Stat Comput (2016) 26:93–105
DOI 10.1007/s11222-014-9474-0

Wavelet-based gradient boosting

E. Dubossarsky · J. H. Friedman · J. T. Ormerod · M. P. Wand

Received: 13 November 2012 / Accepted: 15 April 2014 / Published online: 8 May 2014

Abstract A new data science tool named *wavelet-based gradient boosting* is proposed and tested. The approach is special case of componentwise linear least squares gradient boosting, and involves wavelet functions of the original predictors. Wavelet-based gradient boosting takes advantages of the approximate ℓ_1 penalization induced by gradient boosting to give appropriate penalized additive fits. The method is readily implemented in R and produces parsimonious and interpretable regression fits and classifiers.

Keywords Classification · Data science · Generalized additive models · Nonparametric regression

1 Introduction

We propose a new data science method named *wavelet-based gradient boosting*. The essence of the method is to use wavelet basis functions of candidate predictors and simple linear regression base learners within the gradient boosting paradigm introduced by Friedman (2001), and recently surveyed by Bühlmann and Hothorn (2007)). The inherent sparseness of gradient boosting [e.g. Bühlmann (2006)] is ideal for wavelet basis functions.

The mechanics of wavelet-based gradient boosting are the same as *componentwise linear least squares* gradient boosting, as described in Sect. 4.1 of Bühlmann and Hothorn (2007). The only difference is that each of the d continuous predictors is replaced by K wavelet basis functions of the predictor—resulting in a new and enlarged set of dK predictors. We specifically recommend the default wavelet basis developed in Wand and Ormerod (2011).

Recently, Leitenstorfer and Tutz (2007) investigated the use of componentwise linear least squares to select radial basis functions with varying scales for nonparametric regression. Our approach is similar in spirit, with wavelet basis functions being used instead of radial basis functions. However, wavelet bases have the following two attractions when used in gradient boosting:

- Their localness means that sparse ℓ_1 penalization is appropriate [e.g. Donoho and Johnstone (1994)]. As shown by Efron et al. (2004), gradient boosting with small learning rate is approximately equivalent to ℓ_1 penalization.
- They provide efficient approximations of jagged and jumpy predictor effects. This has the potential to improve prediction when such effects are present in the population sense.

We have tested wavelet-based gradient boosting on both simulated and actual data and found it to be effective, and a useful addition to the data science arsenal. In particular, the simulation study given in Sect. 3.3 shows that wavelet-based

E. Dubossarsky
Presciient Pty. Ltd, Epping, Australia
http://presciient.com
e-mail: enquiries@presciient.com

J. H. Friedman
Department of Statistics, Stanford University,
Stanford, CA 94305, USA
e-mail: jhf@stanford.edu

J. T. Ormerod
School of Mathematics and Statistics, University of Sydney,
Sydney, NSW 2006, Australia
e-mail: jormerod@sydney.edu.au

M. P. Wand (✉)
School of Mathematical Sciences, University of Technology, Sydney,
Broadway, Ultimo, NSW 2007, Australia
e-mail: matt.wand@uts.edu.au

gradient boosting offers improved performance when there are jagged or jumpy predictor effects.

Whilst potential use in data science applications are the primary motivation for wavelet-based gradient boosting, it also constitutes a new approach to wavelet-based nonparametric regression (e.g. Vidakovic 1999;Nason 2008) and generalized additive model fitting. The last decade has seen several proposals that have involved the use of gradient boosting in generalized additive model analysis [e.g. Binder and Tutz (2008)].

The R computing environment (R Development Core Team 2012) provides good support for gradient boosting, via packages such as **gbm** (Ridgeway 2012) and **mboost** (Hothorn et al. 2011). Each of these packages have functionality that supports wavelet-based gradient boosting, and our illustrations make use of the function `glmboost()` in **mboost**.

Section 2 details wavelet-based gradient boosting. Illustrative examples and the results of a simulation study are given in Sect. 3. Concluding remarks are made in Sect. 4. An appendix contains some technical details required visualization of wavelet-based gradient boosting outputs.

2 Details of wavelet-based gradient boosting

Wavelet-based gradient boosting applies componentwise linear least squares gradient boosting (Bühlmann and Hothorn 2007, Sect. 5.1) to wavelet basis functions of the predictors. In this section we describe the algorithm's inputs and mechanics.

2.1 Inputs

The primary inputs for wavelet-based gradient boosting are:

- A regression-type data set,
- A loss function.

Secondary inputs are:

- The maximal number of boosting iterations,
- The learning rate parameter,
- The wavelet basis size.

Each of these is now briefly explained.

2.1.1 Regression-type data

Regression-type data take the form: $\boldsymbol{y}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$. Here $\boldsymbol{y}$ is an $n \times 1$ vector of response variables or class indicators and, for each $1 \leq \ell \leq d$, $\boldsymbol{x}_\ell$ contains values on the ℓth predictor x_ℓ. For now we assume that each of these predictors are continuous. Section 2.7 explains the extension to mixed predictor data.

2.1.2 Loss function

For any two equal-length vectors $\boldsymbol{y}$ and $\boldsymbol{f}$, let $\mathcal{L}(\boldsymbol{y}, \boldsymbol{f})$ be a non-negative number that reflects the distance between $\boldsymbol{y}$ and $\boldsymbol{f}$. Common examples are

$$\mathcal{L}(\boldsymbol{y}, \boldsymbol{f}) = \begin{cases} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{f}\|^2 \\ \text{(squared error loss)} \\ \log_2[1+\exp\{-2(2\boldsymbol{y}-\boldsymbol{1})^T\boldsymbol{f}\}] \\ \text{(binomial log-likelihood loss).} \end{cases} \tag{1}$$

Squared error loss is usually appropriate when the entries of the response vector $\boldsymbol{y}$ are continuous. If $\boldsymbol{y}$ has binary entries (i.e. $y_i \in \{0, 1\}$) then binomial log-likelihood loss is usually preferred. Binomial log-likelihood loss, with logarithm to base 2 rather than natural logarithm, is used because it constitutes an upper bound on the misclassification error. This version is compatible with the R package **glmboost** (Hothorn et al. 2011) discussed in Sect. 2.5.

Gradient boosting requires that the vector of elementwise partial derivatives with respect to the second argument, denoted by $\frac{\partial}{\partial \boldsymbol{f}}\mathcal{L}(\boldsymbol{y}, \boldsymbol{f})$, is well-defined. For the examples given in (1) we have

$$\frac{\partial}{\partial \boldsymbol{f}}\mathcal{L}(\boldsymbol{y}, \boldsymbol{f}) = \begin{cases} \boldsymbol{y} - \boldsymbol{f} \\ \text{(squared error loss)} \\ \dfrac{-2(2\boldsymbol{y} - \boldsymbol{1})}{\log_e(2)[1+\exp\{2(2\boldsymbol{y}-\boldsymbol{1})^T\boldsymbol{f}\}]} \\ \text{(binomial log-likelihood loss).} \end{cases}$$

2.1.3 Maximal number of boosting iterations

The maximal number of boosting iterations, denoted by M, is a positive integer usually in the hundreds or thousands. Appropriate choice of M depends on the data set at hand. Further discussion about the choice of M, in relation to estimation of the optimal stopping position, is discussed in Sect. 2.3.

2.1.4 Learning rate

The *learning rate*, denoted by ν, is a parameter between 0 and 1 and is also known as the *step-length factor*. Provided ν is small, this parameter is of minor importance (Bühlmann and Hothorn 2007). A reasonable default setting is $\nu = 0.1$.

