Tutz, Binder:

# Generalized additive modelling with implicit variable selection by likelihood based boosting

Sonderforschungsbereich 386, Paper 401 (2004)

Online unter: http://epub.ub.uni-muenchen.de/

# Generalized additive modelling with implicit variable selection by likelihood based boosting

Gerhard Tutz[1] & Harald Binder[2]

[1] *Institut für Statistik, Ludwig-Maximilians-Universität München, Germany*

[2] *Klinik für Psychiatrie und Psychotherapie, Universität Regensburg, Germany*

November 2004

**Abstract**

The use of generalized additive models in statistical data analysis suffers from the restriction to few explanatory variables and the problems of selection of smoothing parameters. Generalized additive model boosting circumvents these problems by means of stagewise fitting of weak learners. A fitting procedure is derived which works for all simple exponential family distributions, including binomial, Poisson and normal response variables. The procedure combines the selection of variables and the determination of the appropriate amount of smoothing. As weak learners penalized regression splines and the newly introduced penalized stumps are considered. Estimates of standard deviations and stopping criteria which are notorious problems in iterative procedures are based on an approximate hat matrix. The method is shown to outperform common procedures for the fitting of generalized additive models. In particular in high dimensional settings it is the only method that works properly.

**Keywords:** Generalized additive models, boosting, selection of smoothing parameters, variable selection

# 1 Introduction

Generalized additive models assume that data $(y_i, x_i), i = 1, \ldots, n$, follow the model

$$\mu_i = h(\eta_i), \;\; \eta_i = f_{(1)}(x_{i1}) + \cdots + f_{(p)}(x_{ip})$$

where $\mu_i = E(y_i|x_i)$, $h$ is a specified response function and $f_{(j)}, j = 1, \ldots, p$, are unspecified functions of covariates. As in generalized linear models (McCullagh & Nelder, 1989) it is assumed that $y|x$ follows a simple exponential family, including among others normally distributed, binary, or Poisson distributed responses.

Various estimation procedures have been proposed for the estimation of functions $f_{(j)}$. Backfitting procedures based on local scoring (Hastie & Tibshirani, 1990) use iterative estimates whereas direct estimates are obtained by P-splines (Marx & Eilers, 1998) or the marginal integration method (Linton & Nielsen, 1995). For an overview of fitting methods see Schimek & Turlach (2000).

The proposed algorithms work well for given smoothing parameters and given set of variables. Serious problems occur if the number of variables is large and smoothing parameters as well as variables from the set of potentially influential variables have to be selected. Even if variables are given, determining the amount of smoothing for the components $f_{(j)}, j = 1, \ldots, p$, implies selection of smoothing parameters from a $p$ dimensional space. This limits additive modelling to the case of small $p$. Reduction to just one smoothing parameter is often inadequate since it implies that the variation in the surface is comparable with respect to all covariates. Problems increase if in addition variables have to be selected. Stepwise selection of variables along a prespecified path of models as offered by S (Chambers & Hastie, 1992) is bound to fail in the case of many variables.

A quite different approach to simultaneous selection of variables and degree of smoothing may be based on boosting concepts which have been derived and successfully used for prediction purposes in the machine learning community (e.g. Freund & Schapire, 1996). More recently, Bühlmann & Yu (2003) demonstrated how additive models may be fitted by iteratively refitting of residuals when using a weak learner. They show that $L_2$Boost which is constructed from $L_2$ loss works very well for high dimensional predictors when used in a componentwise fashion. In the present paper this approach is modified so that not only normally distributed dependent variables are possible but all distribution models familiar from generalized linear models. The weak learner that is used in each step of the algorithm can be based on penalized B-splines or on trees with two terminal nodes and both will be investigated. In Section 2 we introduce the new boosting procedure together with two types of weak learner. Approximate pointwise confidence bands and approximate degrees of freedom (useful for the selection of the number of boosting steps) are derived. In Section 3 the new algorithm is empirically evaluated and compared to other procedures. In Section 4 the method is

applied to the modelling of the number of readmissions in a psychiatric hospital. Although the number of predictors is only five some of the usual GAM procedures fail in this example.

