

promoting access to White Rose research papers



Universities of Leeds, Sheffield and York
<http://eprints.whiterose.ac.uk/>

This is a final published version of a paper accepted for publication in **Journal of Statistical Planning and Inference**.

White Rose Research Online URL for this paper:

<http://eprints.whiterose.ac.uk/42949/>

Paper:

Di Marzio, M and Taylor, CC (2008) *On boosting kernel regression*. Journal of Statistical Planning and Inference, 138 (8). 2483 -2498.

<http://dx.doi.org/10.1016/j.jspi.2007.10.005>

On boosting kernel regression

Marco Di Marzio and Charles C. Taylor

DMQTE, G.d'Annunzio University, ITALY and Department of Statistics, University of Leeds, UK

Abstract

In this paper we propose a simple multistep regression smoother which is constructed in an iterative manner, by learning the Nadaraya-Watson estimator with L_2 boosting. We find, in both theoretical analysis and simulation experiments, that the bias converges exponentially fast, and the variance diverges exponentially slow. The first boosting step is analyzed in more detail, giving asymptotic expressions as functions of the smoothing parameter, and relationships with previous work are explored. Practical performance is illustrated by both simulated and real data.

Key words: Bias Reduction; Boston Housing Data; Convolution; Cross Validation; Local Polynomial Fitting; Positive Definite Kernels; Twicing.

1 Introduction

1.1 Objectives and motivation

Due to impressive performance, boosting (Schapire, 1990; Freund, 1995) has become one of the most studied machine learning ideas in the statistics community. Basically, a B -steps boosting algorithm iteratively computes B estimates by applying a given method, called a *weak learner*, to B different re-weighted samples. The estimates are then combined into a single one which is the final output. This ensemble rule can be viewed as a *powerful committee*, which is expected to be significantly more accurate than every single estimate. A great deal of effort is being spent in developing theory to explain the practical behaviour of boosting, and at the moment a couple of crucial questions appear to have been successfully

Email address:

dimarzio@dmqte.unich.it and c.c.taylor@leeds.ac.uk (Marco Di Marzio and Charles C. Taylor).

addressed. An important question is *why* boosting works. Now it seems that a satisfying response has been provided in that boosting is viewed as a greedy function approximation technique (Breiman, 1997; Friedman, 2001), where a loss function is optimized by B iterative adjustments of the estimate in the function space; at each iteration the weighting system indicates the direction of steepest descent. A second question concerns the Bayes risk consistency, and recent theoretical results (Jiang, 2004) show that boosting is not Bayes risk consistent and regularization techniques are needed. Recent results suggest that some regularization methods exist such that the prediction error should be nearly optimal for sufficiently large samples. Jiang (2004) and Zhang and Yu (2005) analyze boosting algorithms with *early stopping*, whereas alternative regularization methods are considered by Lugosi and Vayatis (2004) and by Zhang (2004). In practical applications, a stopping rule is often determined by recourse to cross-validation based on subsamples. However, an alternative more computationally efficient approach, using AIC-based methods, has recently been proposed by Bühlmann (2006).

To implement boosting we need to choose a loss function and a weak learner. Every loss function leads to a specifically shaped boosting algorithm. For example, *AdaBoost* (Schapire, 1990; Freund and Schapire, 1996) corresponds to exponential losses, and *L₂Boost* (Friedman, 2001; Bühlmann and Yu, 2003) to L_2 losses. Clearly, if a specific weak learner is considered as well, a boosting algorithm can be explicitly expressed as a multistep estimator, and some statistical properties can thus be derived.

In the present paper we propose a new higher-order biased nonparametric regression smoother that results from learning the Nadaraya-Watson (N-W) estimator by L_2 boosting. Note that in polynomial regression bias becomes more serious the higher the curvature of the regression function. In this latter case the use of a higher polynomial fit — that is asymptotically less biased — is not always preferable since: *i*) there is no guarantee that the regression function is sufficiently smooth to ensure explicit expressions for the asymptotic bias; and *ii*) they require much larger samples.

In Section 2 we establish the properties of our boosting algorithm for each iteration: exponentially fast bias reduction and exponentially slow variance inflation are proved. In Section 3 we explore the asymptotic behaviour (as $n \rightarrow \infty$) in the first boosting step and make clear links to previous work. The results of some simulation experiments for univariate data are summarized in Section 4. In Section 5 we investigate the potential of our boosting algorithm when applied to a real multivariate dataset. This extends the methods to higher dimensions, and requires the use of a data-based approach to select the smoothing parameter and number of boosting iterations. Overall, consistent gains almost always emerge with respect to both the N-W estimator and other boosting methods present in the literature. A few concluding remarks are contained in Section 6.

1.2 L_2 boosting

In what follows a description of L_2 boosting suitable for our aims is given; more details can be found in the references.

Given three real random variables, X , Y and ε assume the following regression model for their relationship

$$Y = m(X) + \varepsilon, \quad \text{with} \quad \mathbb{E} \varepsilon = 0, \quad \text{var} \varepsilon = \sigma^2, \quad (1)$$

where X and ε are independent. Assuming that n *i.i.d.* observations $S := \{(X_i, Y_i), i = 1, \dots, n\}$ drawn from (X, Y) are available, the aim is to estimate the mean response curve $m(x) = \mathbb{E}(Y \mid X = x)$. Note that $m(x) = r(x)/f(x)$ where $r(x) := \int yg(x, y) dy$, $f(x) := \int g(x, y) dy$ and g is the joint density of (X, Y) . This is the random design model, in the fixed design model we have a set of fixed, ordered points, x_1, \dots, x_n that are often assumed equispaced, so the sample elements are $s := (x_i, Y_i; i = 1, \dots, n)$.

L_2 boosting is a procedure of iterative residual fitting where the final output is simply the sum of the fits. Formally, consider a *weak learner* \mathcal{M} that is a crude smoother. An initial least squares fit is denoted by \mathcal{M}_0 . For $b \in [1, \dots, B]$, \mathcal{M}_b is the sum of \mathcal{M}_{b-1} and a least squares fit of the residuals $S_e := \{X_i, e_i := Y_i - \mathcal{M}_{b-1}(X_i)\}$. The L_2 boosting estimator is \mathcal{M}_B .

Typically, the minimal loss obtainable over all boosting iterations, will be achieved after a finite number of iterations. Actually, the more B increases, the more \mathcal{M}_B becomes complex and tends to closely reproduce the sample (overfitting). Therefore, a stopping rule is needed.

2 L_2 boosting and local polynomial fitting

2.1 Local polynomial and L_2 boosting

Given p , usually 0, 1 or 2, to estimate $m(x)$ we could solve

$$\min_{\beta_0, \dots, \beta_p} \sum_{i=1}^n \left\{ Y_i - \sum_{j=0}^p \beta_j (X_i - x)^j \right\}^2 K_h(X_i - x) \quad (2)$$

by fitting $\sum_{j=0}^p \beta_j (\cdot - x)^j$ to S . This gives $\widehat{m}^{(j)}(x; S, h) := j! \widehat{\beta}_j$. Here, the weight function $K_h(\cdot) := K(\cdot/h)/h$ is non-negative, symmetric and unimodal, and $h > 0$ is the bandwidth. This class of estimators is known as local polynomial regression

smoothers (see, for example, Fan and Gijbels, 1996). As mentioned in Section 1.1, the use of a p -degree polynomial is meaningful only when $m^{(p+1)}(x)$ exists; this constraint is regarded as the most severe required by this approach.

Notice that $\widehat{m}^{(j)}(x; S, h)$ is a least squares fit, so each pair h and p identifies a weak learner for L_2 boosting. But how to select p ? It is known that for a successful implementation of boosting we need a *weak* learner. Now within the class of the local polynomial smoothers the case $p = 0$ — called the N-W smoother — can be regarded as crude because it is the simplest polynomial (a constant term) to employ in the fit of the Taylor series expansion of m .

2.2 The Nadaraya-Watson smoother and L_2 BoostNW

Given a sample S , we want to estimate $m(x)$ in model (1). The N-W estimator is

$$\widehat{m}_{\text{NW}}(x; S, h) := \frac{n^{-1} \sum_{i=1}^n K_h(x - X_i) Y_i}{n^{-1} \sum_{i=1}^n K_h(x - X_i)}$$

which, as stated, is the solution of Equation (2) when $p = 0$. For the simplest interpretation, note that a N-W fit is a locally weighted average of the responses.