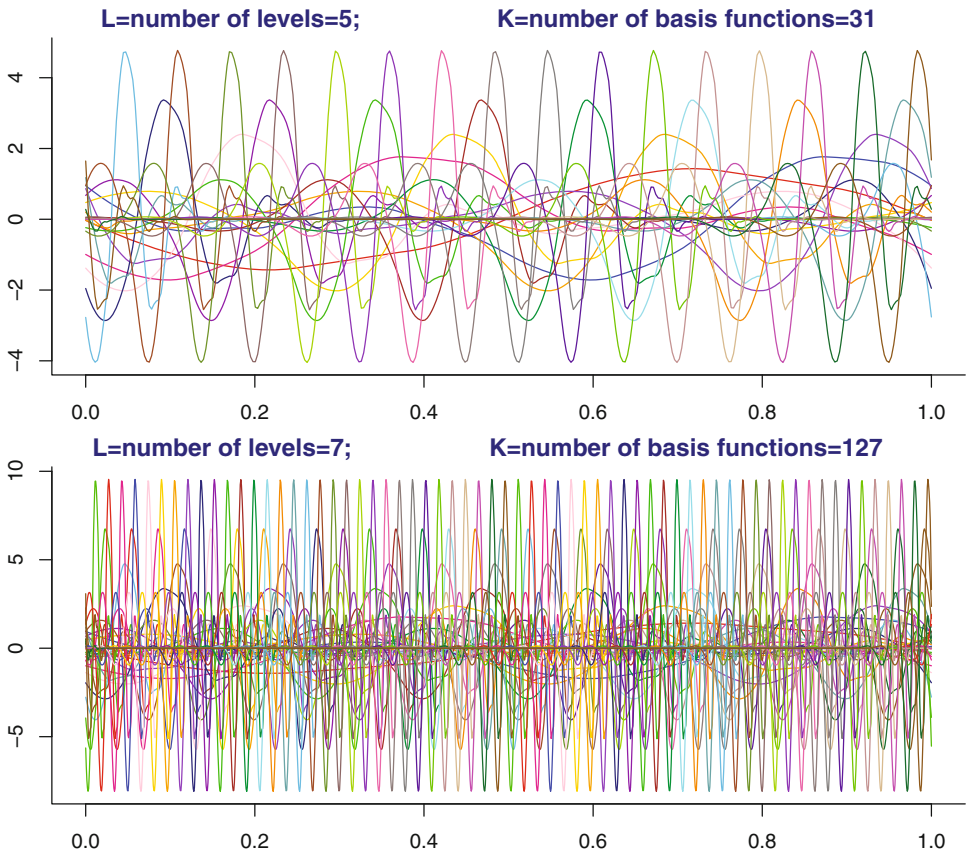

Fig. 1 Wavelet basis functions on [0, 1] for two values of the number of levels parameter L

2.1.5 Wavelet basis size

This parameter is the number of wavelet basis functions per predictor, and denoted here by K, and must satisfy $K = 2^L - 1$ for L a positive integer. For simplicity we take K to be fixed across each of the d predictors. The extension to variable K is straightforward, but is omitted to maintain presentational simplicity. As indicated by Fig. 10 of Wand and Ormerod (2011), the choice of K has a minor effect on the resulting function estimates, assuming that it is reasonably large. We have found $K = 127$ to be a good default.

2.2 Wavelet bases construction

Section 3.1 of Wand and Ormerod (2011) describes generation of a default wavelet basis function matrix $\boldsymbol{Z}_{\boldsymbol{x}}$ from a general univariate sample $\boldsymbol{x} \equiv (x_1, \ldots, x_n)$. The (i, k) entry of $\boldsymbol{Z}_{\boldsymbol{x}}$ is

$$z_k(x_i), \quad 1 \le i \le n, \ 1 \le k \le K,$$

where $\{z_k(\cdot) : 1 \le k \le K\}$ is a set of wavelet basis functions over the range of the x_is and $K = 2^L - 1$ for an integer L called the *level*. Figure 1 shows the z_k functions based on the Daubechies 5 mother wavelet for varying values of L. R software for efficient computation of $\boldsymbol{Z}_{\boldsymbol{x}}$ is described in Sect. 2.5. Note that this basis is such that it should be accompanied by a constant function for the full fit. This differs from popular spline bases, which often require low degree polynomial functions to be included in addition to the constant function.

For now let $K = 2^L - 1$ be constant across all input predictor variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$. Then horizontally concatenate these matrices to obtain

$$\boldsymbol{Z} = [\boldsymbol{Z}_{\boldsymbol{x}_1} \cdots \boldsymbol{Z}_{\boldsymbol{x}_d}]. \tag{2}$$

Note that $\boldsymbol{Z}$ is an $n \times dK$ matrix containing a "super dictionary" of wavelet basis functions evaluated at the observed predictor values.

2.3 Akaike information criteria for optimal stopping estimation

Akiake information criteria (AIC) come in various forms that trade off deviance and effective degrees of freedom measures. Consider a wavelet-based vector of fitted values

$$\widehat{\boldsymbol{f}} \equiv \widehat{\alpha}\mathbf{1} + \boldsymbol{Z}\widehat{\boldsymbol{u}}$$

where $\mathbf{1}$ is the $n \times 1$ vector of ones, $\boldsymbol{Z}$ is as given in (2), $\widehat{\alpha}$ is a scalar and $\widehat{\boldsymbol{u}}$ is a $dK \times 1$ vector of wavelet coefficients.

Then the classical definition of AIC is

$$\mathrm{AIC}(\widehat{\boldsymbol{f}}) \equiv 2\,\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{f}}) + 2\,\mathrm{edf}(\widehat{\boldsymbol{f}}) \tag{3}$$

where $\mathrm{edf}(\widehat{\boldsymbol{f}})$ is a measure of the *effective degrees of freedom* of $\boldsymbol{f}$. A simple and appropriate effective degrees of freedom measure is

$$\mathrm{edf}(\widehat{\boldsymbol{f}}) \equiv 1 + \text{number of non-zero entries in } \widehat{\boldsymbol{u}}. \tag{4}$$

The comment by Hastie (2007) on Bühlmann and Hothorn (2007) and the authors' rejoinder contains some interesting discussion on appropriate effective degrees of freedom measures for gradient boosting in general. For wavelet-based gradient boosting, (4) has the advantage that it scales well to large input data sets. It also has attractive unbiasedness properties for the ℓ_1-type penalization induced by gradient boosting (Zou et al. 2007).

In the special case of squared error loss Hurvich et al. (1998) have identified some drawbacks with the classical AIC formula and proposed the following *corrected* AIC:

$$\mathrm{AIC}_C(\widehat{\boldsymbol{f}}) \equiv \log\{2\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{f}})/n\} + 1 + \frac{2\{\mathrm{edf}(\widehat{\boldsymbol{f}}) + 1\}}{n - \mathrm{edf}(\widehat{\boldsymbol{f}}) - 2}. \tag{5}$$

Section 5.4 of Bühlmann and Hothorn (2007) contains relevant discussion on the choice of model selection criterion. Apart from (3) and (5), they also mention the g-prior based minimum description length criterion of Hansen and Yu (2001) as a possibility for squared error loss. In lieu of a full investigation into the relative merits of the various model selection criteria for wavelet gradient boosting, we use AIC_C is used in our example involving squared error loss (Sect. 3.1).

2.4 Algorithm

The wavelet-based gradient boosting algorithm with inputs as described in Sect. 2.1 as Algorithm 1. The following notation is used in Algorithm 1:

- $\boldsymbol{z}_j \equiv j$ th column of $\boldsymbol{Z}$, $\quad 1 \le j \le dK$,
- $\widehat{u}_j^{(m)} \equiv j$th entry of $\widehat{\boldsymbol{u}}^{(m)}$, $\quad 1 \le j \le dK$,
- $\|\boldsymbol{v}\|^2 \equiv \boldsymbol{v}^T\boldsymbol{v}$ for a general column vector $\boldsymbol{v}$.

REMARKS:

1. In the case of squared error loss, it is common to instead use the corrected AIC expression (5) in the AIC updating step. The example in Sect. 3.1 uses corrected AIC.
2. As a proposed new data science tool, Algorithm 1 has the advantages of being computationally simple and scaling well to large data sets. All steps are arithmetic in nature and devoid of computational complexities such as convergence and root-finding. The storage and number of computations are linear in the sample size once d, K and M have been fixed.
3. The estimated coefficients, $\widehat{\alpha}$ and $\widehat{\boldsymbol{u}}^{(m^*)}$, are the minimally sufficient outputs of Algorithm 1. However, conversion of these outputs, in combination with the input data, to interpretable graphical summaries requires additional effort. Sections 2.8 and 3, as well as an appendix, tackle this issue.
4. The base learner in Algorithm 1 is single predictor "through-the-origin least squares regression", where the "x-variable" is a wavelet basis function of one of the original predictors.
5. The updates $\widehat{w}_j \leftarrow \boldsymbol{z}_j^T\boldsymbol{r}/\|\boldsymbol{z}_j\|^2, 1 \le j \le dK$ can be written in matrix notation as $\widehat{\boldsymbol{w}} \leftarrow \boldsymbol{Z}^T\boldsymbol{r}/\mathrm{diagonal}\,(\boldsymbol{Z}\boldsymbol{Z}^T)$ where $\mathrm{diagonal}(\boldsymbol{M})$ is the vector containing the diagonal entries of the square matrix $\boldsymbol{M}$ and the quotient of two vectors is applied in an element-wise fashion.
6. If m^* equals, or is close to, the maximal number of iterations M then it is likely that the global minimum of the AIC criterion has not been obtained. When this happens it is recommended that Algorithm 1 be re-run with a larger value of M.
7. The *Sparse L_2 Boost* algorithm of Bühlmann and Yu (2006), which yields sparser solutions than componentwise linear least squares gradient boosting, could be used as the basis for an alternative version of Algorithm 1. This is yet to be explored.

