups of adjacent variables. By using boosting techniques we select subsets of

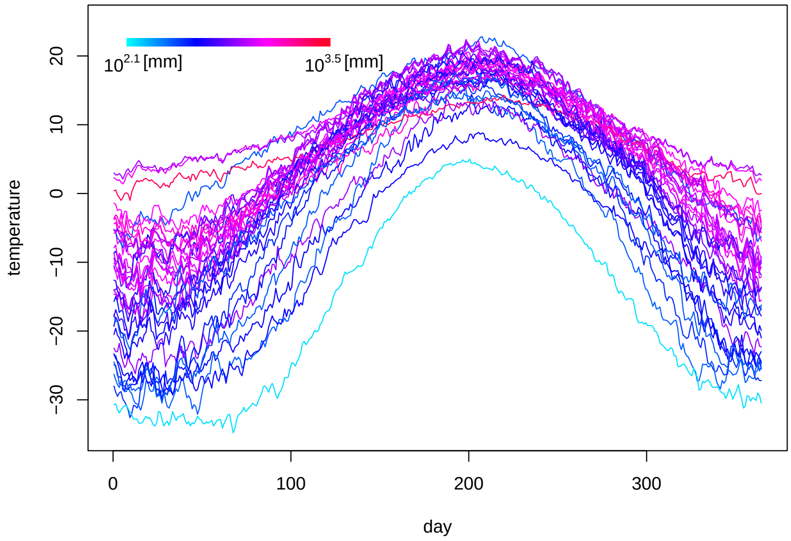


Figure 2: Temperature profiles of 35 Canadian weather stations; colors corresponding to (log) total annual precipitation.

predictors, whose coefficients are interpretable. Moreover, it is demonstrated that the selection of groups of predictors improves the accuracy of prediction when compared to alternative procedures.

There is a whole range of methods that applies to signal regression functional data. Classical instruments are partial least squares (PLS) and principal-component regression (PCR). More recently developed tools aim at constraining the coefficient vector to be a smooth function; see Hastie and Mallows (1993), Marx and Eilers (1999), Marx and Eilers (2005). But also the much older ridge regression (Hoerl and Kennard, 1970), the new elastic net (Zou and Hastie, 2005) and the fused lasso (Tibshirani et al., 2005) are able to handle highdimensional predictor spaces. In addition to these parametric approaches we will also consider random forests (Breiman, 2001), which on various occasions have turned out to be highly efficient in terms of prediction.

A first illustration of the difference between methods is given in Figure 3, where the coefficient function resulting from lasso, ridge regression, generalized ridge regression with first-difference penalty, P-splines signal regression, functional data approach (Ramsay and Silverman, 2005) and the proposed BlockBoost is shown for the Canadian weather data which has become some sort of benchmark data set. It is seen that lasso selects only few variables - theoretically at most n variables as pointed out by Zou and Hastie (2005). In the considered example selecting only few variables means selecting only few days, whose mean

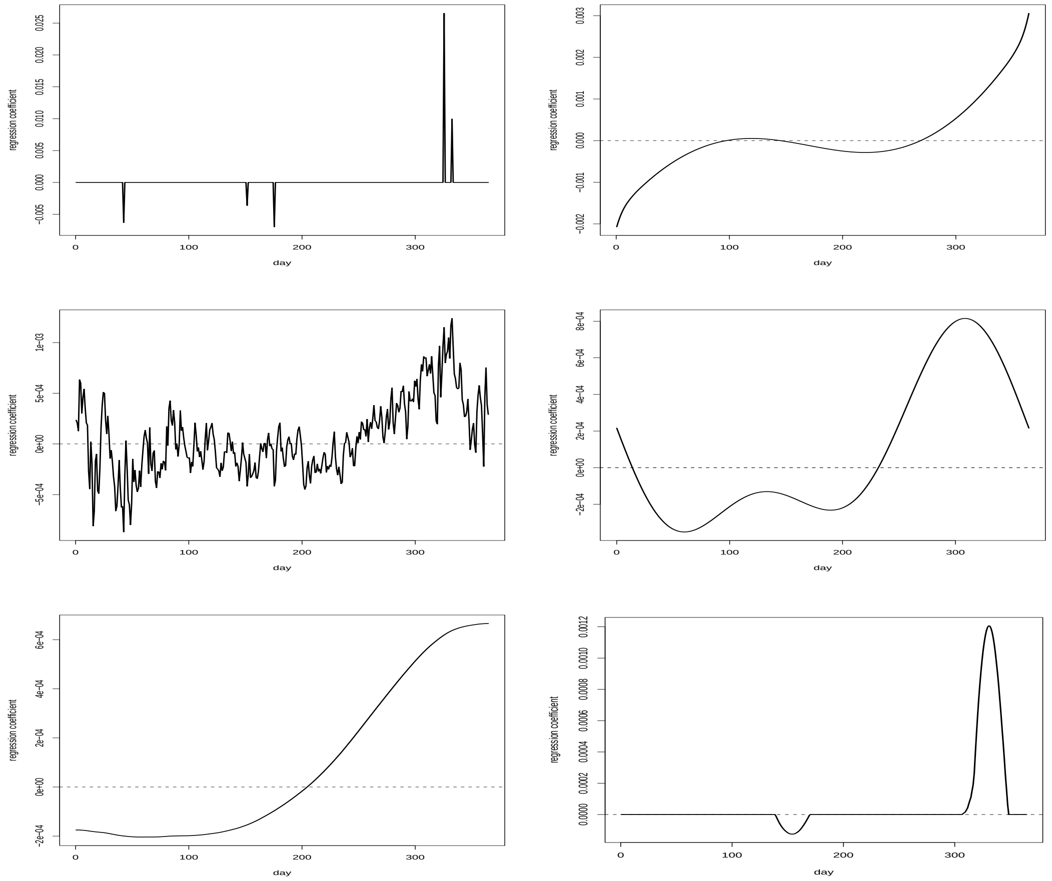


Figure 3: Regression coefficients estimated by various methods; column by column, top down: lasso, ridge, generalized ridge with first-difference penalty, P-splines, functional data approach, BlockBoost; temperature profiles of 35 Canadian weather stations as predictors, log total annual precipitation as response.

temperature is assumed to be relevant for the total annual precipitation. By construction ridge regression takes into account all variables. Smoothing the coefficient function is possible by penalizing differences between adjacent coefficients - or by using smooth basis functions - as proposed by Marx and Eilers (1999), Ramsay and Silverman (2005). But still almost every day's temperature is considered to be important. BlockBoost selects only some periods instead, e.g. some weeks in late autumn / early winter. The smooth BlockBoost estimates basically result from penalizing (first) differences between adjacent coefficients. Details of the procedures are given in 2.2. Nevertheless we want to note that P-splines are based on 35 B-spline basis functions with equally spaced knots and duplicated boundary knots, for details see Hastie et al. (2001). To increase smoothness a third difference penalty is imposed on the basis coefficients, see Marx and Eilers

(1999). By contrast, in the functional data approach Fourier basis functions for smoothing both the functional regressors and the coefficient function have been used, for details see Ramsay and Silverman (2005).