Now we recall a bias approximation of $\widehat{m}_{\text{NW}}(x; S, h)$ useful for the next section. Härdle (1990) gives a detailed treatment of the subject.

Let this usual set of conditions hold

- (1) x is an interior point of the sample space, i.e. $\inf(\text{supp}f) + h \leq x \leq \sup(\text{supp}f) - h$;
- (2) m and f are twice continuously differentiable in a neighbourhood of x ;
- (3) the kernel K is a symmetric PDF;
- (4) $h \rightarrow 0$ and $nh \rightarrow \infty$ as $n \rightarrow \infty$;
- (5) f'' is continuous and bounded in a neighbourhood of x .

Indicate by $\widehat{r}(\cdot; S, h)$ and $\widehat{f}(\cdot; h)$ the numerator and denominator of $\widehat{m}_{\text{NW}}(\cdot; S, h)$, respectively. Using condition (4), it has been shown that the leading term of a large sample approximation gives $E \widehat{m}_{\text{NW}} \approx E \widehat{r} / E \widehat{f}$. It is easy to show that

$$E \widehat{r}(x; S, h) = r(x) + \frac{h^2}{2} \mu_2 r''(x) + O(h^4); \quad (3)$$

where $\mu_k := \int v^k K(v) dv$, and that

$$E \widehat{f}(x; h) = f(x) + \frac{h^2}{2} \mu_2 f''(x) + O(h^4). \quad (4)$$

Therefore, the bias is of order $O(h^2)$, in particular:

$$m(x) - \widehat{m}_{\text{NW}}(x; S, h) \approx \frac{h^2 \mu_2}{2f(x)} \{r''(x) - m(x)f''(x)\}. \quad (5)$$

We propose L_2 boosting with the N-W estimator as our weak learner using the following pseudocode:

Algorithm: $L_2\text{BoostNW}$

1. (*Initialization*) Given S and $h > 0$,
 - (i) $\widehat{m}_0(x; h) := \widehat{m}_{\text{NW}}(x; S, h)$.
2. (*Iteration*) Repeat for $b = 1, \dots, B$
 - (i) $e_i := Y_i - \widehat{m}_{b-1}(X_i; h) \quad i = 1, \dots, n$;
 - (ii) $\widehat{m}_b(x; h) := \widehat{m}_{b-1}(x; h) + \widehat{m}_{\text{NW}}(x; S_e, h)$, where $S_e = \{(X_i, e_i), i = 1, \dots, n\}$.

2.3 Properties of $L_2\text{BoostNW}$

Let $(x_1, y_1), \dots, (x_n, y_n)$ be data from model (1), the Nadaraya-Watson estimates at the observation points are compactly denoted as

$$\widehat{\mathbf{m}}_0 = \mathbf{N}\mathbf{K}\mathbf{y}$$

where $\widehat{\mathbf{m}}_0^T := (\widehat{m}_0(x_1; h), \dots, \widehat{m}_0(x_n; h))$, $\mathbf{N}^{-1} := \text{diag}(\{\sum_{i=1}^n K_h(x_1 - x_i)\}, \dots, \{\sum_{i=1}^n K_h(x_n - x_i)\})$, $\mathbf{y}^T := (y_1, \dots, y_n)$ and $(\mathbf{K})_{ij} := K_h(x_i - x_j)$. Notice that this fit is linear, but unfortunately the hat matrix $\mathbf{N}\mathbf{K}$ is not symmetric, therefore the detailed theory established by Bühlmann & Yu (2003) for L_2 boosting of symmetric learners is not applicable here. Indicate as $\text{spec}(\mathbf{A})$ the set of the characteristic roots of the square matrix \mathbf{A} . It will be apparent that $L_2\text{BoostNW}$ works properly only if $\text{spec}(\mathbf{N}\mathbf{K}) \subset (0, 1]$; in Theorem 1 we define a class of kernels satisfying such a property. In Theorem 2 we give finite sample accuracy measures of $L_2\text{BoostNW}$ at step $b \geq 0$.

Theorem 1 *If the continuous second-order kernel K is*

- 1) *a Fourier-Stieltjes transform of a finite measure; and*
- 2) *symmetric and unimodal;*

then $\text{spec}(\mathbf{N}\mathbf{K}) \subset (0, 1]$. Moreover, $\min \text{spec}(\mathbf{N}\mathbf{K}) < 1$.

Proof: See the Appendix.

According to Theorem 1, gaussian and triangular kernels yield strictly positive characteristic roots, *while many popular ones such as Epanechnikov, Biweight and*

Triweight do not. This is somewhat surprising, in view of the fact that the kernel choice is often influenced by computational convenience.

Now define the mean squared error of $L_2\text{BoostNW}$, averaged over the observation points x_1, \dots, x_n , as

$$\text{ave-MSE}(\widehat{\mathbf{m}}_b; \mathbf{m}, \sigma^2) := \text{ave-bias}^2(\widehat{\mathbf{m}}_b; \mathbf{m}) + \text{ave-var}(\widehat{\mathbf{m}}_b; \sigma^2),$$

where $\mathbf{m}^T := (m(x_1), \dots, m(x_n))$ is the vector of the regression function at the observation points,

$$\text{ave-bias}^2(\widehat{\mathbf{m}}_b; \mathbf{m}) := \frac{1}{n} \sum_{i=1}^n (\mathbb{E}\widehat{m}_b(x_i; h) - m(x_i))^2,$$

and

$$\text{ave-var}(\widehat{\mathbf{m}}_b; \sigma^2) := \frac{1}{n} \sum_{i=1}^n \text{var} \widehat{m}_b(x_i; h).$$

Theorem 2 *Let $(x_1, y_1), \dots, (x_n, y_n)$ be data from model (1), then*

$$\text{ave-bias}^2(\widehat{\mathbf{m}}_b; \mathbf{m}) = \frac{1}{n} \mathbf{m}^T (\mathbf{U}^{-1})^T \text{diag}((1 - \lambda_k)^{b+1}) \mathbf{U}^T \mathbf{U} \text{diag}((1 - \lambda_k)^{b+1}) \mathbf{U}^{-1} \mathbf{m},$$

$$\text{ave-var}(\widehat{\mathbf{m}}_b; \sigma^2) = \frac{\sigma^2}{n} \text{trace}\{\mathbf{U} \text{diag}(1 - (1 - \lambda_k)^{b+1}) \mathbf{U}^{-1} (\mathbf{U}^{-1})^T \text{diag}(1 - (1 - \lambda_k)^{b+1}) \mathbf{U}^T\}.$$

where $\lambda_1, \dots, \lambda_n$ are the characteristic roots of $\mathbf{N}\mathbf{K}$, $b \geq 0$, and \mathbf{U} is a $n \times n$ invertible matrix of real numbers.

Moreover, if $\text{spec}(\mathbf{N}\mathbf{K}) \subset (0, 1]$, then

$$\lim_{b \rightarrow \infty} \text{ave-bias}^2(\widehat{\mathbf{m}}_b; \mathbf{m}) = 0,$$

$$\lim_{b \rightarrow \infty} \text{ave-var}(\widehat{\mathbf{m}}_b; \sigma^2) = \sigma^2,$$

$$\lim_{b \rightarrow \infty} \text{ave-MSE}(\widehat{\mathbf{m}}_b; \mathbf{m}, \sigma^2) = \sigma^2;$$

ave-bias^2 converges exponentially fast, while ave-var converges exponentially slow.

Proof: See the Appendix.

For a given step b , the bias-variance tradeoff emerges: increased characteristic roots correspond to a bandwidth reduction, with obvious consequences. But it is also apparent that, if Theorem 1 holds, for each $k \in [1, \dots, n]$ we have

$$\lim_{h \rightarrow 0} (1 - \lambda_k)^{b+1} = \lim_{b \rightarrow \infty} (1 - \lambda_k)^{b+1} = 0; \quad (6)$$

this suggests that bandwidth selection needs to be accomplished by taking into account the boosting iterations planned. However, as in the case of L_2 boosting of symmetric learners (Bühlmann & Yu (2003)), the bias decreases exponentially fast towards zero, while variance increases exponentially slow towards σ^2 , which shows a resistance to overfitting.