Algorithm 1 The wavelet-based gradient boosting algorithm

Inputs: Regression-type data $\boldsymbol{y}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$; loss function $\mathcal{L}(\cdot,\cdot)$; $M \in \mathbb{N}$; $0 < \nu \le 1$; $K = 2^L - 1$; $L \in \mathbb{N}$.

Construct: $\boldsymbol{Z} = [\boldsymbol{Z}_{\boldsymbol{x}_1} \cdots \boldsymbol{Z}_{\boldsymbol{x}_d}]$ ($n \times dK$ matrix containing wavelet basis functions of each predictor).

Initialize: $\boldsymbol{u}^{(0)} \leftarrow \boldsymbol{0}$; $\widehat{\alpha} \leftarrow \underset{\alpha\in\mathbb{R}}{\mathrm{argmin}}\,\mathcal{L}(\boldsymbol{y}, \alpha\boldsymbol{1})$; $\widehat{\boldsymbol{f}} \leftarrow \widehat{\alpha}\boldsymbol{1}$; $\mathcal{A} \leftarrow \{0\}$.

For $m = 1, \ldots, M$:

$\boldsymbol{r} \leftarrow \left[-\frac{\partial}{\partial \boldsymbol{f}}\mathcal{L}(\boldsymbol{y}, \boldsymbol{f})\right]_{\boldsymbol{f}=\widehat{\boldsymbol{f}}}$; $\widehat{w}_j \leftarrow \boldsymbol{z}_j^T\boldsymbol{r}/\|\boldsymbol{z}_j\|^2,\ 1 \le j \le dK$;

$j^\star \leftarrow \underset{1\le j\le dK}{\mathrm{argmin}}\,\|\boldsymbol{r} - \widehat{w}_j\,\boldsymbol{z}_j\|^2$; $\widehat{u}_{j^*}^{(m)} \leftarrow \widehat{u}_{j^*}^{(m-1)} + \nu\,\widehat{w}_{j^*}$;

$\widehat{\boldsymbol{f}} \leftarrow \widehat{\boldsymbol{f}} + \widehat{u}_{j^\star}\,\boldsymbol{z}_{j^\star}$; $\mathcal{A} \leftarrow \mathcal{A} \cup \{j^*\}$;

$\mathrm{edf}(\widehat{\boldsymbol{f}}) \leftarrow$ cardinality of $\mathcal{A}$;

$\mathrm{AIC}^{(m)} \leftarrow 2\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{f}}) + 2\,\mathrm{edf}(\widehat{\boldsymbol{f}})$.

$m^* \leftarrow \underset{1\le m\le M}{\mathrm{argmin}}\,\mathrm{AIC}^{(m)}$.

Outputs: $\widehat{\alpha}$, $\widehat{\boldsymbol{u}}^{(m^*)}$.

2.5 R implementation

A practical advantage of Algorithm 1 is that it can be implemented efficiently and relatively painlessly in R (R Develop-

ment Core Team 2012) using existing R functions `ZDaub()` (Wand and Ormerod 2011) and `glmboost()` in the package **mboost** (Hothorn et al. 2011). The code for `ZDaub()` is part of the supplementary materials that accompany Wand and Ormerod (2011) on the *Electronic Journal of Statistics* web-site. This function efficiently computes the $\boldsymbol{Z}_{\boldsymbol{x}_j}$ matrices needed to build the wavelet dictionary matrix $\boldsymbol{Z}$. Efficiency is afforded by the Discrete Wavelet Transform and its implementation in the R package **wavethresh** (Nason 2010).

The following code illustrates this for an $n = 1000, d = 3$ simulated regression data set with $K = 127$ basis functions per predictor and maximal number of iterations $M = 5000$, assuming that `ZDaub()` and `glmboost()` have been made available to the current session:

```
x1 <- runif(1000) ; x2 <- runif(1000) ; x3 <- runif(1000)
y <- (2.2*(4*sin(4*pi*x1)-sign(x1-0.3)-sign(0.72-x1))
     +10*sign(x2-(1/3))-7*sign(x2-(2/3))+rnorm(1000))
Z1 <- ZDaub(x1,range.x=c(0,1),numLevels=7)
Z2 <- ZDaub(x2,range.x=c(0,1),numLevels=7)
Z3 <- ZDaub(x3,range.x=c(0,1),numLevels=7) ; Z <- cbind(Z1,Z2,Z3)
glmboostObj <- glmboost(Z,y,center=FALSE,
              control=boost_control(mstop=5000))
AICobj <- AIC(glmboostObj,method=``corrected'',df=``actset'')
mStar <- mstop(AICobj)
uHatmStar <- as.numeric(extract(glmboostObj[mStar],
                        ``coefficients'',which=1:ncol(Z)))
par(mfrow=c(2,2)) ; plot(AICobj) ; plot(x1,Z1%*%uHatmStar[1:127])
plot(x2,Z2%*%uHatmStar[128:254]) ; plot(x3,Z3%*%uHatmStar[255:381])
```

The plots of the fits produced by this code are rudimentary. Improved display is illustrated in Sect. 3.

We also note that `glmboost()` supports several loss functions, including those corresponding to the Poisson, Negative Binomial and Laplace log-likelihoods as well as the Cox proportional hazards model and Huber M-estimation. A full listing can be obtained via the R commands `library(mboost) ; help(Family)`.

2.6 Aside: nonparametric regression and generalized additive model fitting

Even though data science is our main focus, it is worth noting—as an aside—that Algorithm 1 constitutes new nonparametric regression and generalized additive model fitting methodology.

2.6.1 Nonparametric regression

Consider the nonparametric regression setting

$$y_i = f(x_i) + \varepsilon_i, \quad 1 \le i \le n,$$

where (x_i, y_i) is the observed data, f is a unknown regression mean function and the ε_i are independent and approximately Gaussian with mean zero. Donoho (1995) describes conditions on f appropriate for wavelet-based nonparametric regression. Such functions could be quite jagged and have jump discontinuities.

Algorithm 1 with $d = 1$ and squared error loss is an effective means of estimating f because it returns a sparse solution akin to ℓ_1 penalization. The following R code illustrates this for an $n = 2,000$ regression data-set generated from the *heavisine* function (Donoho and Johnstone 1994) on [0, 1]:

```
x <- (1:2000)/2000
fTrue <- function(x)
  return(2.2*(4*sin(4*pi*x)-sign(x-0.3)-sign(0.72-x)))
y <- fTrue(x) + rnorm(2000)
Z <- ZDaub(x,range.x=c(0,1),numLevels=7)
glmboostObj <- glmboost(Z,y,center=FALSE,family=Gaussian(),
                        control=boost_control(mstop=1000))
AICobj <- AIC(glmboostObj,method=``corrected'',df=``actset'')
mStar <- mstop(AICobj)
fhat <- predict(glmboostObj[mStar],new=Z)
plot(x,y,pch=``.'',col=``orange'') ; lines(x,fTrue(x),col=``blue'')
lines(x,fhat,col=``red'')
```

Non-Gaussian nonparametric regression can be handled via appropriate modification of the loss function used in Algorithm 1.