# 2 Likelihood based boosting

## 2.1 Basic concepts

Boosting was originally developed in the machine learning community as a mean to improve classification procedures (e.g. Schapire, 1990). The basic concept to use a classifier iteratively with differing weights on the observations and combine the results in a committee voting has been shown to reduce the misclassification error drastically. More recently, it has been shown that boosting is a way of fitting an additive expansion in basis functions when the single basis functions represent the results of one iteration of the boosting procedure. The procedure is based on gradient descent by use of specific loss functions (Breiman, 1999; Friedman et al., 2000). From this view it is no longer restricted to classification problems. It may also be used in regression problems although the main body of the literature still focuses on classification problems. Friedman et al. (2000) replace the exponential loss function which underlies classical AdaBoost by the binomial log-likelihood yielding LogitBoost. Bühlmann & Yu (2003) investigate $L_2$ loss which yields the $L_2$Boost algorithm.

In the following a likelihood based procedure, called GAMBoost (Generalized Additive Model Boost), is proposed which aims at maximizing the likelihood in generalized additive models. In the case of the logit model and binomial likelihood it is quite close to LogitBoost but instead of Newton steps it uses Fisher scoring steps which are common in generalized linear models. The different algorithm comes from the focus on semiparametrically structured regression in the form of additive models for all kinds of link functions and distributions of the dependent variable. Moreover, variable selection by componentwise learners is incorporated.

As in generalized linear models let $y_i|x_i$ have a distribution from a simple exponential family $f(y_i|x_i) = \exp\{(y_i\theta_i - b(\theta_i))/\phi + c(y_i, \phi)\}$ where $\theta_i$ is the canonical parameter and $\phi$ is a dispersion parameter. Instead of assuming a linear predictor $\eta_i = x_i^T\beta$ in each boosting step the fitting of a simple learner (estimator)

$$\eta_i = \eta(x_i, \gamma)$$

is assumed, where $\gamma$ is a finite or infinite–dimensional parameter. If the learner is a decision tree, $\gamma$ describes the variable to be split, the split points and the values of the piecewise constant fitted function. For regression splines $\gamma$ describes the

weights of the spline functions and (potentially) the location of the knots. The likelihood to be maximized is given by

$$\begin{aligned} l(\gamma) &= \sum_{i=1}^{n} l(y_i, \eta_i) \\ &= \sum_{i=1}^{n} (y_i\theta_i - b(\theta_i))/\phi + \ c(y_i, \phi) \end{aligned}$$

where the canonical parameter $\theta_i$ is a function of $\eta_i = \eta(x_i, \gamma)$. The basic algorithm for likelihood based boosting is as follows.

---

**Likelihood based boosting**

*Step 1 (Initialization)*

For given data $(y_i, x_i)$, $i = 1, \ldots, n$, fit the model

$$\mu_{(0)}(x) = h(\eta(x, \gamma))$$

by maximizing the likelihood $l(\gamma)$ yielding $\hat{\eta}_{(0)}(x) = \eta(x, \hat{\gamma})$, $\hat{\mu}_{(0)}(x) = h(\hat{\eta}_{(0)}(x))$.

*Step 2*

For $l = 0, 1, \ldots$ fit the model

$$\mu_i = h(\hat{\eta}_{(l)}(x_i) + \eta(x_i, \gamma)) \tag{1}$$

to data $(y_i, x_i)$, $i = 1, \ldots, n$, where $\hat{\eta}_{(l)}(x_i)$ is treated as an offset and $\eta(x_i)$ is estimated by the learner $\eta(x_i, \gamma_l)$.