2 Feature Extraction by Blockwise Boosting

2.1 Regularization in Signal Regression

Let the data be given by $(y_i, x_i(t))$, $i = 1, \dots, n$, where y_i is the response variable and $x(t), t \in I$ denotes a function defined on an interval $I \subset \mathbb{R}$, also called the signal. A functional linear model for scalar responses has the form

$$y_i = \beta_0 + \int x_i(t)\beta(t)dt + \varepsilon_i$$

where $\beta(.)$ is a parameter function and ε_i with $E(\varepsilon_i) = 0$ represents a noise variable, cf. Ramsay and Silverman (2005). The naive approach, fitting by least squares frequently yields perfect fit of the data with poor predictive value. A more promising approach is based on regularization with roughness penalties where

$$L(\beta) = \sum_{i=1}^{n} \left\{ (y_i - \beta_0 - \int x_i(s)\beta(s)ds) \right\}^2 + \lambda \int |\beta^{(m)}(t)|^q dt$$

is minimized, with $\beta^{(m)}$ denoting the mth derivative. For $q \to 0$ one obtains a variable selection procedure, if $m = 1$ zero-order variable fusion (Land and Friedman, 1997) results. For $q = 1$ the choice $m = 0$ corresponds to the lasso (Tibshirani, 1996), and $m = 1$ corresponds to first-order variable fusion (Land and Friedman, 1997). The value $q = 2$ yields ridge regression type estimators ($m = 0$) and generalized ridge regression ($m \geq 1$), Ramsay and Silverman (2005) use $m = 2$ as (one type of) roughness measure.

While ridge type estimators with $m \geq 1$ yield smooth functions $\beta(s)$, variable selection methods and the lasso usually reduce the number of predictors. One strength of the lasso is that it shrinks parameters and performs variable selection simultaneously. However, there are strong limitations, since the lasso selects at most n predictors. Variable selection strategies, including the lasso, are "equivariant" methods referring to the fact that they are equivariant to permutations of the predictor indices (Land and Friedman, 1997). In contrast, "spatial" methods regularize by utilizing the spatial nature of the predictor index. In this sense generalized ridge regression and functional data analysis based on second derivatives are spatial methods - but without reducing the predictor space. An alternative spatial method is the more recently proposed fused lasso (Tibshirani et al., 2005). It uses two penalty terms; the first is the usual lasso penalty and the second corresponds to $m = 1$ and $q = 1$, i.e. variable fusion. Thus the second term enforces sparsity that refers to first differences of parameters with the effect that piecewise constant fits are obtained.

2.2 Feature Extraction by Blockwise Boosting

The method proposed here reduces the predictor space by variable selection and simultaneously regularizes the parameter function by utilizing a metric defined on the signal space I. The discretized form of the functional linear model is given by

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i \tag{1}$$

where $x_{ij} = x_i(t_j), \beta_j = \beta(t_j)$ for values $t_1 < \cdots < t_p, t_j \in I$. For simplicity we take the values $t_1, \ldots, t_p$ as equidistant, $t_{j+1} - t_j = \Delta$. For the (original) marzipan data the digitization along the wavelength axis yields $p = 600$, where $\Delta = 2$ nm has been chosen (for details see Section 4.2).

Boosting

One building block of the proposed method is boosting. Boosting has been developed in the machine learning community with the focus on classification problems, see Schapire (1990) or Freund and Schapire (1996). More recently, based on work by Breiman (1998) or Breiman (1999), it has been extended to regression problems by Friedman et al. (2000), Bühlmann and Yu (2003), Bühlmann (2006). In this article we restrict the consideration to a version of the L_2Boost algorithm which essentially is a repeated least squares fitting of residuals.

Let the data be given by $(y_i, x_i), i = 1, \ldots, n$, $x_i^T = (x_{i1}, \ldots, x_{ip})$ and the underlying regression structure be given by $E(y|x) = \eta(x)$. Boosting estimates $\eta(x)$ in an iterative way. Let $\hat{\eta}_r(x)$ be the estimator of $\eta(x)$ in the rth step. In the next step of the algorithm one considers the data $(u_i, x_i), i = 1, \ldots, n$, where

$$u_i = y_i - \hat{\eta}_r(x_i)$$

is the current residual. When estimating the regression of u_i on x_i one employs an estimator $\hat{f}(x, \{u_i, x_i\})$ (a learner in machine learning terminology) which uses data $\{u_i, x_i\}$. Fitting of the regression model for data (u_i, x_i) yields the improved estimate. A short outline of the algorithm is given in the following.

L2Boost

Step 1 (Initialization)

An initial estimate $\hat{\eta}_0(.)$ is obtained by fitting a simple model to data (y_i, x_i), for example the intercept model.

Step 2 (Residual fit)

For $r = 0, 1, 2, \ldots$ compute residuals $u_i = y_i - \hat{\eta}_r(x_i), i = 1, \ldots, n$, and use the learner $\hat{f}(.; \{u_i, x_i\})$ on the data $\{u_i, x_i\}$. The improved fit is obtained by

$$\hat{\eta}_{r+1}(.) = \hat{\eta}_r(.) + \nu \hat{f}(., \{u_i, x_i\})$$

where $\nu \in (0, 1]$ is a shrinkage parameter.

An estimator or learner, in the sense of the preceding algorithm, is composed from two components, the fitting procedure and the structure that is fitted. Bühlmann and Yu (2003) focus on least squares fitting and linear learners like regression splines. By using small ν a weak learner is obtained with superior performance when compared to alternative smooth regression methods.
For the fitting of linear models an attractive tool which implies variable selection is *componentwise boosting*. Let the learner be defined by least squares fitting of single parameters and selection of the component that shows the best fit. That means one uses

$$\hat{f}(x; \{u_i, x_i\}) = \hat{\beta}_{\hat{s}} x_{\hat{s}}$$

where $\hat{\beta}_j = \sum_{i=1}^n u_i x_{ij} / \sum_{i=1}^n x_{ij}^2$ is the least squares fit for the jth component (centered data) and

$$\hat{s} = \arg \min_{1 \leq j \leq p} \sum_{i=1}^n (u_i - \hat{\beta}_j x_{ij})^2$$

is the selection operator that selects the best fit. That means in each step of the algorithm only one coefficient is refitted. Variables that are never selected are not taken into the model. Componentwise boosting may be directly applied to the signal regression model (1). One obtains variables selection but based on an equivariant method.

Blockwise Boosting

The method proposed here differs from componentwise boosting in the way features are selected. Rather than selecting single variables the approach selects groups of variables where the grouping of variables is based on a metric. Since a signal $x_i(.)$ may be seen as a mapping $x_i : I \to \mathbb{R}$ one utilizes that a metric is available on I. A potentially relevant part of the signal $x_i(.)$ may be characterized by $\{x_{i,U}(t) | t \in U(t_0)\}$ where $U(t_0)$ is defined as a neighborhood of $t_0 \in I$, i.e. $U(t_0) = \{t | \|t - t_0\| \leq \delta\}$ for some metric $\|.\|$ on I. For the digitized signal the potentially relevant signal part turns into the group of variables $\{x_{ij} | t_j \in U(t_0)\}$. When the Euclidean metric is used, $U(t_0)$ may be identified as a sub-interval from I.

Blockwise boosting aims at updating groups of adjacent variables $\{x_{ij} | t_j \in U\}$ for alternative sets U. For simplicity in the updating procedure we use subsets

$U_s = U_k(t_s) = [t_s, t_s + (k-1)\Delta], k \in \{1, 2, \dots\}$. Thus for $k = 1$ one obtains the limiting case of single variables $\{x_i(t_1)\}$, $\{x_i(t_2)\}$, for $k = 2$ one gets pairs of variables $\{x_i(t_1), x_i(t_2)\}, \{x_i(t_2), x_i(t_3)\}$, etc. In the following k is considered as fixed and the index k is suppressed.