3 The first boosting step ($B = 1$)

3.1 L_2 BoostNW reduces the bias of the N-W estimator

In this Section we consider the asymptotic bias and variance at the first boosting step. This is an alternative perspective to the one used in the previous section in which conditional expectations were obtained for finite samples.

Theorem 3 *Assuming conditions (1)–(5) hold, after the first boosting step we have*

$$\begin{aligned} \mathbb{E} \widehat{m}_1(x; h) &= m(x) + o(h^2), \\ \text{var} \widehat{m}_1(x; h) &\leq 4 \text{var} \widehat{m}_0(x; h). \end{aligned}$$

Proof: See the Appendix.

As a consequence, we observe a reduction in the asymptotic bias from $O(h^2)$ in Equation (5) to $o(h^2)$ above. This conclusion is consistent with that found by Di Marzio and Taylor (2004, 2005), where boosting kernels gives higher order bias for both *density estimation* and *classification*. More generally, bias reduction was noted by Friedman *et al.* (2000) when considering *Adaboost*. Since at the first step the magnitude order of the variance is preserved, then the mean squared error is reduced provided that n is sufficiently large.

3.2 Links to previous work

3.2.1 Twicing and higher order kernels: theory

The procedure of adding the smoothing of the residuals to the first smoothing was firstly suggested by Tukey (1997, p. 526–531) and called *twicing*; he also suggested the possibility of further iterations. After this, Stuetzle and Mittal (1979) pointed

out that twicing for kernel regression in a fixed, equispaced design is

$$\widehat{m}_{SM}(x; s, h) := 2n^{-1} \sum_{i=1}^n K_h(x - x_i) Y_i - n^{-1} \sum_{i=1}^n K_h(x - x_i) \sum_{j=1}^n K_h(x_i - x_j) Y_j$$

where $x_1 = 0$ and $x_n = 1$. They observed that the second summand contains a discretization of a convolution of the kernel with itself. Thus, for a sufficiently fine, equispaced, fixed design twicing approximates the estimator $\widehat{m}_{SM}(x; s, h) = n^{-1} \sum K_h^*(x - x_i) Y_i$ with $K_h^* := 2K_h - (K * K)_h$. (The convolution between pdfs f and g is $(f * g)(x) := \int f(x - y)g(y) dy$.) Now note that K_h^* is a higher order kernel, here called the convolution kernel, consequently \widehat{m}_{SM} is a higher order bias method.

Note that although both are higher order biased, \widehat{m}_{SM} is defined only for the fixed equispaced design case, while \widehat{m}_1 is indifferent to the design. An obvious question concerns the possibility of extending \widehat{m}_{SM} to the random design, and so we will compare — both theoretically and numerically — the performance of such an extension with \widehat{m}_1 . Assume that K_h is a normal density with mean zero and standard deviation h , denoted as ϕ_h , because in this case the convolution is simply $(\phi * \phi)_h = \phi_{\sqrt{2}h}$. There are two main options of implementing twicing by using the convolution kernel:

$$\begin{aligned} \widehat{m}_1^*(x; S, h) &:= \frac{\sum_{i=1}^n \phi_h^*(x - X_i) Y_i}{\sum_{i=1}^n \phi_h^*(x - X_i)} = \frac{\sum_{i=1}^n \{2\phi_h(x - X_i) - \phi_{\sqrt{2}h}(x - X_i)\} Y_i}{\sum_{i=1}^n \{2\phi_h(x - X_i) - \phi_{\sqrt{2}h}(x - X_i)\}} \\ \widehat{m}_2^*(x; S, h) &:= 2 \frac{\sum_{i=1}^n \phi_h(x - X_i) Y_i}{\sum_{i=1}^n \phi_h(x - X_i)} - \frac{\sum_{i=1}^n \phi_{\sqrt{2}h}(x - X_i) Y_i}{\sum_{i=1}^n \phi_{\sqrt{2}h}(x - X_i)} \end{aligned}$$

in which \widehat{m}_1^* can be viewed as the closest one to \widehat{m}_{SM} . It simply amounts to dividing \widehat{m}_{SM} — that is a consistent estimator of r — by a density estimate: it is a ‘higher order N-W’ smoother derived from a ‘higher order Priestley-Chao’ one. Surely \widehat{m}_2^* appears a more direct implementation of the twicing idea and is most similar to \widehat{m}_1 . We can compare each of these to \widehat{m}_1 , and all three estimators can be compared (bias and variance etc.) with the true regression m in simulations.

Denote the numerator and denominator of $\widehat{m}_1^*(x; h)$ by $\widehat{r}_1^*(x; h)$ and $\widehat{f}_1^*(x; h)$. Naïvely plugging these in to equations (3) and (4) we would get

$$\begin{aligned} E \widehat{m}_1^*(x; S, h) &= \left\{ 2r(x) + h^2 \mu_2 r''(x) - r(x) - \frac{(\sqrt{2}h)^2}{2} \mu_2 r''(x) + o(h^2) \right\} \\ &\quad \times \left\{ 2f(x) + h^2 f''(x) - f(x) - \frac{(\sqrt{2}h)^2}{2} \mu_2 f''(x) + o(h^2) \right\}^{-1} \\ &= m(x) + o(h^2) \end{aligned}$$

and the $O(h^2)$ bias terms appears to have been eliminated. However, we note that, since \widehat{f}_1^* can easily take the value 0, the approximation $E \widehat{m}_1^* \approx E \widehat{r}_1^* / E \widehat{f}_1^*$ is no

longer valid. Although the numerator and denominator become zero at the same time, the denominator can take negative values while the numerator is positive, so the estimator will be very unstable.

Secondly, if we use equation (5) to obtain $E \widehat{m}_2^*$ we get

$$\begin{aligned} E \widehat{m}_2^*(x; S, h) &= 2 E \widehat{m}_{\text{NW}}(x; S, h) - E \widehat{m}_{\text{NW}}(x; S, \sqrt{2}h) \\ &= m(x), \end{aligned}$$

and so the $O(h^2)$ bias term is apparently eliminated for this estimator also.

3.2.2 Twicing and higher order kernels: some simulations

Our objective here is to compare the estimators and their ability to reduce bias in different parts of the sample space. We will adopt the experimental design of Hastie and Loader (1993) who considered adaptive kernels for use at the boundary. Specifically, we take $n = 50$ points which are (i) equispaced, and (ii) $X \sim f(x)$ with $f(x) = 6x(1-x)I_{[0,1]}(x)$. For each x_i we generate $Y_i = x_i^2 + \varepsilon_i$ with $\varepsilon_i \sim N(0, 1)$. Given h we can estimate the mean integrated squared error MISE $\widehat{m} = E \int (\widehat{m} - m)^2$ for each estimator \widehat{m} , including the basic N-W estimator.

In Table 1 we give the mean integrated squared bias, and the mean integrated variance corresponding to the optimal choice of smoothing parameter — to minimize MISE over the full range — for each estimator. In the case of boosting, the number of iterations was optimized over all pairs (h, b) . As in Hastie and Loader (1993), we give a breakdown according to the interior, and center of the range $[0, 1]$. The results are estimated from 200 simulations of sample size $n = 50$.

It can be seen that most of the MISE is due to the contribution at the boundaries, particularly the bias-squared, which is an order of magnitude greater. All three bias reduction methods make most of their impact in the boundary contribution, with the bias showing a substantial decrease and only a modest increase in variance, with an overall reduction in MISE compared with the standard N-W estimator. For this example, it seems that \widehat{m}_1 (boosting one iteration) is the best *single iteration* method for both equispaced and random design data. However, we note that the bias is not as small — even after several boosting iterations — as that obtained by Hastie and Loader (1993) for their local linear regression estimator, which used an adaptive smoothing parameter near the boundaries.