2.6.2 Generalized additive model fitting

Algorithm 1 also provides a new way to fit a generalized additive model such as

$$y_i \overset{\text{ind.}}{\sim} \text{Poisson}[\exp\{f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i})\}], \quad 1 \le i \le n. \tag{6}$$

Algorithm 1 with loss function $\mathcal{L}(\boldsymbol{y}, \boldsymbol{f}) = -\boldsymbol{y}^T + \mathbf{1}^{\mathbf{T}} \exp(\boldsymbol{f})$ is appropriate for fitting (6). The performance of this approach is outside the scope of the current article.

2.7 Issues regarding discrete and categorical predictors

As it stands Algorithm 1 is oblivious to whether the predictors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ correspond to continuous or discrete variables. Wavelet-based gradient boosting should still return reasonable fits when some of the predictors are discrete since many of the corresponding $\boldsymbol{Z}$ matrix columns are exactly zero. Such columns will seldom be selected and, if they are, will make no difference to the fit. Each of the examples in Sect. 3 adopt this strategy for ease of implementation.

Nevertheless, it is somewhat wasteful to carry along K basis functions of a highly discrete predictor variable when only a few of them are relevant. For very large problems, where storage and computation time at a premium, one could consider simpler wavelet functions. For example, if $\boldsymbol{x}_j$ is binary predictor with 0/1 coding then it may suffice to use the

Haar wavelet basis with $L = K = 1$. In this case $z_1(\boldsymbol{x}_j) = 2\boldsymbol{x}_j - 1$ is the corresponding column of the $\boldsymbol{Z}$ matrix. Similar schemes could be used for other highly discrete, non-binary, predictor variables. This is yet to been studied closely.

Algorithm 1 requires that all input predictor variables are numerical. A simple way around this problem is to introduce $l-1$ indicator variables for each l-level categorical variable. A potential shortcoming of this approach is dependence on the choice of indicator variable coding. Once again, we have not yet delved into this issue.

2.8 Relative importance scores

A simple and effective way of assessing the relative importance of the candidate predictors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ is to compare spread summaries of the fitted values vectors $\widehat{\boldsymbol{f}}_1, \ldots, \widehat{\boldsymbol{f}}_d$. First note that each $\widehat{\boldsymbol{f}}_j$ is formed from the output coefficient vector $\widehat{\boldsymbol{u}}^{(m^*)}$ via the calculation

$$\widehat{\boldsymbol{f}}_j = \boldsymbol{Z}_{\mathcal{I}_j} \widehat{\boldsymbol{u}}^{(m^*)}_{\mathcal{I}_j}$$

where $\{\mathcal{I}_1, \ldots, \mathcal{I}_d\}$ is the partition of the column indices of $\boldsymbol{Z}$ corresponding to the basis functions for each of the d predictors and $\boldsymbol{Z}_{\mathcal{I}_j}$ is the sub-matrix of $\boldsymbol{Z}$ with columns $\mathcal{I}_j$. A similar definition applies to sub-vectors of $\widehat{\boldsymbol{u}}^{(m^*)}$. If exactly K basis functions are used for each $\boldsymbol{x}_j$ then

$$\mathcal{I}_j = \{(j-1)K + 1, \ldots, jK\}, \quad 1 \le j \le d.$$

Under these definitions $\widehat{\boldsymbol{f}} = \widehat{\alpha}\boldsymbol{1} + \sum_{j=1}^{d} \widehat{\boldsymbol{f}}_j$ at the completion of Algorithm 1.

Any of the common spread summary statistics, such as standard deviation, interquartile range and median absolute deviation, could be used to form importance scores from the $\widehat{\boldsymbol{f}}_j$. Our default importance score for the jth predictor is

$$s_j \equiv \text{standard deviation of } \widehat{\boldsymbol{f}}_j \text{ entries,}$$

possibly with a small percentage (e.g. 2.5 %) of the ordered entries of $\widehat{\boldsymbol{f}}_j$ trimmed from each end. The jth *relative* importance score is then $r_j \equiv s_j \Big/ \sum_{j=1}^{d} s_j$, which we express as percentage in the examples to follow.

Relative importance scores provide a quick and simple way of achieving a parsimonious summary of the data. Those predictors with relative importance scores below a particular threshold, such as 1%, can be deemed unimportant and omitted from the fit.

2.9 Extensions to allow interactions

In its current form Algorithm 1 returns a fit that is additive in the predictors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$. Extensions that allow for interactions between the $\boldsymbol{x}_j$s could be contemplated, but at the cost of increased complexity and reduced interpretability. The wavelet basis functions discussed in Sect. 2.2 have natural and well-established extensions to higher dimensions (e.g. Vidakovic 1999; Nason 2008) which could be used to allow interactions. In principle, this simply involves wider $\boldsymbol{Z}$ matrices being used in Algorithm 1. However, there are certain to be pertinent practical issues, such trading-off between level of detail and computational and storage costs.

3 Illustrations

We now provide illustration of wavelet-based gradient boosting and demonstrate its data science abilities, for both the regression and classification contexts.

3.1 Sydney residential property prices

This illustration uses data on $n = 999$ residential property prices of houses that were sold in Sydney, Australia, during 2001. There data were assembled as part of an unpublished study by A. Chernih and M. Sherris at the University of New South Wales. Several predictors, pertaining to features such as geographical position, socio-economic status, proximity to amenities and air pollution levels, were also recorded. As a response variable we used

$$y_i = \log_e(\text{sale price of } ith \text{ house})$$

which exhibits approximately normality.

We fed these data into Algorithm 1 with the squared error loss function, maximal number of boosting iterations $M = 1000$, learning rate $\nu = 0.1$ and number of basis functions per predictor fixed at $K = 127$. Corrected AIC was used to estimate the optimal stopping position.

The right panel of Fig. 2 shows the resulting AIC path and the relative importance scores. As shown in the AIC plot, the estimated stopping position $m^* = 509$. The corresponding effective degrees of freedom is

$$\text{edf}(\widehat{\boldsymbol{f}}) = 1 + \text{number of non-zero entries in } \widehat{\boldsymbol{u}}^{(m^*)} = 141$$

which represents a substantial reduction compared with the $1 + 22 \times 127 = 2,795$ candidate basis functions.

Figure 2 also shows the ordered relative importance scores. The predictors `income`, `particulate matter 10` and `degrees longitude` stand out as being the most important predictors of sale price. The vertical dashed line corresponds to a threshold set at 1 % relative importance. Note that 9 of the 22 predictors are deemed unimportant according to this threshold—resulting in a more parsimonious 13-predictor fit.

Figure 3 displays the fits for the most important predictors, in order, over highest-density region grids. Such grids are explained in an appendix, and are crucial for display since the