Let $X^{(s)}$ denote the design matrix of variables from U_s, i.e. $X^{(s)}$ has rows $(x_i(t_s), \dots, x_i(t_{s+k-1})) = (x_{is}, \dots, x_{i,s+k-1})$. An update step of blockwise boosting will be based on estimating the vector $b^{(s)} = (b(t_s), \dots b(t_{s+k-1}))^T$ from data $(u, X^{(s)})$ where $u = (u_1, \dots, u_n)^T$ denotes the current residual. Least squares fitting cannot be recommended, since variables in $X^{(s)}$ tend to be highly correlated. Alternatives are smooth estimates of β_s along the lines of Marx & Eilers or a ridge type estimator. The simple ridge estimator has the form

$$\hat{b}^{(s)} = (X^{(s)T} X^{(s)} + \lambda I)^{-1} X^{(s)T} u$$

where λ is chosen very large in order to obtain a weak learner. Due to the shrinkage properties of the ridge estimator the (additional) shrinkage parameter ν from the L_2Boost algorithm is superfluous and can be set to 1. Indeed, in terms of prediction accuracy, the performance of a blockwise simple ridge estimator was very encouraging. The resulting coefficient function however tends to be quite wiggly. Smoother coefficients can be obtained for example by penalizing differences between adjacent coefficients. But since penalizing differences does not necessarily shrink coefficients, we additionally penalize (the square of) the first and the last coefficient in the considered block U_s. Hence the parameter estimate we use is the generalized ridge estimator

$$\hat{b}^{(s)} = (X^{(s)T} X^{(s)} + \lambda \Omega)^{-1} X^{(s)T} u$$

with the identity matrix from above being replaced by the penalty matrix

$$\Omega = D^T D, \ D = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 1 \\ 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Ω is tridiagonal, with 2s on the diagonal and -1s on the off-diagonals. In addition to overall shrinkage penalizing the coefficients at the boundaries of the blocks yields smoother transitions when two or more blocks (selected in different boosting iterations) are overlapping. In summary for fixed span k and tuning parameter λ the proposed boosting algorithm is as follows.

BlockBoost

Step 1 (Initialization)

For $s = 1, \ldots, p-k+1$ fit a linear model to data $(y, X^{(s)})$ by using generalized ridge regression as weak learner. From the resulting estimates $\hat{b}^{(s)} = (b_s^{(s)}, \ldots, b_{s+k-1}^{(s)})^T = (X^{(s)T}X^{(s)} + \lambda\Omega)^{-1}X^{(s)T}y$ select the best by minimizing the residual sum of squares

$$\hat{s}_0 = \arg \min_{1 \le s \le p-k-1} \|y - X^{(s)}\hat{b}^{(s)}\|^2.$$

Let $\hat{\beta}^{(0)} = (\beta_1^{(0)}, \ldots, \beta_p^{(0)})^T$ be defined by components

$$\hat{\beta}_j^{(0)} = \begin{cases} \hat{b}_j^{(\hat{s}_0)} & t_j \in U_{\hat{s}_0} \\ 0 & \text{otherwise} \end{cases}.$$

Step 2 (Residual fit)

For $r = 0, 1, 2, \ldots$ compute residuals $u_i = y_i - x_i^T\hat{\beta}^{(r)}, i = 1, \ldots, n$ and fit for $s = 1, \ldots, p-k+1$ a linear model to data $(u, X^{(s)})$ where $u^T = (u_1, \ldots, u_n)$. From the resulting ridge type estimates $\hat{b}^{(s)} = (X^{(s)T}X^{(s)} + \lambda\Omega)^{-1}X^{(s)T}u$ choose $\hat{s}_{r+1}$ such that the residual sum of squares is minimized

$$\hat{s}_{r+1} = \arg \min_{1 \le s \le p-k+1} \|u - X^{(s)}\hat{b}^{(s)}\|^2.$$

Let $\hat{\beta}^{(r+1)}$ be defined by components

$$\hat{\beta}_j^{(r+1)} = \begin{cases} \hat{\beta}_j^{(r)} + \hat{b}_j^{(\hat{s}_{r+1})} & t_j \in U_{\hat{s}_{r+1}} \\ \hat{\beta}_j^{(r)} & \text{otherwise} \end{cases}.$$

BlockBoost as a feature extraction method assumes that not the whole signal is relevant for the explanation of the response. It aims at identifying important areas of the signal. But note, though k is fixed and hence k adjacent β-coefficients are updated in every iteration, the finally resulting 'relevant' parts of the signal may have different lengths, since subsets U_{s_r} and U_{s_l}, selected in iteration r and l may overlap. In contrast to smooth methods like P-splines based signal regression, variable selection in the form of area selection is part of the strategy. This has already been illustrated by Figure 3 (bottom right column).

As selection criterion for the next update the algorithm uses the residual sum of squares (RSS). Of course this criterion may be replaced by a cross-validation (CV) or generalized cross-validation (GCV) criterion.

Stopping the boosting iterations and choice of the span k

In boosting procedures a stopping rule is needed to avoid overfitting. We employ the corrected version of AIC (Hurvich et al., 1998) as proposed by Bühlmann (2006). This criterion is based on the boosting hat matrix which maps the response vector y into the space of fitted values. In the rth iteration the boosting hat matrix B_r is defined by (Bühlmann and Yu, 2003)

$$B_r = \sum_{l=0}^{r} H^{(s_l)} \prod_{m=1}^{l} (I - H^{(s_{l-m})}),$$

with ridge type hat matrix

$$H^{(s)} = X^{(s)}(X^{(s)T}X^{(s)} + \lambda\Omega)^{-1}X^{(s)T}$$

and s_l denoting the variable block that has been selected in the lth boosting iteration. It is easy to show that

$$B_r = I - \prod_{m=0}^{r}(I - H^{(s_{r-m})}) = I - (I - H^{(s_r)})(I - H^{(s_{r-1})}) \cdots (I - H^{(s_0)}).$$

The AIC in the rth iteration is defined by (Hurvich et al., 1998)

$$AIC_c(r) = \log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - (B_r y)_i)^2\right) + \frac{1 + \text{trace}(B_r)/n}{1 - (\text{trace}(B_r) + 2)/n}.$$

Given an upper bound R^* for the candidate number of boosting iterations the optimum iteration number M can be estimated by (Bühlmann, 2006)

$$\hat{M} = \text{argmin}_{0 \leq r \leq R^*} AIC_c(r).$$

For selecting the span k we propose a quite simple strategy: select a rough grid K of possible k-values, e.g. $K = \{20, 30, 40, 50\}$. For every $k \in K$ run the BlockBoost algorithm with the stopping rule presented above; choose the k with minimum AIC at the stopping point.

3 Simulations

3.1 An Illustration

Before comparing the blockwise boosting approach with competing methods we illustrate the performance in a small simulation study. Let the (digitized) signals be generated by

$$x_i(t) = \sum_{k=1}^{5}(b_{ik}\sin(t\pi(5 - b_{ik})/150) - m_{ik}) + 15, \qquad (2)$$

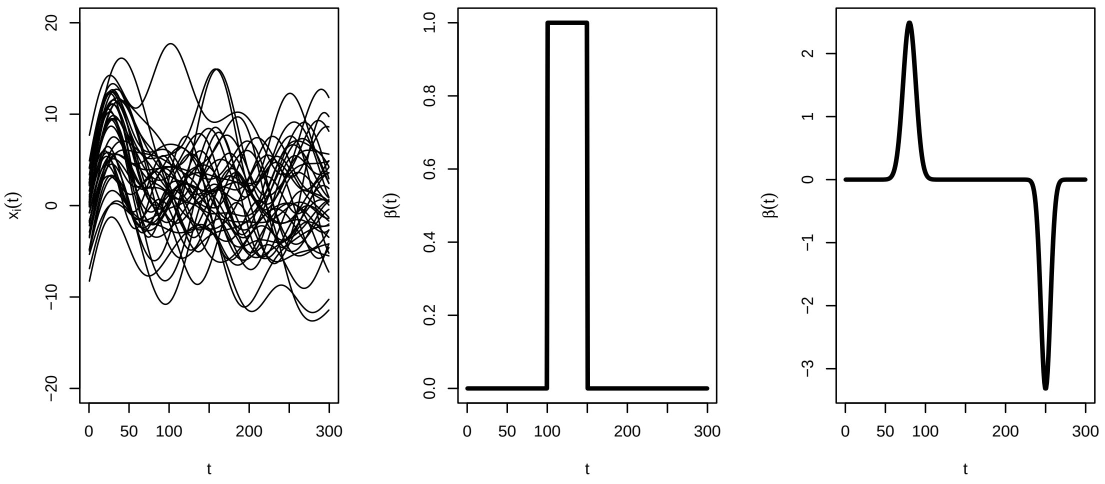


Figure 4: Illustration of the simulation study; from left to right: A set of 40 generated signals, the plateau function $\beta(t) = I_{(100,150)}(t)$ and a normal-mixture function.