4 Simulation study ($B \geq 1$)

In this section we report the conclusions of a simulation study which verifies the finite sample performance of $L_2\text{BoostNW}$. To explore the potential of the method,

Bias ²	Equispaced				Random			
	h	Bound.	Centre	Total	h	Bound.	Centre	Total
\hat{m}_{NW}	0.23	0.00772	0.000506	0.00823	0.20	0.02249	0.001243	0.02374
\hat{m}_1	0.33	0.00560	0.000716	0.00632	0.28	0.01806	0.001102	0.01916
\hat{m}_1^*	0.29	0.00629	0.000540	0.00683	0.26	0.02032	0.001089	0.02141
\hat{m}_2^*	0.31	0.00572	0.000523	0.00624	0.26	0.01990	0.001335	0.02123
\hat{m}_k	0.46	0.00487	0.001078	0.00595	0.48	0.01571	0.001208	0.01691
Variance	Equispaced				Random			
	h	Bound.	Centre	Total	h	Bound.	Centre	Total
\hat{m}_{NW}	0.23	0.02231	0.01145	0.03375	0.20	0.02850	0.01087	0.03937
\hat{m}_1	0.33	0.02350	0.01062	0.03411	0.28	0.03120	0.01061	0.04181
\hat{m}_1^*	0.29	0.02324	0.01112	0.03436	0.26	0.02958	0.01096	0.04055
\hat{m}_2^*	0.31	0.02359	0.01121	0.03480	0.26	0.02997	0.01060	0.04056
\hat{m}_B	0.46	0.02418	0.00988	0.03405	0.48	0.03274	0.01026	0.04300

Table 1

Best MISEs decomposed for several kernel regression estimators: \hat{m}_{NW} — standard Nadaraya-Watson; \hat{m}_1 — twicing; \hat{m}_i^* , $i = 1, 2$ — higher order kernel methods; \hat{m}_B with $B = 4, 6$ (optimal) boosting iterations of $L_2BoostNW$ for equispaced and random spacing, respectively. Integrated bias-squared and variance evaluated over the boundary region $[0, 0.3) \cup (0.7, 1]$, and centre region $[0.3, 0.7]$ for fixed (equi-spaced) design, and random design points $x_i, i = 1, \dots, 50$. Averages taken over 200 simulations, with bandwidth chosen to minimize MISE in each case.

we defer the selection of the bandwidth and number of boosting iterations, and present the performance that each method gives when the bandwidth is optimally selected. Results which use a cross-validation selection of the required parameters will be discussed in section 5.

Our study is made of two parts. The first one is aimed to illustrate the general performance of $L_2BoostNW$ as a regression method *per se*; here we have chosen the models used by Fan and Gijbels (1996, pg. 111). In the second part we compare $L_2BoostNW$ with the L_2 boosting regression method proposed by Bühlmann and Yu (2003) using their simulation model. This comparison is particularly interesting because their method is closely related to ours, in that they learn a nonparametric smoother (using splines) by L_2 boosting.

4.1 General performance

Fan and Gijbels (1996) characterize their case studies as difficult estimation problems due to the level of the *noise to signal ratio* values. Consider model (1) with ε normally distributed; the simulation models are specified in Table 2, where, as

Model	$m(x)$	σ
1	$x + 2 \exp(-16x^2)$	0.4
2	$\sin(2x) + 2 \exp(-16x^2)$	0.3
3	$0.3 \exp\{-4(x+1)^2\} + 0.7 \exp\{-16(x-1)^2\}$	0.1
4	$0.4x + 1$	0.15

Table 2

The simulation models of Fan and Gijbels (1996).

suggested by Fan and Gijbels, a random design was adopted: for models 1, 2 and 3 $X \sim U(-2, 2)$ and for model 4 $X \sim N(0, 1)$. We performed simulations for sample sizes 50, 100 and 200. In Figure 1 we have plotted the integrated mean squared error for $n = 50$ for various values of (h, B) . The plots confirm that boosting can reduce the MISE if the smoothing parameter is chosen correctly. Numerical summaries are given in Table 3, which also include information for other sample sizes. This Table shows the best MISEs (calculated from 200 samples) of L_2 boosting with the N-W smoother as the weak learner, as well as the gain in MISE which can be achieved by boosting with respect to the N-W estimator.

The results for model 4 suggest that a very large smoothing parameter, together with very many boosting iterations, are preferred for data which are generated by a straight line. In Figure 2 we plot the estimates $\widehat{m}_b(x; h)$ for various values of b , and then compare the values $\widehat{m}_{255}(x; h)$, $\widehat{m}_{1000}(x; h)$ and $\widehat{m}_{10000;h}(x)$ with the true model and the standard (OLS) regression line. It can be seen that the effect of boosting has given a very similar result to a nonparametric polynomial fit with degree $p = 1$. This approximation seems to hold true of the other models as well, but we have preferred this example because it shows that boosting fixes one of the main problems of the N-W smoother *i.e.* — as pointed out by Müller (1993) — the difficulty of estimating straight regression lines when X is not uniformly distributed.

4.2 Comparison with boosted splines

The simulation model used by Bühlmann and Yu (2003) is specified by

$$m(X) := 0.8 + \sin(6X), \quad X \sim U(-1/2, 1/2), \quad \varepsilon \sim N(0, 4).$$

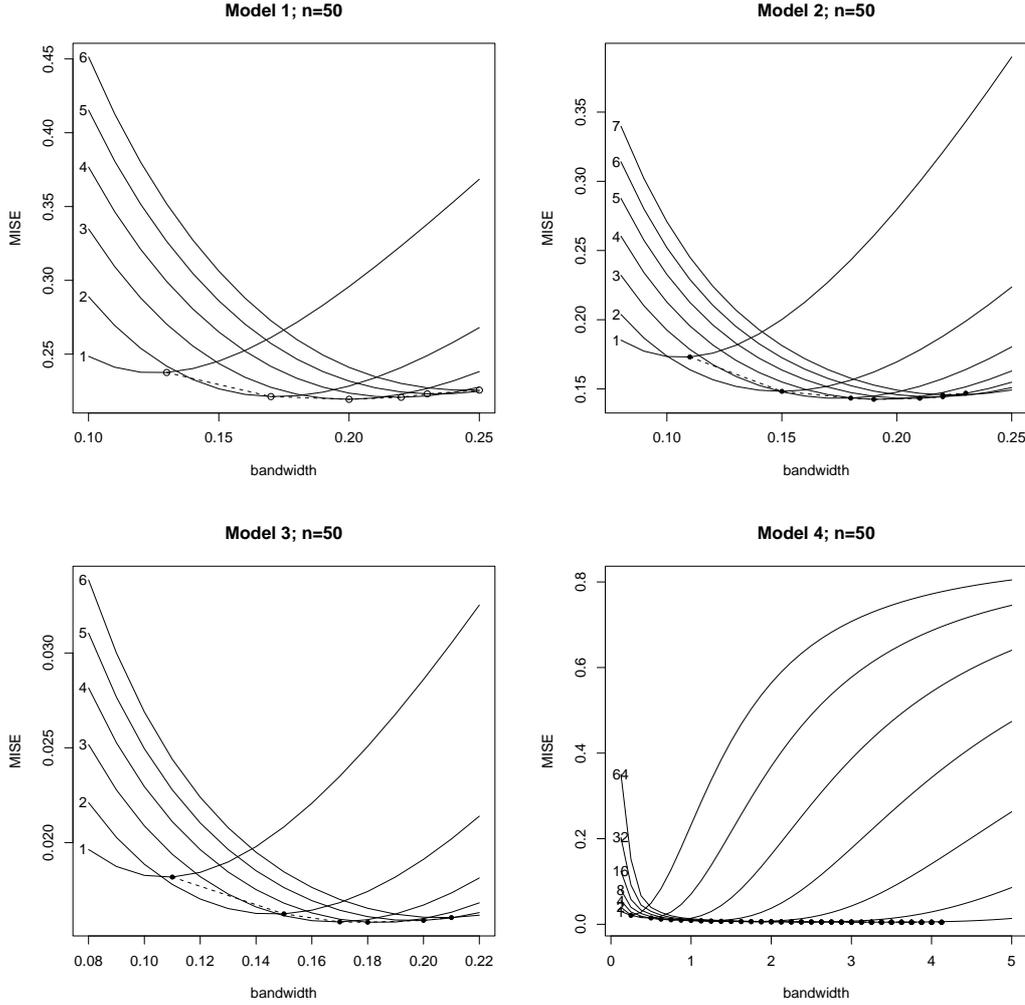


Fig. 1. MISEs for $n = 50$ for each of the four models given in Table 2. These are given as a function of the smoothing parameter for various values of $B = 1, 2, 3, \dots$ as shown in the plots). The points (joined by a dashed line) are the optimal values (over h) for each B .