Set

$$\hat{\eta}_{(l+1)}(x_i) = \hat{\eta}_{(l)}(x_i) + \hat{\eta}(x_i, \hat{\gamma}_l). \tag{2}$$

---

In model (1) the predictor is considered as an unknown function of $x_i$ which is estimated by $\eta(x_i, \hat{\gamma})$. The estimate $\eta(x_i, \hat{\gamma})$ may represent a tree with fixed number of knots, multivariate splines or some other learner determined by $\gamma$. Likelihood based boosting will be called GAMBoost if the fitted predictor is additive in the variables $x_1, \ldots, x_p$. Although in principle any estimation method for additive models could be used, common procedures are restricted to few variables. Thus in the following we will focus on the *component-wise approach* which means that in each iteration only the contribution of a single variable is determined. A simple learner of this type which has often been used in boosting is a tree with only two

terminal nodes (stumps). With stumps the selection of the variable to be updated is done implicitly by tree methodology. When using regression splines, model fitting within the algorithm contains a selection step in which one variable $x_j$ is selected and only the corresponding function $f_{(j)}$ is updated. The componentwise update has the advantage that the selection of variables is performed by the fitting of simple models which contain only one variable. In particular we will use regression splines and trees with two terminal nodes. Regression splines provide simple learners which in contrast to trees yield smooth estimates. The estimation procedures for both approaches are derived below.

## 2.2 Generalized additive model boosting for specific smoothers

### 2.2.1 Penalized regression splines

*Componentwise smoothing* means that in each step only one variable is used as learner. For regression splines that means in each step the model

$$\mu_i = h(\hat{\eta}_i^{(l)} + f_{(j)}(x_{ij})) \tag{3}$$

is fit where $f_{(j)}$ is an unspecified function of the $jth$ variable. By using knots $\tau_1^{(j)} < \cdots < \tau_m^{(j)}$ from the range of the $jth$ variable and corresponding basis functions $B_1^{(j)}(x), \ldots B_m^{(j)}(x)$ regression splines fit

$$\mu_i = h(\hat{\eta}_i^{(l)} + z_{ij}^T \gamma) \tag{4}$$

where $z_{ij}^T = (B_1^{(j)}(x_{ij}), \ldots B_m^{(j)}(x_{ij}))$. Although the vector $\gamma$ is specific for variable $j$ the dependence on $j$ is suppressed in the notation. B-splines (used e.g by Eilers & Marx, 1996; Marx & Eilers, 1998) provide a numerically rather stable procedure. In the spirit of penalized B-splines (P-splines) it is useful to use many basis functions, say 20, and penalize them by use of a tuning parameter $\lambda$ (see e.g. Ruppert, 2002).

Fitting of model (4) is then based on maximizing the penalized likelihood

$$l_p(\gamma) = l(\gamma) - \frac{1}{2}\gamma^\tau \Lambda \gamma$$

where $\Lambda$ is a penalty matrix constructed such that $\gamma^T \Lambda \gamma$ penalizes first order differences $\sum_i (\gamma_{i+1} - \gamma_i)^2$ or higher order differences of parameters which correspond to basis functions of adjacent knots. In matrix notation one obtains the penalized score function

$$s_p(\gamma) = s(\gamma) - \Lambda\gamma$$

where

$$s(\gamma) = Z_j D(\gamma)\Sigma(\gamma)^{-1}(y-\mu) = Z_j^T W(\gamma)D(\gamma)^{-1}(y-\mu)$$

with $y^T = (y_1, \ldots, y_n)$, $\mu^T = (\mu_1, \ldots, \mu_n)$, $Z_j^T = (z_{1j}, \ldots, z_{nj})$,
$D(\gamma) = Diag(\partial h(\eta_1)/\partial\eta, \ldots, \partial h(\eta_n)/\partial\eta)$, $\eta_i = \hat{\eta}_i^{(l)} + z_{ij}^T\gamma$, $\Sigma(\gamma) = Diag(\sigma_1^2, \ldots, \sigma_n^2)$,
$\sigma_i^2 = var_\gamma(y_i)$, $W(\gamma) = D(\gamma)\Sigma(\gamma)^{-1}D(\gamma)$. The penalized Fisher matrix $F_p(\gamma) = E(-\partial^2(\gamma)/\partial\gamma\partial\gamma^T)$ has the form

$$F_p(\gamma) = F(\gamma) + \Lambda$$

where

$$F(\gamma) = E(-\partial^2 l(\gamma)/\partial\gamma\partial\gamma^T) = Z_j^T W(\hat{\gamma})Z_j.$$