$i = 1, \dots, n$, where $t \in (0, 300), b_{ik} \sim U(0, 5), m_{ik} \sim U(0, 2\pi)$ with $U(a, b)$ denoting the uniform distribution on interval $[a, b]$. The signals are observed at equidistant points $t_j \in (0, 300)$, $j = 1, \dots, 300$.

The left panel of Figure 4 shows $n = 40$ generated signals. We consider two parameter functions, the first is the simple plateau function

$$\beta(t) = I_{(100,150)}(t),$$

which is constant on interval $(100, 150)$ and should be hard to fit by smooth updates (see Figure 4, middle). So in this special case for illustration we consider the above mentioned simple ridge blockwise estimator. The second function is given by

$$\beta(t) = \frac{50}{\sqrt{2\pi}} \left(\frac{1}{8} \exp\left(-\frac{1}{2} \left(\frac{t-80}{8} \right)^2 \right) - \frac{1}{6} \exp\left(-\frac{1}{2} \left(\frac{t-250}{6} \right)^2 \right) \right),$$

which is also shown in Figure 4 (right panel) and fitted using the proposed penalty matrix Ω. The response is computed according to the functional linear model (1) with $\varepsilon_i \sim N(0, 30^2)$. In addition, measurement error $\tau_{ij} \sim N(0, 0.25^2)$ is added to the signals at the observation points.

Each simulation example is run 50 times. Figure 5 and 6 show the resulting mean estimates for $\lambda = 10^4$ and $\lambda = 10^6$, resp. $\lambda = 10^5$ and $\lambda = 10^7$ when the cross-validation selection criterion has been used. The rough λ-values have been

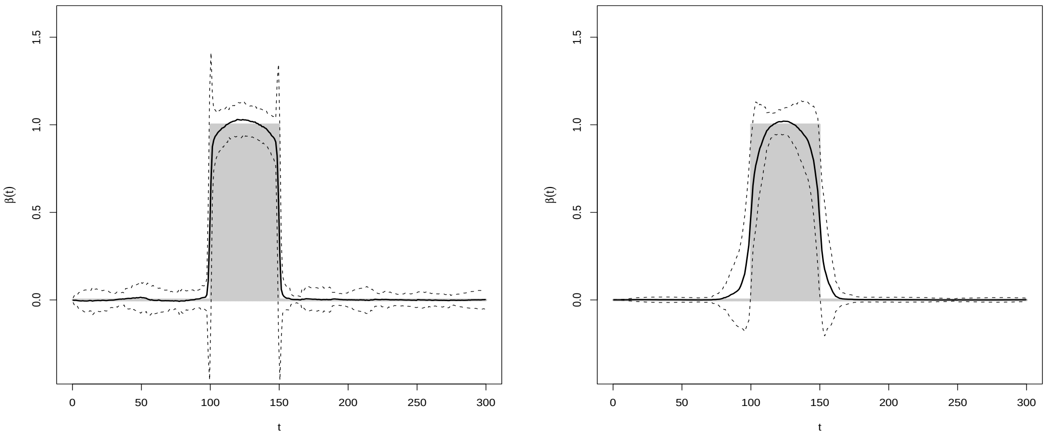


Figure 5: The considered plateau function and mean estimate by (simple ridge) BlockBoost for $\lambda = 10^4$ (left) and $\lambda = 10^6$ (right), +/- 2 (estimated) standard deviation curves of $\hat{\beta}$ (dashed lines).

chosen on the basis of a (generalized) ridge regression with penalty matrix I, resp. Ω, but without boosting and taking into account all 300 predictors, i.e. $k = 300$ and $R^* = 0$. In the actual simulation the span has been chosen by $k = 50$ in the plateau function example and $k = 30$ for the normal-mixture example. In both cases, and at least in the relevant regions, the mean functions do not change substantially when shrinkage, resp. penalty is increased. Not surprisingly however, higher penalty causes lower variability, which can be seen from the dashed lines. These variability bounds are created by adding and subtracting two times the (estimated) standard deviation curves of $\hat{\beta}(t)$. In the plateau function example variability is particularly high at the relevant region's boundaries. Of course, compared to (blockwise) simple ridge, in the proposed smooth setting higher λ-values cause higher smoothing rather than higher shrinkage, but nevertheless shrinkage is obviously done in every iteration. Since the mean curves in the right panel of Figure 5 and 6 are not closer to zero than those in the left panel, it can be assumed that higher shrinkage, resp. smoothing is compensated by a higher number of boosting iterations. Moreover, compared to the true regression function in the normal mixture example, with both $\lambda = 10^5$ and $\lambda = 10^7$, the finally estimated coefficient function is shrunken only a little bit towards zero.

Beside considering different λ-values, when the true regression function is known, it is possible to estimate the resulting mean squared errors $E(\|\hat{\beta}(.) - \beta(.)\|^2)$ for every selection criterion. The distance between estimated and true regression function is measured in terms of the metric $\|\hat{\beta}(.) - \beta(.)\|^2 = \int_I (\hat{\beta}(t) - \beta(t))^2 \, dt$. Table 1 shows the MSE (derived from 50 simulation runs) for both simulation examples, fixed $\lambda = 10^6$, penalty matrix Ω in both cases and varying

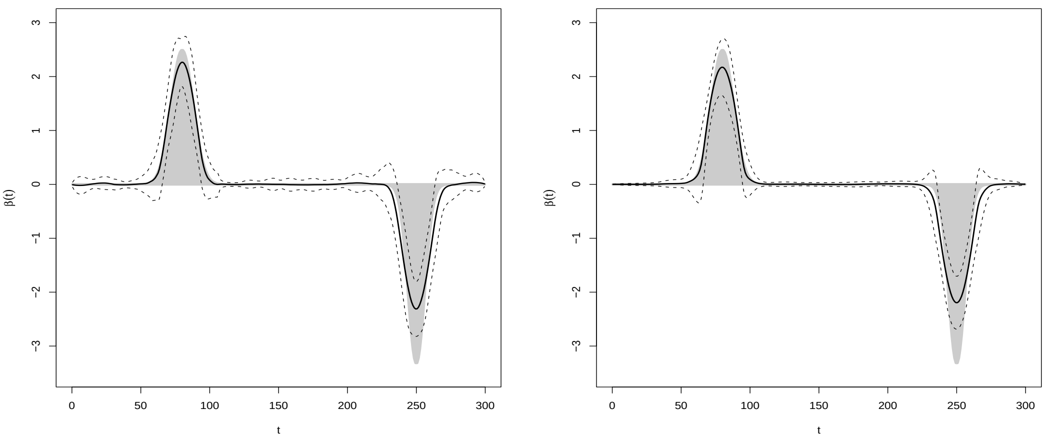


Figure 6: The considered normal mixture functions and mean estimate by (gen. ridge) BlockBoost for $\lambda = 10^5$ (left) and $\lambda = 10^7$ (right), +/- 2 (estimated) standard deviation curves of $\hat{\beta}$ (dashed lines).

		regression function	
		plateau function	normal mixture
selection criterion	RSS	7.947 (0.128)	17.430 (1.229)
	CV	7.896 (0.124)	17.450 (1.301)
	GCV	7.957 (0.125)	17.297 (1.223)

Table 1: MSE $E(\|\hat{\beta}(.) - \beta(.)\|^2)$ for the considered regression functions and varying selection criteria; estimated standard errors in parentheses.

selection criteria. With the standard errors (given in parentheses) in mind there do not really seem to be any differences between different criteria. Nevertheless we decided to use CV as default in the following sections.