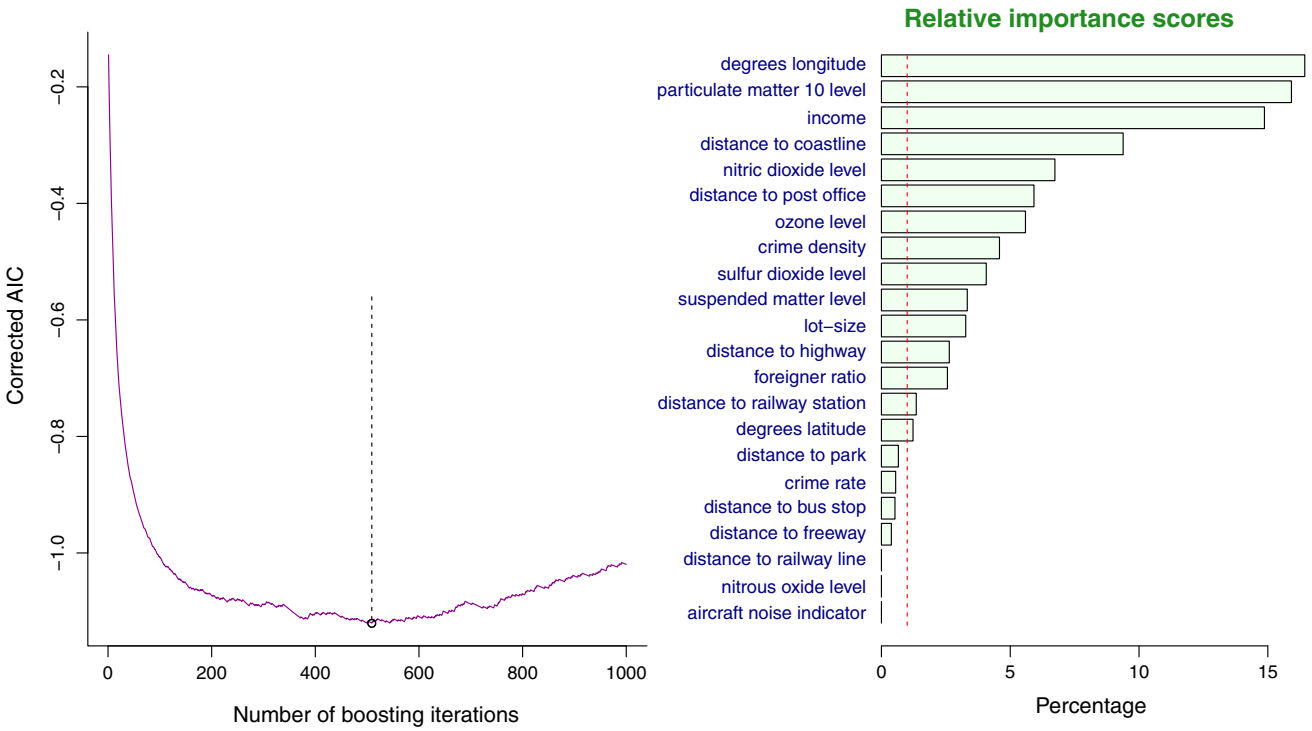

Fig. 2 *Left panel* plot of the corrected AIC path for wavelet-based gradient boosting applied to the Sydney residential property prices data. The minimum is achieved at $m^* = 509$ boosting iterations. *Right panel* ordered relative importance scores (percentage) for each of the candidate predictors in the Sydney residential property prices data set. The *dashed vertical line* corresponds to a 1 % threshold

fits can behave quite erratically in regions where the predictor data are sparse. In each panel, the estimated mean response is plotted for the labelled variable's grid and all other predictors are set to their average. The vertical axis is fixed across all panels, which permits visual comparison of each predictor's relative importance.

In the upper left-hand panel of Fig. 3 The predictor `degrees longitude` is seen to have an approximate positive linear impact on mean logarithmic house price between 150.8° and 151.17° degrees longitude. This indicates the relative expensiveness of Sydney's eastern suburbs compared to western Sydney. The peak corresponds approximately to the meridian of the exclusive North Shore region of Sydney. The panel underneath, for `distance to coastline`, shows the dramatic impact of proximity to Sydney's coastline. The sharp drop in the mean response over the first kilometer inland from the coast is well-captured by the wavelet basis functions. A similar comment applies to `distance to highway`, with a steep upward ramp in the first 100 m away from the closest highway. Subtler interpretations may be apparent from other panels in Fig. 3.

3.2 Spam filtering

Our second example involves spam filtering, based on a dataset used for illustration of several methods of classification in Hastie et al. (2009). In particular, we wanted to see if wavelet-based gradient boosting applied to this example can achieve cross-validatory misclassification rates comparable with existing methods.

The response variable is

$$y_i = \begin{cases} 0 & \text{if the ith e-mail message is normal,} \\ 1 & \text{if the ith e-mail message is spam} \end{cases}$$

for $1 \le i \le 4,601$. The full data were fed into Algorithm 1 with $M = 5,000$ and $\nu = 0.1$ and the loss function set to be binomial log-likelihood. The AIC-estimated optimal stopping position was $m^* = 2,875$

$$\text{edf}(\widehat{\boldsymbol{f}}) = 1 + \text{number of non-zero entries in } \widehat{\boldsymbol{u}}^{(m^*)} = 264$$

represents a $264/(1 + 57 \times 127) = 3.6\,\%$ compression of the full set of candidate predictors.

Figure 4 shows the AIC path and ordered relative importance scores. The most important variable is seen to be `percent words matching 'hp'`. Since these two letters are the initials of the worker's company (Hewlett Packard), one would expect the occurrence of the character string 'hp' to be strongly predictive of e-mail type.

Fits for the 15 most important variables are plotted in Fig. 5. E-mail messages with a high percentage of words matching the company's initial letters are seen to contribute to a normal e-mail classification. The reverse applies to e-mail messages with a high percentage of dollar signs—consistent with spam

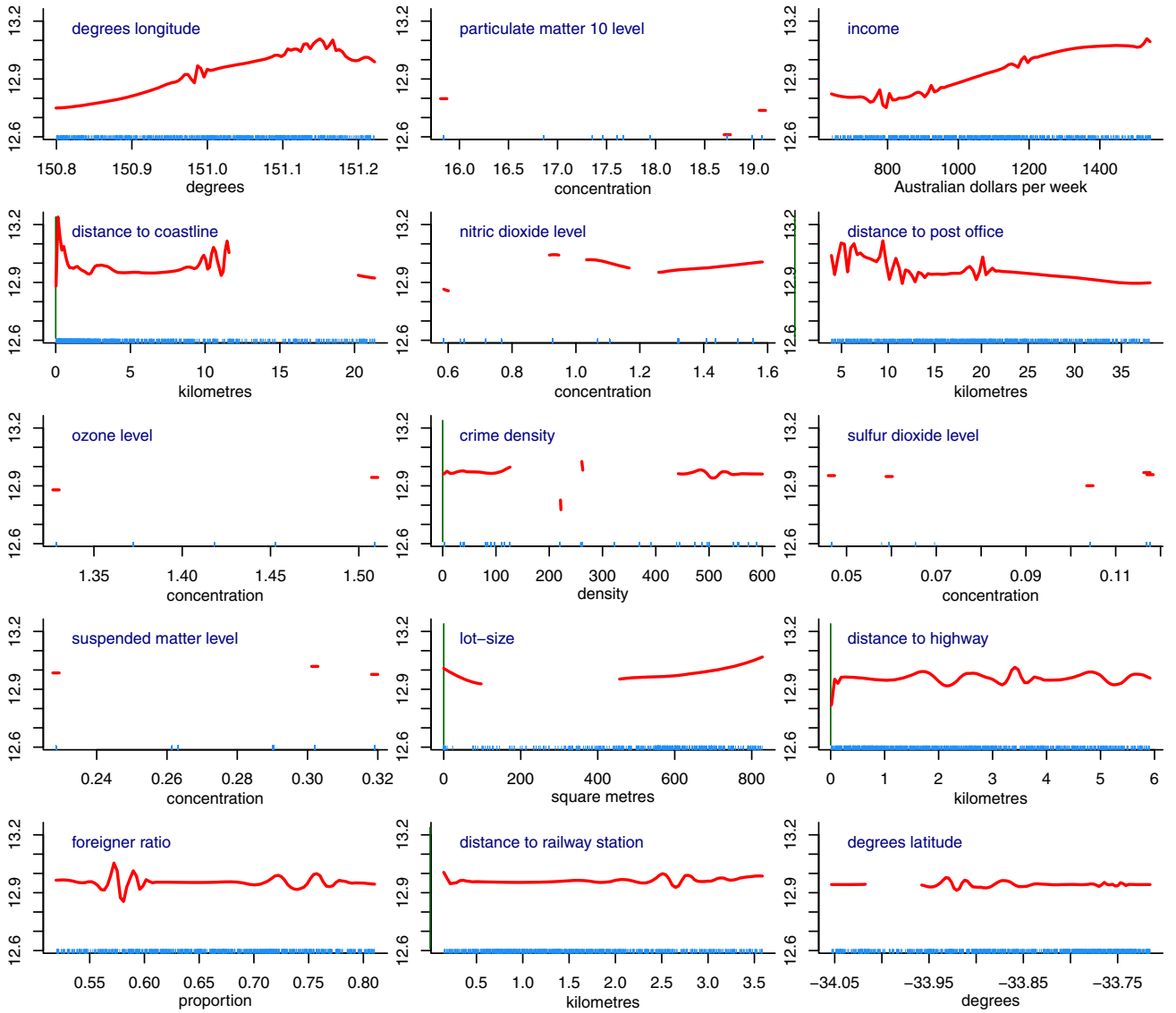

Fig. 3 Wavelet-based additive fits for the predictor variables having highest relative importance in the Sydney residential property prices data-set. Plotting is done over 80 % highest-density region grids

e-mail being more likely to contain monetary values. Similar interpretations apply to the other curves in Fig. 5.