They estimate $m(x)$ by using splines as the weak learner in L_2Boost with 100 samples drawn for each of four sample sizes. The accuracy criterion is equivalent to MISE and is estimated in the usual way. The values are summarized in Table 4, where the results of the original study are also shown. Note that both methods are optimized over their relevant parameters and so the comparison should be meaningful. For very small sample sizes $L_2boosting$ is outperformed by its base learner: marginally in the case of N-W; dramatically in the case of splines. In fact, our results are uniformly better for all sample sizes, and although our base learner is asymptotically inferior to splines, for all n $L_2BoostNW$ outperforms the boosted splines. So the best results were obtained when the N-W estimator is weaker than splines and the need of a really weak learner to employ in boosting seems confirmed.

Model	\hat{m}_{NW}		$p = 1$		$p = 2$		$L_2BoostNW$			gain
$n = 50$	h	MISE	h	MISE	h	MISE	MISE	k	h	
1	.13	.2477	.17	.2909	.29	.3367	.2268	3	.20	8.4%
2	.1	.1841	.15	.2101	.26	.2459	.1482	4	.19	19.5%
3	.1	.0196	.15	.0250	.27	.0286	.0167	4	.19	14.8%
4	.25	.0205	6*	.0044	4*	.0068	.0049	70*	3.3*	76.1%
<hr/>										
$n = 100$										
1	.12	.1247	.12	.1346	.20	.1383	.1104	4	.19	11.5%
2	.09	.0866	.10	.0907	.18	.0898	.0687	6	.19	20.7%
3	.09	.0095	.10	.0101	.19	.0104	.0077	6	.19	19.0%
4	.21	.0101	10*	.0019	4*	.0029	.0021	200*	5.3*	79.3%
<hr/>										
$n = 200$										
1	.09	.0658	.10	.0683	.16	.0642	.0557	5	.14	15.4%
2	.07	.0439	.08	.0445	.15	.0401	.0334	7	.18	24.0%
3	.07	.0049	.08	.0050	.15	.0045	.0037	8	.18	24.5%
4	.14	.0059	10*	.0011	4*	.0014	.0011	164-927	6.0*	81.4%

Table 3

Simulation results from boosting kernel regression using Fan & Gijbels models shown in Table 2. Gain is percentage improvement of the best boosting estimate over the best N-W smoothing, $p = 1, 2$ correspond to local polynomial fitting using Equation (2); * = boundary values of the grid used.

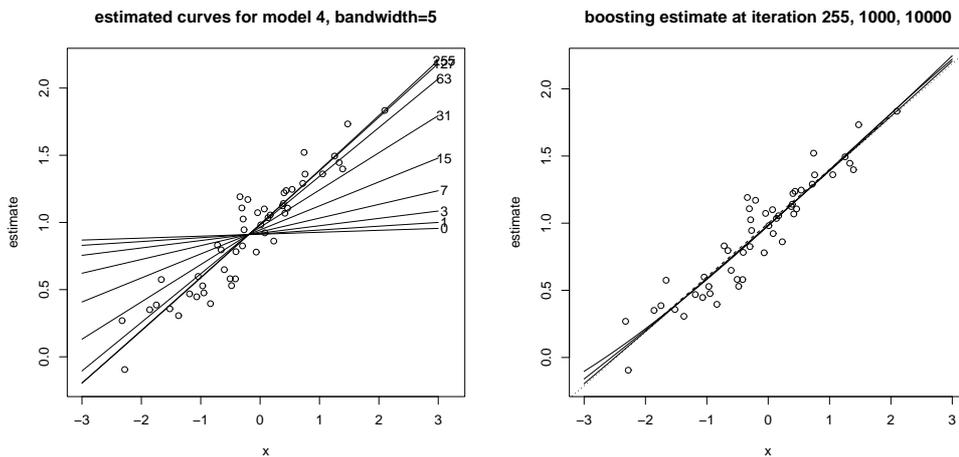


Fig. 2. Fitted line over $x \in [-3, 3]$ for 50 observations from Model 4 in Table 2. Left: various boosting iterations for smoothing parameter $h = 5$. Right: Fitted line for $B = 255, 1000, 10000$ iterations (continuous), regression line (dotted), true model (dashed).

n	Optimal	Optimal	Optimal	Optimal	Gain	
	N-W estimate	spline estimate	$L_2BoostNW$	L_2Boost spline	$L_2BoostNW$	L_2Boost spline
10	.7423	.7787	.7532	.9968	-1.5%	-28.0%
25	.2797	.3338	.2783	.3349	0.5%	-0.3%
50	.1505	.1657	.1460	.1669	3.0%	-0.7%
100	.0893	.0933	.0873	.0905	2.2%	0.9%
1000	.0148	.0128	.0086	.0113	42.0%	12.2%

Table 4

$L_2BoostNW$ performances when estimating the model used by Bühlmann and Yu (2003). The performances of smoothing splines and L_2Boost used by Bühlmann and Yu (2003) are also reported.

4.3 Optimal values of h and B

The result in Equation (6) suggest that h needs to increase with B . The case studies of Fan and Gijbels depicted in Figure 1 unequivocally suggest that boosting reduces oversmoothing effects if intensively iterated. Here we illustrate this by a new, *ad hoc* example based on the model used by Bühlmann and Yu. We drew 200 samples with $n = 200$ and estimated the regression function for various bandwidths and boosting iterations. The accuracy results are shown in Figure 3, where many MISE/iteration curves are depicted. The best MISE occurs when $h \approx 0.2$, but values quite close to this occur for each setting of the bandwidth. Remarkably, note that when the bandwidths are around 3–3.5 times bigger than 0.2, nearly optimal MISEs are reached after several hundreds of iterations, and moreover the best N-W estimate is always beaten for $B > 700$! Finally, Figure 3 also suggests how bandwidths of the same magnitude work similarly, another reason to conclude that boosting is less sensitive to the bandwidth selection task than standard kernel regression.

Overall, note that regularizing through oversmoothing in conjunction with many iterations increases the combinations of (h, B) for which boosting works. Thus, the potential of reducing the need of an accurate bandwidth selection and stopping rule clearly emerges.

5 Application to multidimensional real data

In this section we investigate the behaviour of our smoother in a more practical scenario, *i.e.* by using multivariate data and selecting both the smoothness degree

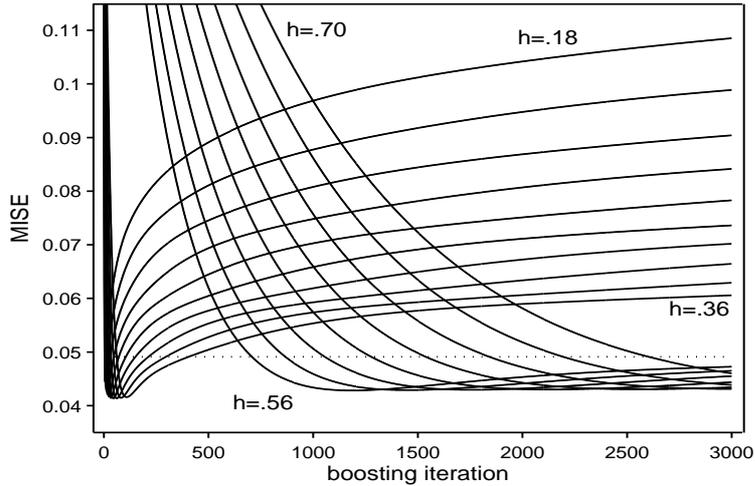


Fig. 3. $L_2BoostNW$ estimates of the Bühlmann and Yu model. MISEs for $n = 200$ given as a function of boosting iteration for various values of h . Dotted line: best MISE of the N-W estimator.

and the number of boosting iterations by cross-validation. We extend our smoother in the most usual way, that lies in building multiplicative kernels with a diagonal bandwidth matrix. In particular, with D -dimensional data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, we employ, with obvious notation, the following weight function

$$\prod_{d=1}^D K_h(x_d - x_{id}).$$

We use the normal kernel function because this ensures that the conditions of Theorem 1 hold in the multivariate setting.