3.2 Comparison between methods

Let the signal again be generated by (2). The error terms in $y_i = \sum_j x_i(t_j)\beta(t_j) + \varepsilon_i$ are specified by $\varepsilon_i \sim N(0, 10^2), \varepsilon_i \sim N(0, 30^2), \varepsilon_i \sim N(0, 50^2)$. These settings correspond to signal-to-noise ratios of about 30, 10 and 6. As before measurement error τ_{ij} is added on the signals at t_j considering the cases $\tau_{ij} = 0, \tau_{ij} \sim N(0, 0.25^2), \tau_{ij} \sim N(0, 1)$. The respective signal-to-noise ratio is ∞, $\in [16, 24]$ resp. $\in [4, 6]$ (depending on t_j). The considered parameter function is the normal-mixture function. In order to evaluate the performance of blockwise boosting we compare the proposed algorithm to several other procedures. All computations were carried out by the statistical program R, see R Development Core Team (2007) for further information. In particular we compare the following procedures:

- Principal Components Regression PCR (Massy, 1965) with the coefficients being estimated using the R-package pls; the number of components is determined by leaving one out cross-validation (CV).

- Partial Least Squares PLS (Wold, 1975); as above we use the R-package pls, PLS is performed using the classical orthogonal scores algorithm, as described in Martens and Naes (1989); CV serves to estimate the number of latent factors.

- Lasso (Tibshirani, 1996); the computations are done based on to the algorithm of Efron et al. (2004), which is implemented in the R-package lars. The amount of shrinkage is determined by 5-fold cross-validation.

- Ridge Regression (Hoerl and Kennard, 1970) with the optimum amount of shrinkage being estimated by CV.

- Generalized Ridge Regression with first-difference penalty; again we use CV to determine the penalty parameter.

- Functional Data Approach (Ramsay and Silverman, 2005), from now on denoted by FDA; there is an updated R-package fda available now. For smoothing the signal and the coefficient vector we use 60 and 40 B-spline basis functions respectively. According to Ramsay & Silverman roughness of the regression function is to be penalized; so we chose to penalize curvature, with a penalty parameter chosen by CV.

- P-Splines; here we use the same functions as before for FDA, but without smoothing the signal, and imposing a third-difference penalty on the B-spline coefficients as proposed by Marx and Eilers (1999); as before the penalty parameter is determined by CV.

- Elastic Net (Zou and Hastie, 2005) estimated via the R-package elasticnet. For fixing the shrinkage parameters we use the procedure proposed by Zou & Hastie.

- Blockwise Boosting; we select k as described above, with $K = \{20, 30, 40, 50\}$, where the amount of shrinkage/smoothing has been fixed by $\lambda = 10^6$.

It is seen that all relevant tuning parameters are chosen in an adequate and widely accepted way to ensure fairness when the different methods are compared. We also investigated Random Forests (Breiman, 2001) and (weighted) k-Nearest

Neighbors, whose performance however was not competitive in the considered simulation setting.

For every combination of error term and measurement error specification we create a test set of $n_t = 500$ observations and 100 learning data sets, each consisting of 40 observations. Selection of tuning parameters is based on the respective learning data set only. In each case we investigate two measures of performance, the accuracy of the parameter estimate

$$MSE_\beta = \sum_j (\beta(t_j) - \hat{\beta}(t_j))^2$$

and the prediction mean squared error

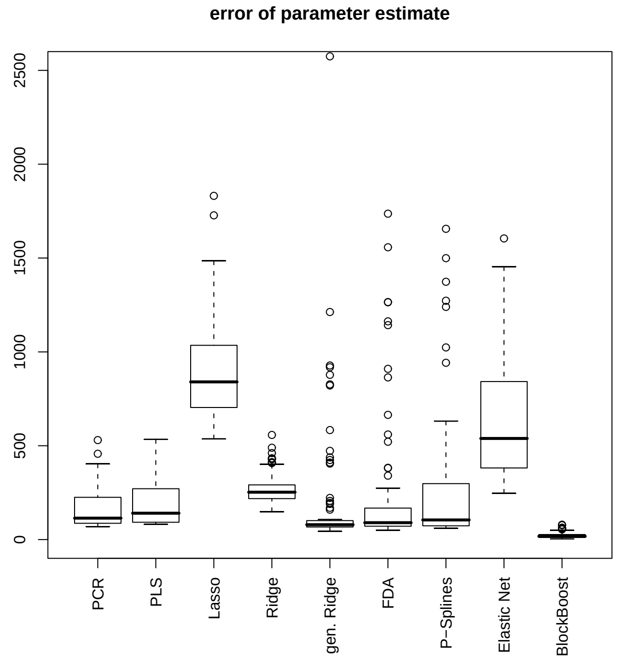
$$MSE_y = \frac{1}{n_t} \sum_{i=1}^{n_t} (y_i - \hat{y}_i)^2.$$

The first quantity is an approximation of the (squared) distance $\int_I (\beta(t) - \hat{\beta}(t))^2 \, dt$. Figure 7 shows a graphical summary of the observed MSE_β and MSE_y values when test set and training data sets are created according to the last combination of error term and measurement error, i.e. $\varepsilon_i \sim N(0, 50^2)$, $\tau_{ij} \sim N(0, 1)$. It should be noted that in the case of FDA and P-Splines some outliers of MSE_β and MSE_y are not shown because they were too extreme. It is seen from Fig. 7 that BlockBoost has superior performance not only in terms of the mean or median. Also variability is very low when compared to the other procedures. Due to the occurrence of extreme outliers (mainly in case of FDA and P-Splines) the mean does not seem to be the right measure to compare the different methods in a summarized way for all simulation settings. So Table 2 and 3 show the results in terms of the median over the 100 learning data sets.

τ_{ij}	$= 0$			$\sim N(0, 0.25^2)$			$\sim N(0, 1)$		
$\varepsilon_i \sim N$	$(0,10^2)$	$(0,30^2)$	$(0,50^2)$	$(0,10^2)$	$(0,30^2)$	$(0,50^2)$	$(0,10^2)$	$(0,30^2)$	$(0,50^2)$
PCR	59	67	87	93	206	94	84	97	114
PLS	60	67	83	71	80	90	83	106	140
Lasso	2254	2634	2664	792	1654	2144	350	636	840
Ridge	57	63	74	109	115	178	95	152	252
gen. Ridge	56	63	78	58	67	76	62	68	79
FDA	57	63	87	57	65	87	61	66	90
P-Splines	59	65	105	60	66	98	62	66	105
Elastic Net	2128	2982	4412	331	1535	2034	103	244	539
BlockBoost	13	15	18	13	13	16	15	16	18

Table 2: Median of MSE_β over the 100 learning data sets for PCR, PLS, lasso, ridge and generalized ridge (first-difference penalty) regression, FDA, P-splines, elastic net and BlockBoost.