We used 10-fold cross-validation to estimate the test error of wavelet-based gradient boosting for the spam filtering data-set. The average (standard deviation) confusion matrix is shown in Table 1. The mean (standard deviation) misclassification rate is 6.49% (1.18%). Allowing for the margin of error, this is comparable with previously published misclassification rates for these data (e.g. Table 9.1 of Hastie et al. 2009).

3.3 Simulation study

We conducted a simulation study to demonstrate the advantages afforded by wavelet-based gradient boosting when the predictor effects have jumpy or jagged features. The study involved both Gaussian and binary response data. Let x_{ij}, $1 \leq i \leq n$, $1 \leq j \leq 9$ be independent uniform random variables on the unit interval. For the Gaussian response case we generated the responses according to the additive model

$$y_i \sim N\left(\sum_{j=1}^{9} f_j(x_{ij}), 0.01\right), \quad 1 \leq i \leq n, \tag{7}$$

while the binary response model used

$$y_i \sim \text{Bernoulli}\left[\frac{\exp\left\{5\left(\sum_{j=1}^{9} f_j(x_{ij}) - 3\right)\right\}}{1 + \exp\left\{5\left(\sum_{j=1}^{9} f_j(x_{ij}) - 3\right)\right\}}\right], \quad 1 \leq i \leq n. \tag{8}$$

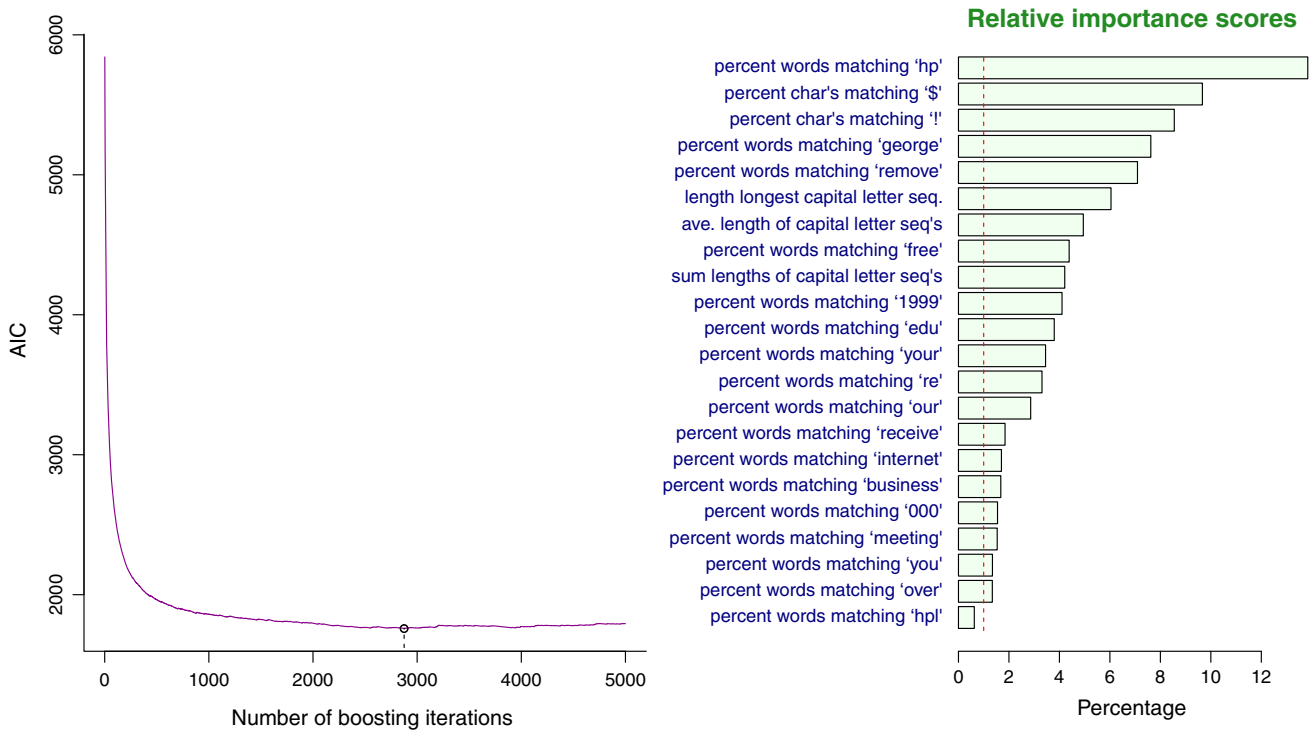

Fig. 4 *Left panel* plot of the AIC path for wavelet-based gradient boosting applied to the spam filtering data. The minimum is achieved at $m^* = 2,875$ boosting iterations. *Right panel* ordered relative importance scores (percentage) for each of the candidate predictors in the spam filtering data set. The *dashed vertical line* corresponds to a 1 % threshold

The f_j are given by

$$
\begin{aligned}
f_1(x) &= \frac{\Phi(6x-3)}{2\Phi(3)-1} + \Phi(3) - 1,\\
f_2(x) &= \tfrac{1}{2}\{1+\sin(3\pi x^2)\},\\
f_3(x) &= 1.4746\{\tfrac{7}{10}\phi(15x-3)\\
&\quad -\phi(7x-5.6)+0.39894\},\ f_4(x) = I(x>0.6),\\
f_5(x) &= 1.6603\Big[10x^2\, I(x\le 0.2)\\
&\quad -\{3(x-0.65)^2-0.15\}I(0.2<x\le 0.7)\\
&\quad +\{5(x-0.7)^2\}I(x>0.7)+0.1572\Big]\\
f_6(x) &= \sum_{k=1}^{11} h_i \max(0, 1-|x-t_i|/w_i)^4,\\
f_7(x) &= f_8(x) = f_9(x) = 0,
\end{aligned}
$$

where ϕ and Φ are the standard normal density and cumulative distribution functions, $I(\mathcal{P})$ is the indicator of the proposition $\mathcal{P}$ being true,

$$
\begin{aligned}
&(t_1,\ldots,t_{11}) = (0.1, 0.13, 0.15, 0.23, 0.25, 0.4, 0.44,\\
&\quad 0.65, 0.76, 0.78, 0.81),\\
&(h_1,\ldots,h_{11}) = (40, 50, 30, 40, 50, 42, 21, 43, 31,\\
&\quad 51, 42)/51 \quad \text{and}\\
&(w_1,\ldots,w_{11}) = (0.015, 0.015, 0.018, 0.03, 0.03, 0.09,\\
&\quad 0.03, 0.03, 0.015, 0.024, 0.015).
\end{aligned}
$$

Figure 6 displays the non-zero f_j. The first three f_j are smooth functions. However, f_4 and f_5 contains jumps and f_6 is very jagged since we wanted to see if wavelet-based gradient boosting offers improvements when such effects are present. Each of f_j, $1 \le j \le 6$, have been set up to range between 0 and 1 over the unit interval.

We generated 100 replications of (7) with $n = 1000$ and (8) with $n = 10,000$. The higher sample size was used for the binary response case to account for the inherent lower signal-to-noise ratio. Each set of data was then fed into the `gamboost()` functions with both B-spline and radial basis functions from the **mboost** package. For wavelet boosting we used the `glmboost()` and `ZDaub()` functions as described in Section 2.5. We ran the boosting methods for using 127 (B-spline, radial or wavelet) basis functions, for each of the 9 predictors and chose the model with the smallest corrected AIC in the Gaussian case and smallest AIC in the binary case.