One step in Fisher scoring is given by

$$\hat{\gamma}_{new} = F_p(\hat{\gamma})^{-1}s_p(\hat{\gamma}). \tag{5}$$

Although Fisher scoring steps could be iterated it is sufficient to use just one step since the fit is within an iterative procedure which is based on weak learners. Thus one sets $\hat{\eta}^{(l+1)}(x_i) = \hat{\eta}^{(l)}(x_i) + Z_j\hat{\gamma}_{new}$. It should be noted that the Fisher step in (5) is different from the usual Fisher step which has the form $\hat{\gamma}_{new} = \hat{\gamma} + F_p(\hat{\gamma})^{-1}s_p(\hat{\gamma})$. The reason is that in boosting one seeks an additive correction of the already fitted terms by fitting the residuals. As is seen below this effect results from scoring step (5).

In componentwise boosting in each boosting step it is necessary to decide which of the predictor variables is used in fitting model (3). A straightforward criterion is to link the choice of the variable to the improvement of fit by one Fisher scoring step. For likelihood based models an appropriate measure is the deviance which depends on the form of the considered distribution. For binary responses $y_i \in \{0, 1\}$ the mean $\hat{\mu}_i$ corresponds to $\hat{\pi}_i$ where $\pi_i = P(y_i = 1|x_i)$ and the deviance may be written as

$$Dev = -2\sum_{i=1}^{n}\log(1 - |y_i - \hat{\pi}_i|).$$

Then the variable is selected that yields the lowest deviance.

Before summarizing the algorithm it is useful to recall the additive structure that is fitted. In vectorized form the fitting of a generalized model means determination of the additive linear predictor

$$\hat{\eta} = f_1 + \cdots + f_p$$

where $\hat{\eta}^T = (\hat{\eta}_1, \ldots, \hat{\eta}_n)$ is the fitted predictor and $f_j^T = (f_{1j}, \ldots, f_{nj})$, $f_{ij} = f_{(j)}(x_{ij})$ is the fitted vector of the $j$th variable at observation $x_{1j}, \ldots, x_{nj}$. With $Dev(\hat{\eta})$ denoting the deviance when $\hat{\eta}$ is the fitted predictor the algorithm with penalized regression splines (in step 2) has the following form.

*Step 2* (Model fit)

For $l = 0, 1, \ldots$

1. *Estimation step*:
   For $s = 1, \ldots, p$ compute
   $$f_{s,new} = Z_s(Z_s^T \hat{W} Z_s + \lambda\Lambda)^{-1}(Z_s^T \hat{W} \hat{D}^{-1}(y - \hat{\mu}) - \lambda\Lambda\gamma) \tag{6}$$
   where $\hat{W}$, $\hat{D}$, $\hat{\mu}$ are evaluated at $\hat{\eta}^T = (\hat{\eta}_1^{(l)}, \ldots, \hat{\eta}_n^{(l)})$.
2. *Selection step*:
   Set $f_s = f_s^{(l)} + f_{s,new}$ yielding $\hat{\eta}_{s,new}$
   Compute
   $$j = \arg\max_s \{Dev(\hat{\eta}_{s,new}) - Dev(\hat{\eta}_s^{(l)}))$$
3. *Update*:
   Set $f_j^{(l+1)} = f_j^{(l)} + f_{j,new}$

From (6) it is seen that the update has the form of weighted least squares fitting for the scaled residuals $D^{-1}(y - \hat{\mu})$ (without smoothing term). For the canonical link one has $\partial h(\eta_i)/\partial\eta = var(y_i)/\phi$ and the residuals are given by $(y_i - \hat{\mu})\phi/var(y_i)$.