After inspection of Figure 6 a good performance of BlockBoost could be expected. Indeed, the regression function estimated by BlockBoost is quite close to



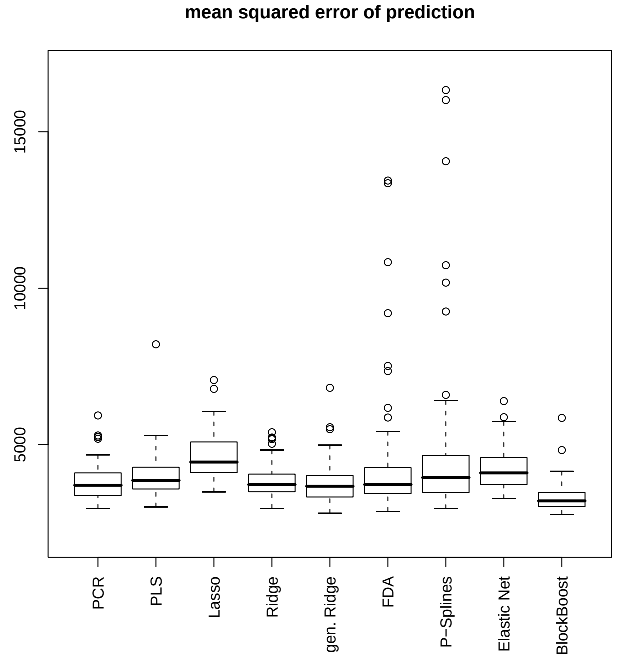


Figure 7: Detailed summary of MSE_β (left) and MSE_y (right) for the considered methods; test set and learning data sets were created using the specifications $\varepsilon_i \sim N(0, 50^2)$, $\varepsilon_{ij} \sim N(0, 1)$.

τ_{ij}	$= 0$			$\sim N(0, 0.25^2)$			$\sim N(0, 1)$		
$\varepsilon_i \sim N$	(0,10^2)	(0,30^2)	(0,50^2)	(0,10^2)	(0,30^2)	(0,50^2)	(0,10^2)	(0,30^2)	(0,50^2)
PCR	151	1250	3748	160	1257	3813	474	1560	3702
PLS	150	1252	3748	163	1283	3921	459	1675	3856
Lasso	132	1142	3262	218	1308	3664	793	2199	4443
Ridge	142	1201	3574	160	1215	3645	470	1630	3723
gen. Ridge	142	1209	3665	155	1277	3703	384	1527	3670
FDA	140	1207	3681	156	1259	3805	383	1562	3724
P-Splines	143	1211	3796	157	1256	3839	383	1529	3947
Elastic Net	153	1238	3661	184	1311	3593	494	1777	4095
BlockBoost	136	1080	3073	157	1065	3155	384	1419	3201

Table 3: Median of MSE_y over the 100 learning data sets for PCR, PLS, lasso, ridge and generalized ridge (first-difference penalty) regression, FDA, P-splines, elastic net and BlockBoost.

the true function, at least if we accept the other methods' performance as a kind of standard. In general the chemometrics regression tools PCR, PLS and ridge as well as the smoothing methods work relatively well. Since lasso and elastic net only select single measurement points, their bad performance is not surprising. Interestingly their performance increases with enlarging measurement error. Poor estimation of the true regression function however does not necessarily cause bad prediction. As Table 3 shows the prediction accuracy of lasso and elastic net is similar to that achieved by other methods. Nevertheless the column minimum is almost always found in the BlockBoost row.

4 Evaluation by Real World Data

Simulation scenarios have the advantage that the underlying structure is known and the accuracy of estimates may be investigated. When dealing with real data sets the true structure is unknown and the statistical model is usually only an approximation. The strength of the approach is that one may investigate how well this approximation works in practice.

4.1 Weather in Canada

The example data are taken from Ramsay and Silverman (2005), the well known monograph about functional data analysis, and can be downloaded from the related website `http://www.functionaldata.org`. Here the average daily precipitation and temperature at 35 Canadian weather stations is reported. As Ramsay & Silverman we try to predict the logarithm of the total annual precipitation from the pattern of temperature variation through the year. In terms of Section 2.2 the temperature profiles, the "signals", are digitized by $p = 365$ observation points.

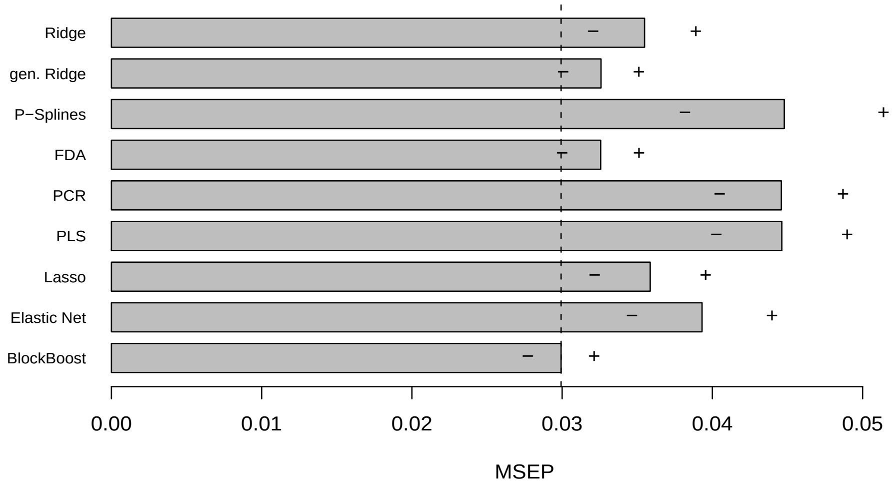


Figure 8: Mean Squared Error of Prediction for the considered methods averaged over all 200 random splits, +/− 2 (estimated) standard errors, minimum marked by dashed line; temperature profiles of 35 Canadian weather stations as predictors, log total annual precipitation as response.

We consider the same methods as in the simulation study with tuning parameters chosen as described above. In the case of FDA, however, we changed the specifications according to Ramsay and Silverman (2005). Above all this means that Fourier basis functions rather than B-splines are used to smooth signal and regression function - due to the periodicity of weather data. B-spline basis functions serve to carry out the P-splines approach proposed by Marx and Eilers (1999). In the BlockBoost algorithm shrinkage is fixed by $\lambda = 10^5$. For the evaluation of the various methods we consider 200 random splits of the data into a training data set of size 25 and a test set of size 10. Performance is measured by the prediction accuracy in the test sample, i.e. a measure as MSE_y in the previous section. Figure 8 summarizes the results in terms of the squared error of prediction averaged over all test observations and random splits. By adding and subtraction 2 estimated standard errors approximate confidence intervals for the performance measure are created.

The winner is BlockBoost; it clearly outperforms PCR, PLS, P-splines and elastic net. It is even competitive to the functional data approach from Ramsay and Silverman (2005), who used the same data to illustrate their method. Concerning prediction accuracy we cannot state that smoothing methods generally outperform other procedures - or the other way round. The neighborhood selection mechanism of BlockBoost however seems to work quite well.