We obtain (h_{CV}, B_{CV}) by leave-one-out cross-validation, *i.e.* as the pair that solves

$$\min_{h,b} = \sum_{i=1}^n (y_i - \widehat{m}_b^{(-i)}(\mathbf{x}_i; h))^2, \quad (7)$$

where $\widehat{m}_b^{(-i)}(\mathbf{x}_i; h)$ is the $L_2BoostNW$ estimate of $m(\mathbf{x}_i)$ when the i th observation is omitted.

We exemplify our method with the Boston housing data. This dataset, created by Harrison and Rubinfeld (1978), has been extensively analyzed in the statistical learning literature; see, for example, Breiman and Friedman (1985), Doksum and Samarov (1995) and Chaudhuri et al. (1997). It contains data for 506 census tracts in the Boston area taken from 1970 Census. Each of these 506 instances has 14 socio-economic variables (13 continuous and one binary). The response variable is the logarithm of the median value of owner-occupied homes in \$1000's. Note that in this dataset many of the explanatory variables have approximately linear relationships, so the curse of dimensionality problem may not be so evident as the

number of variables could lead us to believe.

	MSE	MSE _{opt}	h_{CV}	h_{opt}	B_{CV}	B_{opt}
local linear	0.1593	0.1504	2.00	2.55		
\hat{m}_{NW}	0.2575	0.2553	0.55	0.50		
$L_2BoostNW$	0.1525	0.1477	2.00	1.70	119	115
parametric linear	0.3340					

Table 5

Results from the Boston housing data.

Since we are using a common smoothing parameter for all variables, we have standardized the data. We randomly chose 350 instances as a training set, and the remaining data as a test set. The accuracy criterion was the mean squared error (MSE) on the test data. As a benchmark we used the plain N-W estimator, the local linear polynomial estimator (the solution of Equation (2) when $p = 1$) and a standard parametric linear model. Each cross-validation search of h was performed in the interval $[0, 3]$. The results are summarized in Table 5. The parametric linear fit has a MSE of 0.3340, suggests that a certain linearity is present in the data. This appears confirmed by the good performance of the local linear estimator that yields an accuracy of 0.1593 which outperforms the N-W fit. Concerning our multistep estimator, the cross-validation search of a pair (h, B) was performed over the grid $[0, 1, \dots, 3] \times [1, 2, \dots, 200]$, with a MSE of 0.1525. It is clear from Table 5 that $L_2BoostNW$ performs well in higher dimensions, and that cross-validation can be used to successfully obtain the pair (h, B) . Residual plots, shown in Figure 4 confirm this view. Note also that these accuracy values appear quite similar to the results from model 4 of Table 3 where a univariate linear model was estimated, so our estimator seems to coherently extend its properties to the multivariate setting. Another interesting issue is that the cross-validation search was very precise, because for $B \leq 200$ the best possible MSE of our boosted estimator is 0.1477, and the optimal setting of (h, B) is very similar to the cross-validation solution.

Chaudhuri et al. (1997) have given some motivation for using only RM, LSTAT, DIS as covariates. The results are quite consistent with the ‘all variables’ case.

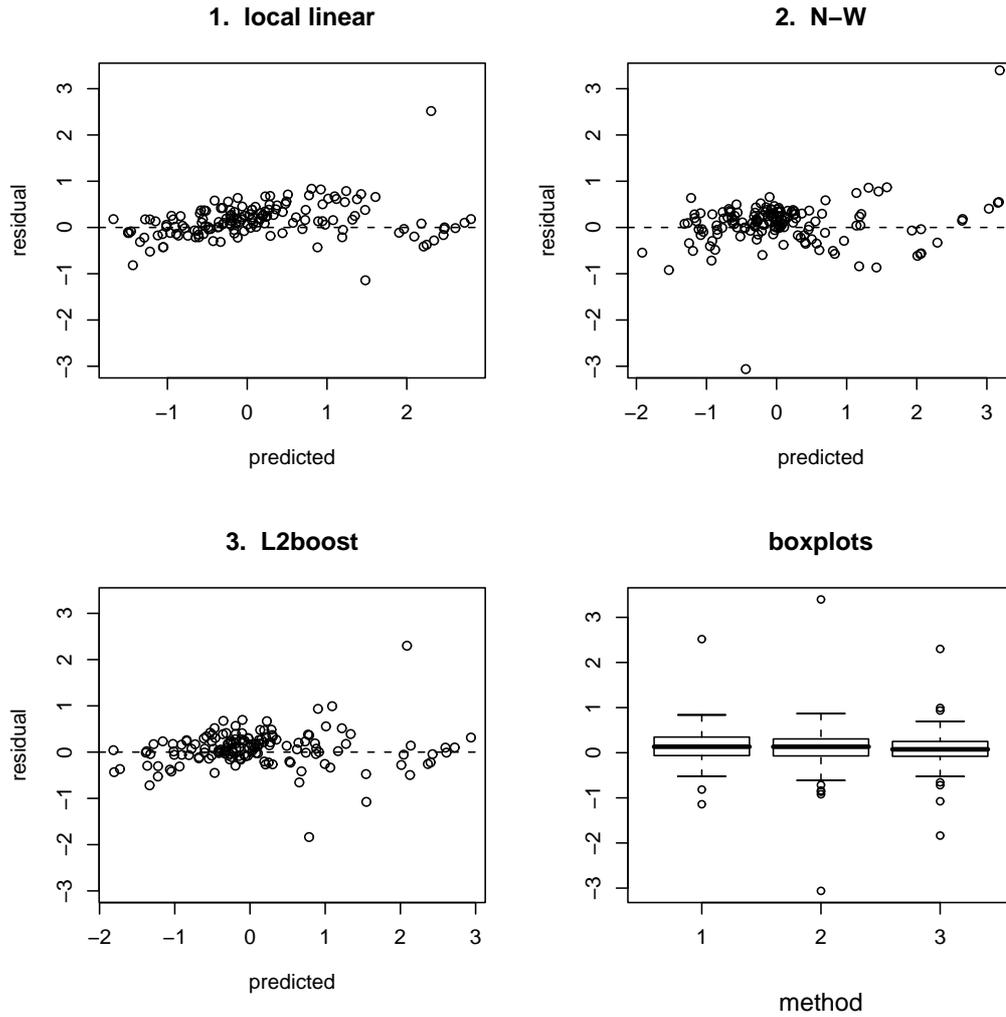


Fig. 4. Residual plots of the Boston housing test data corresponding to the three best models in Table 5.

6 Discussion

6.1 Alternative generalizations

Let $\mathbf{h}^T := (h_0, \dots, h_B)$, $\mathbf{w}^T := (w_0, \dots, w_B)$ be vectors of smoothing parameters and weights respectively, then

$$\mathcal{K}_{\mathbf{h}}^{\mathbf{w}} := \sum_{j=0}^B w_j K_{h_j}$$

is a weighted sum of kernel functions. The convolution kernel used in Section 3.2 is a special case of this formulation with $\mathbf{w} = (2, -1)$ and $\mathbf{h} = (h, \sqrt{2}h)$ and $B = 1$.

We can thus generalize \widehat{m}_2^* to

$$\widehat{m}_g(x; S, \mathbf{w}, \mathbf{h}) := \sum_{j=0}^B w_j \widehat{m}_{\text{NW}}(x; S, h_j). \quad (8)$$

This is simply a linear combination of N-W estimators, each with its own bandwidth. Similar proposals were considered by Rice (1984) and Jones (1993) who used weighted combinations of kernels to improve estimators at the boundary. In order for \widehat{m}_g to be asymptotically unbiased we require $\sum w_j = 1$. Given a vector of bandwidths \mathbf{h} we can choose the w_j to eliminate the bias terms which arise as a consequence of μ_k .