For the Gaussian response case we recorded the mean square and maximum absolute errors between the true function and its estimate, given by

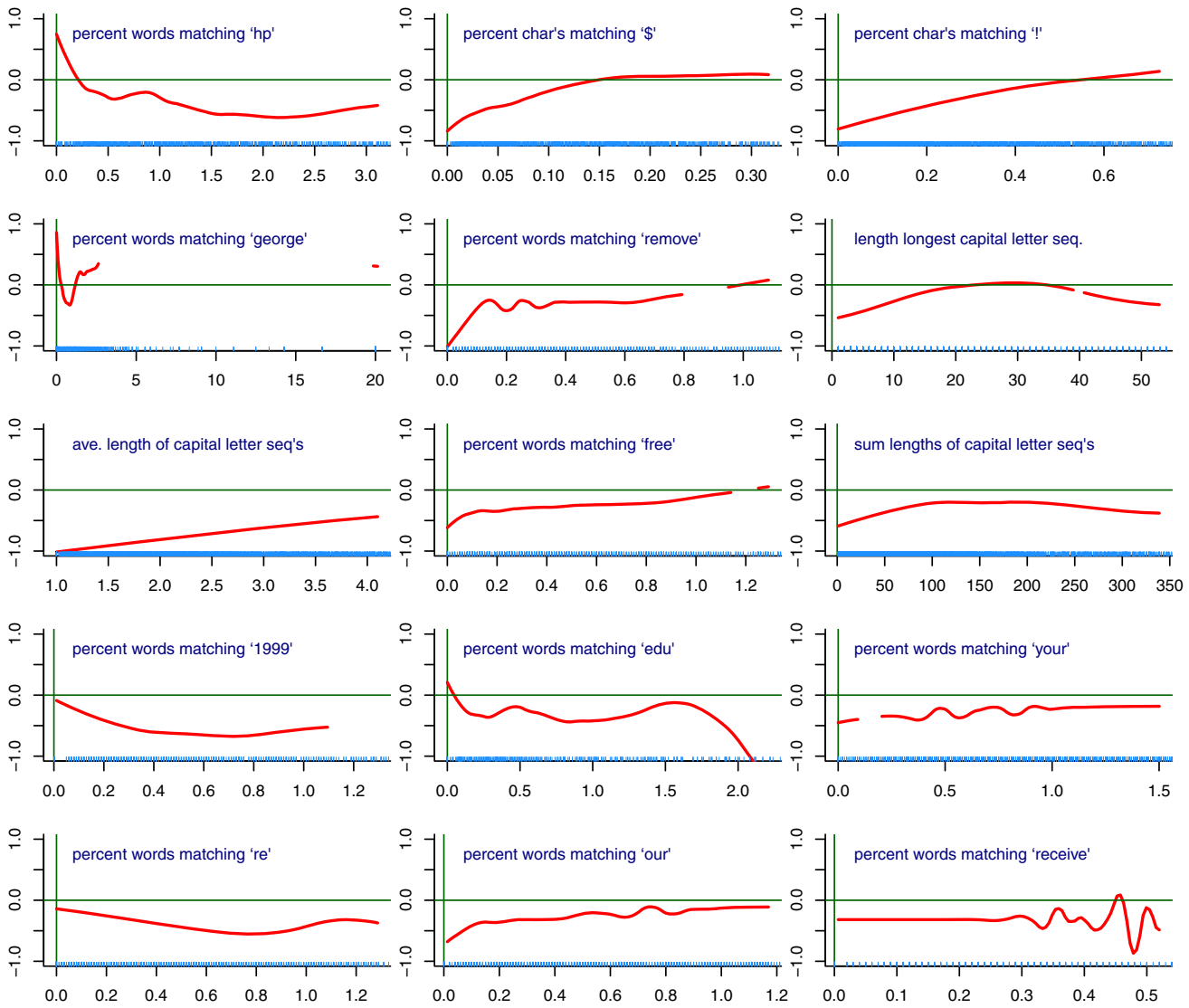

Fig. 5 Wavelet-based additive fits for the predictor variables having highest relative importance in the spam filtering data-set. Plotting is done over 80 % highest-density region grids

$$\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{9}\{\widehat{f}_j(x_{ij}) - f_j(x_{ij})\}\right]^2 \quad \text{and}$$

$$\frac{1}{n}\sum_{i=1}^{n}\max\left|\sum_{j=1}^{9}\{\widehat{f}_j(x_{ij}) - f_j(x_{ij})\}\right|$$

respectively. Note that, due to boundary effects of some wavelet fits, we only considered observations satisfying $0.005 < x_{ij} < 0.995$, $1 \le j \le 9$, when calculating the maximum absolute error measure. For the binary response case we simulated $n = 10000$ new out-of-sample test data cases and recorded the deviance and misclassification error on this test data.

Table 1 Average confusion matrix for the wavelet-based gradient boosting classification of the spam filtering data, based on 10-fold cross-validation

	Classified normal (%)	Classified spam (%)
Actually normal %	58.2	2.4
Actually spam %	4.1	35.3

The results are summarized in Fig. 7. We see that our wavelet boosting approach dominates the other methods for each of the performance measures. This confirms our intuition that wavelets are better suited to handling the jumps in $f_4(x)$, $f_5(x)$ and the jagged features of $f_6(x)$ and can lead to superior performance in applications where such features are present.

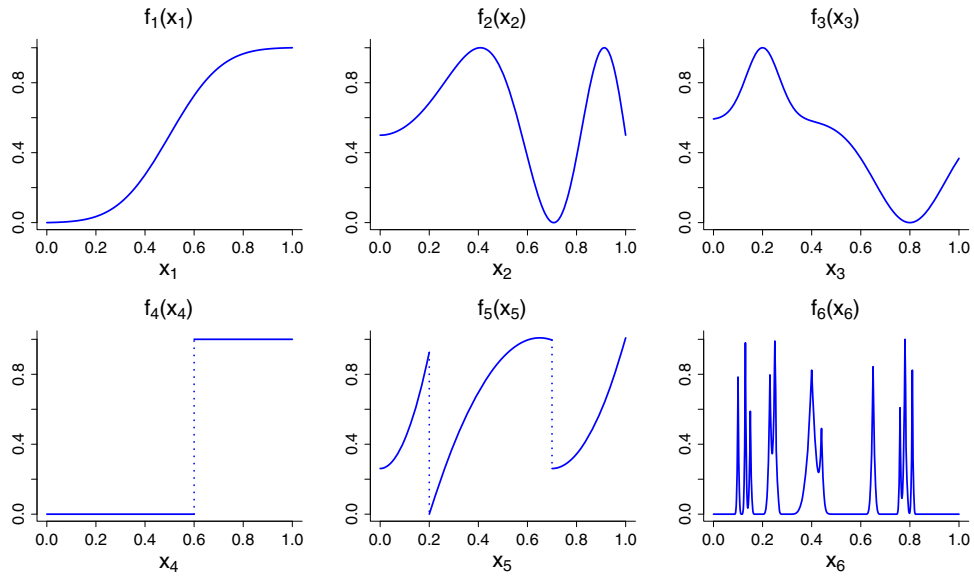

Fig. 6 Non-zero functions used for predictor effects in the simulation study

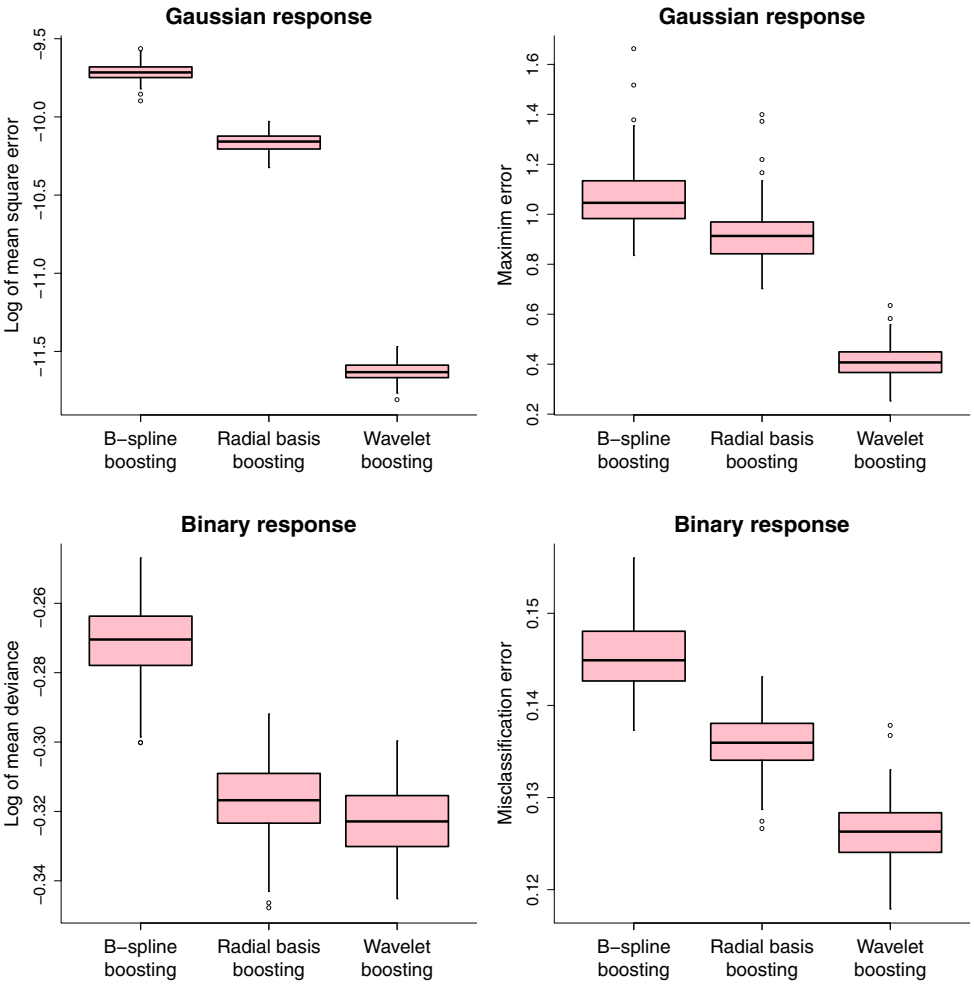

Fig. 7 Side-by-side boxplots for the performance measures for the simulation study described in the text