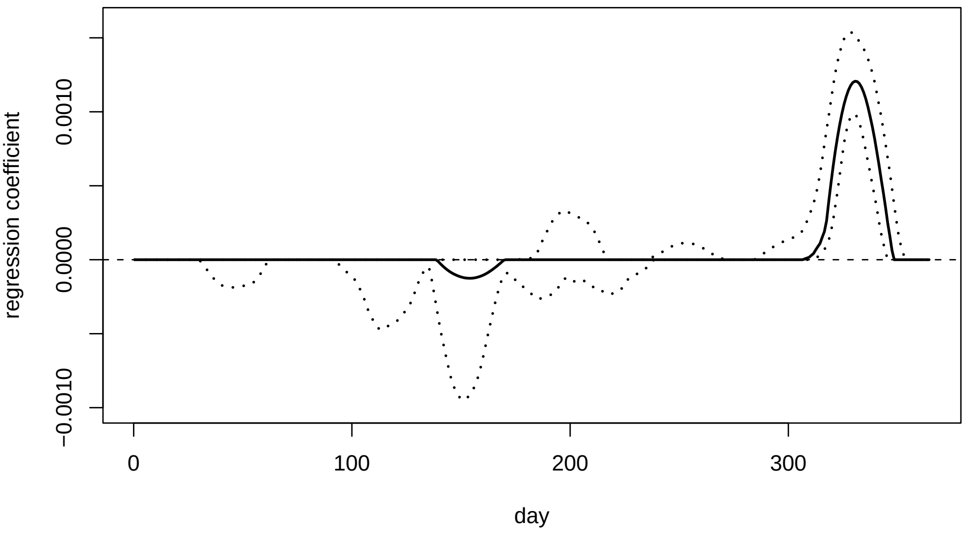


Figure 9: Canadian weather data: Coefficient function estimated by BlockBoost (solid line) together with pointwise 90% bootstrap (percentile) confidence bands (dotted lines).

Evaluating the Variability of BlockBoost

Prediction accuracy of BlockBoost turned out to be competitive on the Canadian weather data set. When real data is investigated, however, the analyst usually needs some measure of variability of the estimated parameters for the data at hand. Since for Boosting procedures in general reliable estimators of variances have not been derived yet, we decided to give pointwise bootstrap confidence bands. For actual computation the R package boot was used. Figure 9 shows the result when the percentile method is chosen. The dotted lines are based on 2000 bootstrap samples of size 35, drawn with replacement from the given weather data. Apparently only the "hump" on the right should be taken seriously, whereas the small negative effect of high temperatures around day 160 may be dismissed.

4.2 Near-Infrared Spectroscopy

Near-infrared (NIR) spectroscopy is based on the absorbtion of electromagnetic radiation at wavelengths in the near infrared region. In food analysis theoretically the concentrations of constituents such as water or carbohydrate can be determined using absorbtion spectroscopy. However, since the chemical information is mostly obscured by changes in the spectra caused by physical properties, NIR spectroscopy requires so-called *calibration* against a reference method for the constituent of interest. This calibration is usually done by linear regression of the reference data on the spectral data, cf. Osborne (2000). Due to the functional shape of the spectra (see Figure 1), deriving the calibration equation in food analysis is a typical application of the regression problem investigated in this article. Also Marx and Eilers (1999) use such a chemometric example to illustrate their P-splines. We use data from Christensen et al. (2004), who applied spectroscopy to measure marzipan composition and compared a number of infrared and near infrared set-ups and sampling techniques; data download from `http://www.models.kvl.dk/research/data/Marzipan`. We look at the wavelengths region between 850 and 2050 nm and use NIR spectra measured with fibre probe on NIRSystems 6500 in steps of 2 nm, for details see Christensen et al. (2004). Absorbance is measured via the transformation $\log(1/R)$, with R denoting the reflectance. Since traditional analytical procedures for determining moisture and sugar content in marzipan are time-consuming and destructive to the sample, cf. Christensen et al. (2004), it is quite attractive to alternatively perform very simple and fast NIR spectroscopy, which in addition allows several constituents to be measured concurrently. Unfortunately the NIR reflectance spectrum is influenced by the particle size of the sample. Thus shifts are generated which are not related to the constituent of interest. Hence many spectroscopists prefer to use derivatives instead of raw spectra. The first derivative for example, the slope of the spectrum, is calculated as the difference between $\log(1/R)$ at two adjacent wavelengths, see Osborne (2000). Consequently for each response (sugar / moisture content) we study two representations of the spectra: the raw and the first-difference spectra, digitized by p measurement points with $p = 600$ and $p = 599$ respectively.

As in the previous example we consider 200 random splits of the sample data into two independent data sets, in the current case consisting of 22 training and 10 test observations respectively. We investigate the same methods as before with tuning parameters chosen as described in subsection 3.2. Since in the current chemometric example we cannot expect periodicity, in the case of FDA the Fourier basis functions are replaced by B-splines. Smoothing the signals and the regression function is done via 100 and 22 basis functions respectively. The latter applies to P-splines, too. Our rationale is to use the number of basis functions corresponding to the number of observations in the training data set, as without penalty this would cause a perfect fit. Generalized ridge regression shows that now much lower shrinkage is needed than in the previous example - especially in case of the difference spectra. So in the BlockBoost algorithm we choose $\lambda = 10^{-1}$ (sugar) / $\lambda = 10^{-2}$ (moisture) and $\lambda = 10^{-3}$ (sugar) / $\lambda = 10^{-4}$ (moisture) for raw and difference spectra respectively.

Again the measure of performance is the prediction in the test sample. Figure 10 and 11 show the performance for the original signal and the difference spectra. For sugar as well as moisture the prediction accuracy of BlockBoost is quite

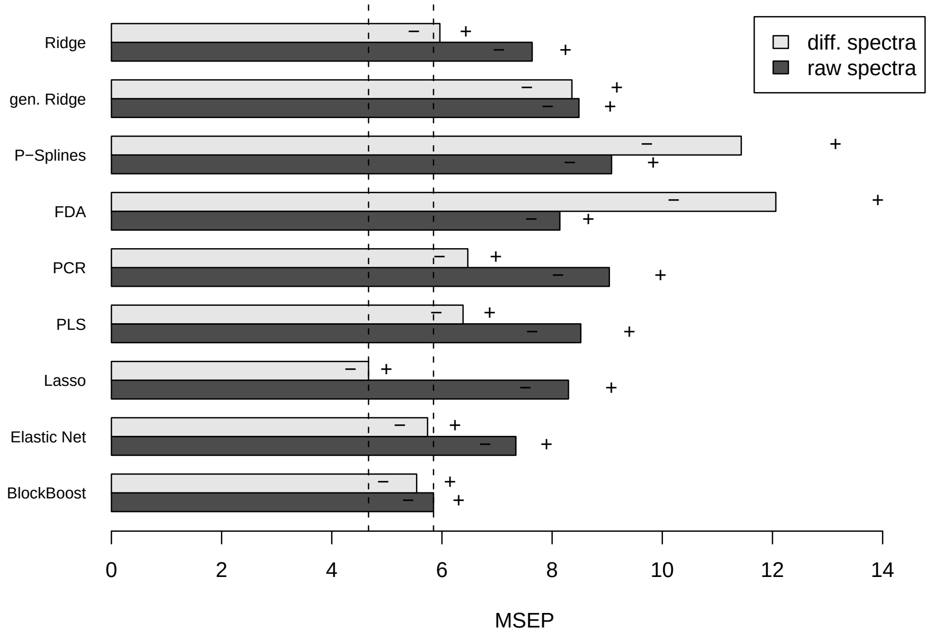


Figure 10: Mean Squared Error of Prediction for the considered methods averaged over all 200 random splits, +/− 2 (estimated) standard errors, minima marked by dashed lines; raw and first-difference NIR spectra as predictors, sugar content as response.

high - when compared to other methods. In three of four considered situations BlockBoost is among the best performing procedures.