Since we have used a normal kernel, for a given bandwidth h we have $\mu_{2k} \propto h^{2k}$, and $\mu_{2k-1} = 0$ so a simple approach to obtain the weights w_j would be to set $\mathbf{h}^T = (h, \sqrt{c}h, \dots, c^{B/2}h)$ for some c , and then to solve $\mathbf{C}\mathbf{w} = (1, 0, \dots, 0)^T$ for \mathbf{w} , where (for $B \geq 1$)

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & c & \dots & c^B \\ \vdots & \vdots & \vdots & \vdots \\ 1 & c^B & \dots & c^{2B-1} \end{pmatrix}$$

and this simplification requires the selection of only two parameters (c and h), for a given B . Note that the above convolution kernel K_h^* uses $c = 2$, and that the solution for \mathbf{w} gives the desired value $(2, -1)$.

As an alternative approach, we could consider obtaining the w_j by ordinary least squares regression, i.e. obtain \mathbf{w} from $\widehat{\mathbf{w}} := (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$ where $\mathbf{Y}^T := (Y_1, \dots, Y_n)$ is the vector of responses, and the j th column of the matrix \mathbf{X} is given by $(\widehat{m}_{\text{NW}}(X_1; S, h_j), \dots, \widehat{m}_{\text{NW}}(X_n; S, h_j))^T$. This approach could also allow for the selection of B through standard techniques in stepwise regression. Also note the connection between (8) and a radial basis function (RBF) representation. In this framework the w_j are the weights, and $m_{\text{NW}}(x; S, h_j)$ act as ‘‘basis functions’’ which are themselves a weighted sum of basis functions. So this formulation is equivalent to a generalized RBF network, in which an extra layer is used to combine estimates, but with many of the weights being fixed.

6.2 Conclusions

We have discussed a multistep kernel regression smoother generated by learning the N-W estimator by L_2 boosting. Our main result is that the bias of $L_2\text{BoostNW}$ decreases exponentially fast towards zero, while the variance increases exponentially slow towards σ^2 and consequently could beat the overall MISE performance

of the ordinary kernel methods in regression. Our experiments show that this superiority occurs for several settings of (h, B) , and also that cross-validation can be successfully used for parameter selection. It is clear that the optimal bandwidth for boosting is greater than the values provided by the standard selection theories. Finally, note that our method is easily extended to multivariate data.

Acknowledgements

The authors are grateful to the Associate Editor and two anonymous referees for their valuable comments which led to considerable improvements in this article.

Appendix

Proof of Theorem 1: Following Bochner's theorem (see e.g. Lax, 2002, p. 144), $\text{spec}(\mathbf{K}) \subset [0, +\infty)$ if and only if K is a Fourier-Stieltjes transform of a finite measure. Due to symmetry and unimodality, $K_h(0) > K_h(x_i - x_j)$ for each $i \neq j$, so $\det \mathbf{K} > 0$. Clearly, $\text{spec}(\mathbf{N}^{1/2} \mathbf{K} \mathbf{N}^{1/2}) = \text{spec}(\mathbf{N} \mathbf{K})$, but $\mathbf{N} \mathbf{K}$ is row stochastic, so apply the Perron-Frobenius theorem for which

$$1 = \min_i \left\{ \sum_{j=1}^n (\mathbf{N} \mathbf{K})_{ij} \right\} \leq \max \text{spec}(\mathbf{N} \mathbf{K}) \leq \max_i \left\{ \sum_{j=1}^n (\mathbf{N} \mathbf{K})_{ij} \right\} = 1$$

and conclude that $\text{spec}(\mathbf{N} \mathbf{K}) \subset (0, 1]$. Finally, $\text{trace}(\mathbf{N} \mathbf{K}) < n$ yields $\min \text{spec}(\mathbf{N} \mathbf{K}) < 1$. \square

Lemma 1 (Bühlmann & Yu, 2003) *Consider linear smoothing by a hat matrix \mathbf{L} with characteristic roots ρ_1, \dots, ρ_n . Operate L_2 boosting with weak learner \mathbf{L} . Then L_2 boosting at step $b \geq 0$ is a linear smoother as well, whose hat matrix is equal to $\mathbf{I} - (\mathbf{I} - \mathbf{L})^{b+1}$.*

Proof: The residual vector at step $b \in [1, \dots, B]$, denoted as \mathbf{e}_b , can be written as

$$\mathbf{e}_b = \mathbf{y} - \widehat{\mathbf{m}}_{b-1} = \mathbf{e}_{b-1} - \mathbf{L} \mathbf{e}_{b-1} = (\mathbf{I} - \mathbf{L}) \mathbf{e}_{b-1}$$

implying $\mathbf{e}_b = (\mathbf{I} - \mathbf{L})^b \mathbf{y}$ for $b \in [1, \dots, B]$. Since $\widehat{\mathbf{m}}_0 = \mathbf{L} \mathbf{y}$, using a telescope-sum argument, we obtain

$$\widehat{\mathbf{m}}_b = \sum_{j=0}^b \mathbf{L} (\mathbf{I} - \mathbf{L})^j \mathbf{y} = (\mathbf{I} - (\mathbf{I} - \mathbf{L})^{b+1}) \mathbf{y}. \quad \square$$

Proof of Theorem 2: From Lemma 1 it follows that the $L_2\text{BoostNW}$ fit is $\widehat{\mathbf{m}}_b = \mathbf{M}_b \mathbf{y}$. Now, the hat matrix \mathbf{M}_b can be written as

$$\mathbf{M}_b = \mathbf{U} \mathbf{D}_b \mathbf{U}^{-1}$$

where $\mathbf{D}_b := \text{diag}(1 - (1 - \lambda_1)^{b+1}, \dots, 1 - (1 - \lambda_n)^{b+1})$; $\lambda_1, \dots, \lambda_n$, are the characteristic roots of $\mathbf{N}\mathbf{K}$, which are real due to theorem 1. As a consequence, the matrix \mathbf{U} , formed by the characteristic vectors of \mathbf{M}_b , has real entries. Notice that $\mathbf{N}\mathbf{K}$ is not symmetric, therefore $\mathbf{U}\mathbf{U}^T \neq \mathbf{I}$. Now

$$\begin{aligned} \text{bias}^2(\widehat{\mathbf{m}}_b; \mathbf{m}) &= (\mathbb{E}[\mathbf{M}_b \mathbf{y}] - \mathbf{m})^T (\mathbb{E}[\mathbf{M}_b \mathbf{y}] - \mathbf{m}) \\ &= ((\mathbf{M}_b - \mathbf{I})\mathbf{m})^T ((\mathbf{M}_b - \mathbf{I})\mathbf{m}) \end{aligned}$$

where

$$\begin{aligned} \mathbf{M}_b - \mathbf{I} &= \mathbf{U}(\mathbf{D}_b - \mathbf{I})\mathbf{U}^{-1} \\ &= \mathbf{U} \text{diag}(-(1 - \lambda_1)^{b+1}, \dots, -(1 - \lambda_n)^{b+1})\mathbf{U}^{-1}. \end{aligned}$$

The covariance matrix is

$$\text{cov}(\mathbf{M}_b \mathbf{y}) = \mathbf{M}_b \text{cov}(\mathbf{y}) \mathbf{M}_b^T = \sigma^2 \mathbf{M}_b \mathbf{M}_b^T = \sigma^2 \mathbf{U} \mathbf{D}_b \mathbf{U}^{-1} (\mathbf{U}^{-1})^T \mathbf{D}_b \mathbf{U}^T,$$

so the variance is

$$\text{var}(\widehat{\mathbf{m}}_b; \mathbf{m}) = \text{trace}(\text{cov}(\mathbf{M}_b \mathbf{y})).$$

Assume that $\text{spec}(\mathbf{N}\mathbf{K}) \subset (0, 1]$. For any $k \in [1, \dots, n]$ the bias order is $O\{(1 - \lambda_k)^{b+1}\}$, while the variance order is $O\{(1 - (1 - \lambda_k)^{b+1})^2\}$. So bias converges exponentially fast and variance converges exponentially slow. \square

Proof of Theorem 3: We have

$$\widehat{m}_1(x; h) = \frac{2\widehat{r}(x; h) - n^{-1} \sum_{i=1}^n K_h(x - X_i) \widehat{m}_0(X_i; h)}{\widehat{f}(x; h)}.$$

Now take the expectation of the numerator and denominator. The expectation of the second term in the numerator can be written as

$$\begin{aligned} \mathbb{E} [K_h(x - X_1) \widehat{m}_0(X_1; h)] &= \mathbb{E} \left[\frac{1}{nh} K_h(x - X_1) \sum_{j=1}^n K_h(X_1 - X_j) Y_j / \widehat{f}(X_1; h) \right] \\ &= \frac{1}{h} \iiint K_h(x - u) K_h(u - v) y f(y | v) \\ &\quad \times \left\{ f(u) + \frac{h^2}{2} \mu_2 f''(u) + o(h^2) \right\}^{-1} f(u) f(v) dy du dv \\ &= \frac{1}{h} \iint K_h(x - u) K_h(u - v) m(v) \\ &\quad \times \left\{ 1 + \frac{h^2 \mu_2 f''(u)}{2f(u)} + o(h^2) \right\}^{-1} f(v) du dv \end{aligned}$$

where the second line was obtained by ignoring the non-stochastic term in the sum (when $j = i$), and the third one uses the fact that $m(v) = \int y f(y | v) dy$.