### 2.2.2 Likelihood boosting by stumps

Classification and regression trees (CARTs) as introduced by Breiman et al. (1984) partition the space of explanatory variables recursively into rectangles within which constants are fitted. The simplest binary tree consists of stumps, i.e. only two terminal nodes are built. There is a wide body of literature on CARTs, in particular stumps have been successfully used for classification purposes (e.g. Friedman et al., 2000). Within a likelihood based concept they may be considered as fitting of a simple model. Given predictor $\eta_i^{(l)}$ a split in variable $x_j$ corresponds to the fitting of the model $M_\delta$ which has the predictor

$$\eta_i = \eta_i^{(l)} + c_1\, I(x_{ij} \leq \delta) + c_2\, I(x_{ij} > \delta) \tag{7}$$

where $I(.)$ denotes the indicator function with $I(a) = 1$ if $a$ is true and $I(a) = 0$ if $a$ is not true. The parameter vector $\gamma$ now contains $c_1, c_2, \delta$.

With $l_{M_\delta}$ denoting the log–likelihood of model $M_\delta$ the deviance $D_{M_\delta} = -2\phi\{l_{M_\delta} - l_{sat}\}$ has the form

$$D_{M_\delta} = (y_i\theta_i - b(\theta_i))/\phi + c(y_i, \phi) \tag{8}$$

where $\eta_i$ is defined by (7).

For the normal distribution model with identity link one obtains

$$D_{M_\delta} = \sum_{x_{ij} \le \delta} (y_i - \mu_i^{(l)} - c_1)^2 + \sum_{x_{ij} > \delta} (y_i - \mu_i^{(l)} - c_2)^2$$

and simple derivation shows that the deviance is minimized by

$$\hat{c}_1 = \sum_{x_{ij} \le \delta} (y_i - \mu_i^{(l)})/n_1, \;\; \hat{c}_2 = \sum_{x_{ij} > \delta} (y_i - \mu_i^{(l)})/n_2$$

where $n_1 = \#\{x_{ij} : x_{ij} \le \delta\}$, $n_2 = n - n_1$. Thus only $\delta$ has to be computed to find the threshold that minimizes $D_{M_\delta}$.

For binary $y_i \in \{0, 1\}$ one obtains for the logit link

$$\begin{aligned} D_{M_\delta} \;\; = \;\; & 2\,\{\sum_{x_{ij} \le \delta} (y_i(\eta_i^{(l)} + c_1) - \log(1 + \exp(\eta_i^{(l)} + c_1))) \\ & + \sum_{x_{ij} > \delta} (y_i(\eta_i^{(l)} + c_2) - \log(1 + \exp(\eta_i^{(l)} + c_2)))\}. \end{aligned}$$

Regrettably maximization of $D_{M_\delta}$ in this case is not as straightforward as the splitting of nodes in common methodology where $\eta_i^{(l)}$ is zero. If the offset $\eta_i^{(l)}$ does not depend on $i$, that is if $\eta_i^{(l)} = c$ holds for constant $c$ (including $c = 0$), an explicit solution for $c_1, c_2$ (depending on $\delta$) is easily derived. However, in likelihood boosting alternative estimation procedures have to be used.

For fixed $\delta, c_1, c_2$ may be estimated by Fisher scoring with predictor

$$\eta_i = \eta_i^{(l)} + (z_{i1}, z_{i2})^T \alpha, \tag{9}$$

where $z_{i1} = I(x_{ij} \le \delta)$, $z_{i2} = I(x_{ij} > \delta)$, $\alpha^T = (c_1, c_2)$. Then one step in Fisher scoring is given by

$$\hat{\alpha}_{new} = F(\hat{\alpha})^{-1} s(\hat{\alpha}). \tag{10}$$

For the logit model which uses canonical link this reduces with $z_i^T = (z_{i1}, z_{i2})$ and $F(\gamma) = Z_{j,\delta}^T \Sigma^{-1} Z_{j,\delta}$, $Z_{j\delta}^T = (z_1, \ldots, z_n)$ to

$$\hat{\alpha}_{new} = (Z_{j,\delta}^T \Sigma Z_{j,\delta})^{-1} Z_{j,\delta}^T (y - \pi^{(l)} - \mu(\hat{\alpha})).$$

The total update $\hat{\gamma}_{new}$ uses the $\delta$ value that minimizes the deviance, say $\delta_{new}$, to obtain $\hat{\gamma}_{new}^T = (\hat{\alpha}_{new}^T, \delta_{new})$.