5 Concluding Remarks

We proposed a boosting technique for implicit feature extraction in signal regression. This procedure is mainly based on repeated (generalized) ridge regression on groups - or blocks - of adjacent variables. So it is called blockwise boosting. It turned out to be highly competitive in both simulation studies and real world data evaluation. However, before running BlockBoost several tuning parameters have to be fixed. Simulation studies showed that the value of the penalty parameter λ does not have to be chosen as accurate as possible. The procedure is comparatively resistent to a modified amount of shrinkage. Stronger shrinkage, resp. smoothing should be compensated by a higher number of boosting iterations. But nevertheless it is necessary to have a rough idea about the right

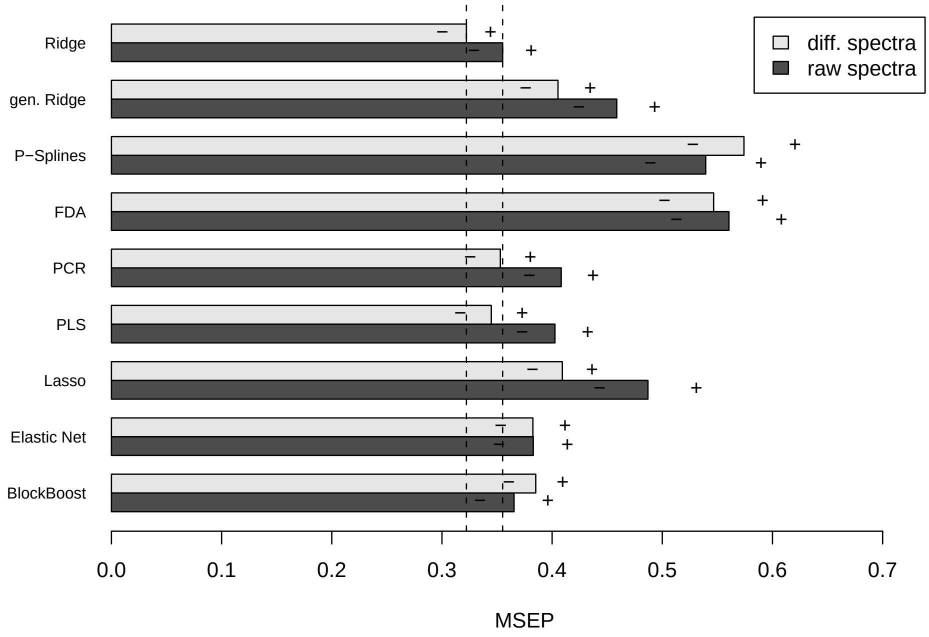


Figure 11: Mean Squared Error of Prediction for the considered methods averaged over all 200 random splits, +/− 2 (estimated) standard errors, minima marked by dashed lines; raw and first-difference NIR spectra as predictors, moisture content as response.

amount of shrinkage. This information can e.g. be gathered by cross validation of a generalized ridge regression without boosting and taking into account all predictors at hand. Further work is to be done concerning the choice of k in the BlockBoost algorithm. We proposed a simple strategy that worked quite well, a desirable procedure however automatically determines the adequate k-value in *every* boosting iteration. One possibility is comparing every $k \in \{1, 2, \ldots, K\}$ with respect to a kind of AIC. But the computational effort is very high.

The procedure proposed here aims at finding relevant areas of the signal and producing smooth parameter estimates on the found intervals. The smooth parameter estimates make the approach attractive for interpretation. We did not dwell on methods like variable fusion (Land and Friedman, 1997) or the fused lasso (Tibshirani et al., 2005), whose focus is on the estimation of piecewise constant functions, which may be interesting in applications with some few peaks and some constant parameters otherwise. Moreover, concerning the predictors, for practical application of the fused lasso Tibshirani et al. use quite rough data from mass spectroscopy and gene expression profiles - with good reason. As

Land and Friedman - the inventors of fusion methodology - sum up, in situations *"where the predictor curves are (...) fairly smooth, simulations bear out that variable fusion is (...) not superior to ridge regression and PLS"*. But rather smooth curves are the type of predictor mainly investigated in the article.

Alternatively to the proposed BlockBoost interval selection may be realized via expanding the coefficient function in basis functions with local support along the lines of Ramsay and Silverman (2005) or Marx and Eilers (1999). After doing so any variable selection technique can be employed on the basis coefficients, maybe the lasso or boosting. The latter has been used for example by Krämer (2006). But choosing the adequate number and placing of basis functions is a complex task, see for example Eilers and Marx (1996). In the given situation the placing of basis functions predetermines to some extent which intervals can be selected at all. So the proposed BlockBoost offers higher flexibility with respect to the areas that can be selected. Furthermore, when for example the coefficient curve is represented by B-Splines (of degree 2 or 3) and the lasso is used for basis coefficient selection, the estimated function often has a camel or dromedary like shape, resulting from the shape of the B-Splines. Also here the proposed technique is more flexible.

Although the presented BlockBoost algorithm is constructed for regression problems, the principal procedure can be used for classification as well. One possibility for handling a two class response (usually 0/1 coded) is adapting the LogitBoost algorithm presented by Friedman et al. (2000) and for example used by Dettling and Bühlmann (2003) in the context of gene expression data. In every boosting iteration LogitBoost creates a real valued working response, which is to be fitted by an (arbitrary) regression function. When using the proposed ridge type estimation based on variable blocks one obtains a version of blockwise boosting for generalized problems.

References

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

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

Breiman, L. (2001). Random forests. *Machine Learning 45*, 5–32.

Bühlmann, P. (2006). Boosting for high-dimensional linear models. *Annals of Statistics 34*, 559–583.

Bühlmann, P. and B. Yu (2003). Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association 98*, 324–339.

Christensen, J., L. Nørgaard, H. Heimdal, J. G. Pedersen, and S. B. Engelsen (2004). Rapid spectroscopic analysis of marzipan - comparative instrumentation. *Journal of Near Infrared Spectroscopy 12*, 63–75.

Dettling, M. and P. Bühlmann (2003). Boosting for tumor classification with gene expression data. *Bioinformatics 19*, 1061–1069.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics 32*, 407–499.

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

Frank, I. E. and J. H. Friedman (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics 35*, 109–148.

Freund, Y. and R. E. Schapire (1996). Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, 148–156.

Friedman, J. H., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics 28*, 337–407.

Hastie, T. and C. Mallows (1993). Discussion of "a statistical view of some chemometrics regression tools". *Technometrics 35*, 140–143.

Hastie, T., R. Tibshirani, and J. H. Friedman (2001). The elements of statistical learning. *Springer-Verlag, New York, USA*.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*, 55–67.

Hurvich, C. M., J. S. Simonoff, and C. Tsai (1998). Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society B 60*, 271–293.

Krämer, N. (2006). Boosting for functional data. *Proceedings of the 17th International Conference on Computational Statistics*, 1121–1128.

Land, S. R. and J. H. Friedman (1997). Variable fusion: A new adaptive signal regression method. Technical report 656, Department of Statistics, Carnegie Mellon University Pittsburg.

Martens, H. and T. Naes (1989). *Multivariate Calibration.* Chichester: Wiley.

Marx, B. D. and P. H. C. Eilers (1999). Genaralized linear regression on sampled signals and curves: A p-spline approach. *Technometrics 41*, 1–13.

Marx, B. D. and P. H. C. Eilers (2005). Multidimensional penalized signal regression. *Technometrics 47*, 13–22.

Massy, W. F. (1965). Principal components regression in exploratory statistical research. *Journal of the American Statistical Association 60*, 234–256.

Osborne, B. G. (2000). Near-infrared spectroscopy in food analysis. In R. A. Meyers (Ed.), *Encyclopedia of Analytical Chemistry*. Chichester: Wiley.

R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Ramsay, J. O. and B. W. Silverman (2005). *Functional Data Analysis* (2nd ed.). New York: Springer.

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

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

Tibshirani, R., M. Saunders, S. Rosset, J. Zhu, and K. Kneight (2005). Sparsity and smoothness vie the fused lasso. *Journal of the Royal Statistical Society B 67*, 91–108.

Wold, H. (1975). Soft modeling by latent variables: The nonlinear partial least squares approach. In J. Gani (Ed.), *Perspectives in Probability and Statistics, Papers in Honour of M.S. Bartlett*. London: Academic Press.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B 67*, 301–320.