Making the change of variable $t = (v - u)/h$ and expanding $m(v) = m(u + ht)$ and $f(v) = f(u + ht)$ in a Taylor series we get the following expansion up to terms of order $O(h^2)$

$$\begin{aligned} \mathbb{E} [K_h(x - X) \widehat{m}_0(X; h)] &\approx \iint K_h(x - u) K(t) \left\{ m(u) + thm'(u) + \frac{t^2 h^2 m''(u)}{2} \right\} \\ &\quad \times \left\{ 1 - \frac{h^2 \mu_2 f''(u)}{2f(u)} \right\} \left\{ f(u) + thf'(u) + \frac{t^2 h^2 f''(u)}{2} \right\} dt du \\ &= \int K_h(x - u) \left[r(u) + \frac{h^2 \mu_2}{2} \{ r''(u) - m(u) f''(u) \} \right] du \\ &\tag{9} \\ &= r(x) + h^2 \mu_2 r''(x) - \frac{h^2 \mu_2}{2} m(x) f''(x). \tag{10} \end{aligned}$$

The RHS of Equation (9) was obtained on simplification and recalling that $r''(u) = m''(u)f(u) + 2m'(u)f'(u) + m(u)f''(u)$; the RHS of Equation (10) has been obtained by making a second change of variable $w = (x - u)/h$, expanding in a Taylor series, and integrating.

To obtain the expectation of the numerator of $\widehat{m}_1(x; h)$, we multiply Equation (3) by 2, then subtract the RHS of Equation (10) to get

$$\mathbb{E} \left[2\widehat{r}(x; h) - \frac{1}{n} \sum_{i=1}^n K_h(x - X_i) \widehat{m}_0(X_i; h) \right] \approx r(x) + \frac{h^2 \mu_2}{2} m(x) f''(x).$$

Finally we divide this by the approximate (up to the second order) expectation of the denominator of $\widehat{m}_1(x; h)$ which is given in Equation (4). We can thus write the following expression for the asymptotic expectation up to terms of order $O(h^2)$

$$\begin{aligned} \mathbb{E} \widehat{m}_1(x; h) &\approx \left\{ r(x) + \frac{h^2 \mu_2}{2} m(x) f''(x) \right\} \left[\frac{1}{f(x)} \left\{ 1 + \frac{h^2 \mu_2 f''(x)}{2f(x)} \right\}^{-1} \right] \\ &= \frac{r(x)}{f(x)} \left\{ \frac{2f(x)}{2f(x) + h^2 \mu_2 f''(x)} + \frac{h^2 \mu_2 f''(x)}{2f(x) + h^2 \mu_2 f''(x)} \right\} \\ &= m(x). \end{aligned}$$

The variance of $\widehat{m}_1(x; h)$ can be written as

$$\text{var } \widehat{m}_0(x; h) + 2 \text{cov}(\widehat{m}_0(x; h), \widehat{m}_{\text{NW}}(x; S_e, h)) + \text{var } \widehat{m}_{\text{NW}}(x; S_e, h)$$

where S_e are the residuals of the first fit. Now since $\text{var } \widehat{m}_{\text{NW}}(x; S_e, h) \leq \text{var } \widehat{m}_0(x; h)$ we have the result.

For the conditional version of the variance result, we could use Theorem 2 substituting $b = 1$ and $b = 0$ into the expression for ave-var. Since $\lambda_k < 1$, we have

$$(2\lambda_k - \lambda_k^2)^2 < 4\lambda_k^2 < 4. \quad \square$$

References

- Breiman, L. (1997). Arcing the edge. Technical Report 486, Dept. Statistics, Univ. California, Berkeley.
- Breiman, L. and J. Friedman (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association* 80, 580–598.
- Bühlmann, P. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics* 34, 559–583.
- Bühlmann, P. and B. Yu (2003). Boosting with the L_2 loss: regression and classification. *Journal of the American Statistical Association* 98, 324–339.
- Chaudhuri, P., K. Doksum, and A. Samarov (1997). On average derivative quantile regression. *The Annals of Statistics* 25, 715–744.
- Di Marzio, M. and C. C. Taylor (2004). Boosting kernel density estimates: a bias reduction technique? *Biometrika* 91, 226–233.
- Di Marzio, M. and C. C. Taylor (2005). Kernel density classification and boosting: an L_2 analysis. *Statistics and Computing* 15, 113–123.
- Doksum, K. and A. Samarov (1995). Nonparametric estimation of global functionals and a measure of explanatory power of covariates in regression. *The Annals of Statistics* 23, 1443–1473.
- Fan, J. and I. Gijbels (1996). *Local polynomial modelling and its applications*. Chapman & Hall, London.
- Freund, Y. (1995). Boosting a weak learning algorithm by majority. information and computation. *Information and computation* 121, 256–285.
- Freund, Y. and R. Schapire (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148–156. Morgan Kaufman, San Francisco.
- Friedman, J., Hastie, T., and R. Tibshirani (2001). Additive logistic regression: a statistical view of boosting. *The Annals of Statistics* 28, 337–407.
- Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *The Annals of Statistics* 29, 1189–1232.
- Härdle, W. (1990). *Applied nonparametric regression*. Cambridge University Press.
- Harrison, D. and D. Rubinfeld (1978). Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management* 5, 81–102.
- Hastie, T. and C. Loader (1993). Local regression: automatic kernel carpentry. *Statistical Science* 8, 120–143.
- Jiang, W. (2004). Process consistency for adaboost. *The Annals of Statistics* 32, 13–29.

- Jones, M. C. (1993). Simple boundary correction for kernel density estimation. *Statistics and Computing* 3, 135–146.
- Jones, M. C., O. Linton, and J. P. Nielsen (1995). A simple bias reduction method for density estimation. *Biometrika* 82, 327–38.
- Lax, P. D. (2002). *Functional Analysis*. Wiley, New York.
- Lugosi, G. and N. Vayatis (2004). On the bayes-risk consistency of regularized boosting methods. *The Annals of Statistics* 32, 30–55.
- Müller, H. G. (1993). Comment on “Local regression: Automatic kernel carpentry” by T. Hastie and C. Loader. *Statistical Science* 8, 120–143.
- Rice, J. A. (1984). Boundary modifications for kernel regression. *Comm. Statist. Theory Meth.* 13, 893–900.
- Schapire, R. (1990). The strength of weak learnability. *Machine Learning* 5, 197–227.
- Stuetzle, W. and Y. Mittal (1979). Some comments on the asymptotic behavior of robust smoothers. In *Smoothing Techniques for Curve Estimation. Proceedings, Heidelberg 1979*, Lecture Notes in Mathematics 757, pp. 191–195. Springer-Verlag, Berlin.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley, Philippines.
- Zhang, T. (2004). Statistical behaviour and consistency of classification methods based on convex risk minimization. *The Annals of Statistics* 32, 56–85.
- Zhang, T. and B. Yu (2005). Boosting with early stopping: convergence and consistency. *The Annals of Statistics* 33, 1538–1579.