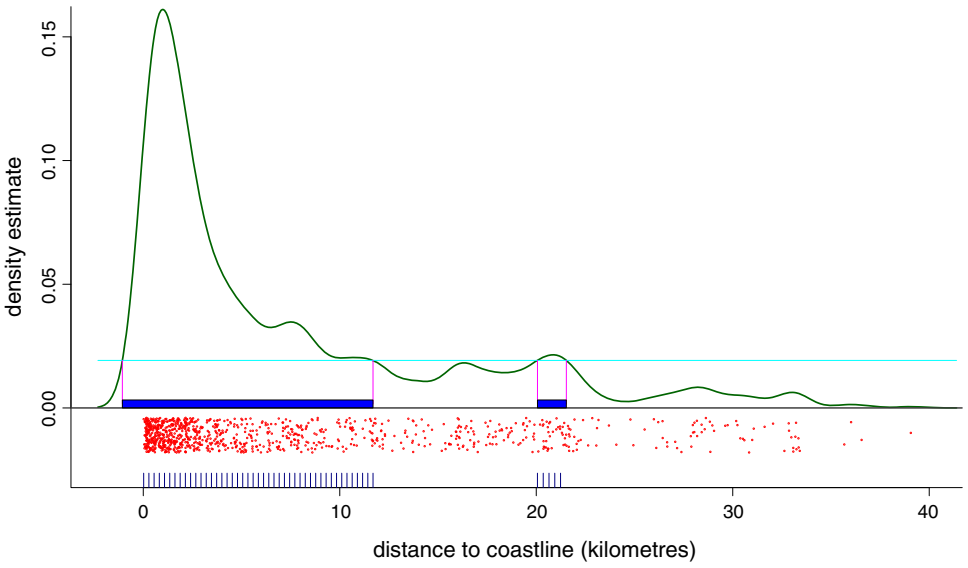

Fig. 8 A HDR grid for the predictor variable `distance to coastline` in the Sydney real estate data example. The data are shown as red dots with vertical jittering to enhance visualization. The green curve is a kernel density estimate and the blue bars are the estimated 80% highest density region. The rug at the base of the plot is the corresponding HDR grid of size 50

4 Conclusion

The use of wavelet basis functions within the gradient boosting paradigm is a natural and, in certain circumstances, beneficial approach. We have demonstrated that wavelet-based gradient boosting has definite practical advantages for data science applications involving jumpy and/or jagged predictor effects. Moreover, we have shown that it can be implemented using existing software within the R computing environment with the addition of a function that generates wavelet basis functions. In summary, wavelet-based gradient boosting is a useful and relatively painless addition to the data science arsenal.

Acknowledgments We are grateful to Andrew Chernih for his provision of the Sydney residential property price data and to Peter Green for his comments on aspects of this research. Partial support was provided by Australian Research Council Discovery Project DP0877055. Assistance from the University of Technology, Sydney's Distinguished Visitor programme is gratefully acknowledged.

Appendix: Highest-density region grids

We now provide details of the highest density region (HDR) grids used in Figures 3 and 5.

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a generic univariate sample and $\widehat{p}$ be a probability density estimate based on $\boldsymbol{x}$. Then a $100(1-\tau)\%$ highest-density region estimate is

$$\widehat{R}_\tau = \{x \in \mathbb{R} : \widehat{p}(x) \geq \widehat{p}_\tau\}$$

where $\widehat{p}_\tau$ is chosen so that the probability mass of $\widehat{p}$ over the set $\widehat{R}_\tau$ does not exceed $1-\tau$. See, for example, Samworth and Wand (2010) for a precise mathematical definition of $\widehat{p}_\tau$.

The most commonly used estimator $\widehat{p}$ for HDR estimation is the kernel density estimator

$$\widehat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where K is a kernel function and $h > 0$ is a bandwidth (see e.g. Wand and Jones 1995). Recently, Samworth and Wand (2010) devised an automatic rule for selection of h in the HDR estimation context. The R package **hdrcde** (Hyndman 2010) implements both HDR estimation and the Samworth-Wand bandwidth selector. Figure 8 shows 80% HDR estimate for the variable `distance to coastline` variable in the Sydney residential property prices data. The corresponding HDR grid of size 50 is shown at the base of the plot.

References

Binder, H., Tutz, G.: A comparison of methods for the fitting of generalized additive models. Stat. Comput. **18**, 87–99 (2008)

Bühlmann, P.: Boosting for high-dimensional linear models. Ann. Stat. **34**, 559–583 (2006)

Bühlmann, P., Yu, B.: Sparse boosting. J. Mach. Learn. Res. **7**, 1001–1024 (2006)

Bühlmann, P., Hothorn, T.: Boosting algorithms: regularization, prediction and model fitting (with discussion). Stat. Sci. **22**, 477–522 (2007)

Donoho, D.L.: De-noising by soft-thresholding. IEEE Trans. Inf. Theor. **41**, 613–627 (1995)

Donoho, D.L., Johnstone, I.M.: Ideal spatial adaptation by wavelet shrinkage. Biometrika **81**, 425–456 (1994)

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. **32**, 407–451 (2004)

Friedman, J.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**, 1189–1232 (2001)

Hansen, M.H., Yu, B.: Model selection and the principle of minimum description length. J. Am. Stati. Assoc. **96**, 746–774 (2001)

Hastie, T.: Comment on paper by Bühlmann & Hothorn. Stat. Sci. **22**, 513–515 (2007)
Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer, New York (2009)
Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M. & Hofner, B.: mboost 2.2. Model-based boosting. R package.(2011) http://cran.r-project.org
Hurvich, C.M., Simonoff, J.S., Tsai, C.: Smoothing parameter selection in nonparametric regression using an improved A kaike information criterion. J. R. Stat. Soc. B **60**, 271–293 (1998)
Hyndman, R.J.: hdrcde 2.15. Highest density regions and conditional density estimation. R package. (2010) http://cran.r-project.org
Leitenstorfer, F., Tutz, G.: Knot selection by boosting techniques. Comput. Stat. Data Anal. **51**, 4605–4621 (2007)
Nason, G.P.: Wavelet Methods in Statistics with R. Springer, New York (2008)
Nason, G.P.: wavethresh 4.5. Wavelets statistics and transforms. R package. (2010) http://cran.r-project.org
R Development Core Team R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, (2012) http://www.R-project.org
Ridgeway G.: gbm 1.6. Generalized boosted regression models. R package. (2012) http://cran.r-project.org
Samworth, R.J., Wand, M.P.: Asymptotics and optimal bandwidth selection for highest density region estimation. Ann. Stat. **38**, 1767–1792 (2010)
Vidakovic, B.: Statistical Modeling by Wavelets. Wiley, New York (1999)
Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman and Hall, London (1995)
Wand, M.P., Ormerod, J.T.: Penalized wavelets: embedding wavelets into semiparametric regression. Electron. J. Stat. **5**, 1654–1717 (2011)
Zou, H., Hastie, T., Tibshirani, R.: On the "degrees of freedom" of the lasso. Ann. Stat. **5**, 2173–2192 (2007)