***Penalized stumps***

Although stumps may be considered a weak learner it turns out that in applications they might not be weak enough. This phenomenon has already been observed by Bühlmann & Yu (2003). In boosting procedures with Newton steps (e.g. LogitBoost) this is often prevented by decreasing the size of the steps. In the context of GAMBoost where Fisher scoring is used we propose penalized stumps as an alternative. Penalized stumps are built from fitting stumps with an penalization on the log-likelihood. With underlying predictor (9) and parameter vector $\alpha^T = (c_1, c_2)$ the corresponding log-likelihood is penalized by adding $\frac{\lambda}{2}\alpha^T \Lambda \alpha$ where $\Lambda$ is given by

$$\Lambda = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

With the penalization given as

$$\lambda \alpha^T \Lambda \alpha = \lambda (c_2 - c_1)^2$$

the difference of parameters is shrunken toward zero. For $\lambda = 0$ one obtains common stumps, for $\lambda \to \infty$ one obtains for penalized estimates $\hat{c}_1 - \hat{c}_2 \to 0$.

The GAMBoost algorithm with penalized stumps is given in the following form:

---

**GAMBoost with Penalized Stumps**

*Step 2* (Model fit)

For $l = 0, 1, \ldots$

1. *Estimation step*:
   For $s = 1, \ldots, p$ and $\delta = x_{1s}, \ldots, x_{ns}$, compute

   $$\hat{\alpha}_{s,\delta,new} = (Z_{s,\delta}^T \hat{W} Z_{s,\delta} + \lambda \Lambda)^{-1} (Z_{s,\delta}^T \hat{W} \hat{D}^{-1} (y - \hat{\mu}))$$

   where $\hat{W}$, $\hat{D}$, $\hat{\mu}$ are evaluated at $\hat{\eta}^T = (\hat{\eta}_1^{(l)}, \ldots, \hat{\eta}_n^{(l)})$.

2. *Selection step*:
   Set $f_{s,\delta} = f_s^{(l)} + Z_{s,\delta} \hat{\alpha}_{s,\delta,new}$ yielding $\hat{\eta}_{s,\delta,new}$
   Compute

   $$(j, \delta_m) = \arg\max_{s,\delta} \{ Dev(\hat{\eta}_{s,\delta,new} - Dev(\hat{\eta}_s^{(l)}))$$

3. *Update*:
   Set $f_j^{(l+1)} = f_j^{(l)} + Z_{j,\delta_m}\hat{\alpha}_{j,\delta_m,new}$

---

By using the specific structure of the design matrix $Z_{s,\delta}$ one may derive a more explicit form of the estimation step yielding

$$\hat{\alpha}_{s,\delta,new} = \frac{1}{(\bar{F}_{x_{ij}\leq\delta}+\lambda)(\bar{F}_{x_{ij}>\delta}+\lambda)-\lambda^2} \cdot \begin{pmatrix} (\bar{F}_{x_{ij}>\delta}+\lambda)\bar{s}_{x_{ij}\leq\delta} + \lambda\bar{s}_{x_{ij}>\delta} \\ (\bar{F}_{x_{ij}\leq\delta}+\lambda)\bar{s}_{x_{ij}>\delta} + \lambda\bar{s}_{x_{ij}\leq\delta} \end{pmatrix}$$

where

$$\bar{F}_{x_{ij}\leq\delta} = \sum_{x_{ij}\leq\delta} \frac{1}{\hat{\sigma}_i^2}\left(\frac{\partial h(\hat{\eta}_i)}{\partial\eta}\right)^2, \quad \bar{F}_{x_{ij}>\delta} = \sum_{x_{ij}>\delta} \frac{1}{\hat{\sigma}_i^2}\left(\frac{\partial h(\hat{\eta}_i)}{\partial\eta}\right)^2$$

$$\bar{s}_{x_{ij}\leq\delta} = \sum_{x_{ij}\leq\delta} \frac{1}{\hat{\sigma}_i^2}\frac{\partial h(\hat{\eta}_i)}{\partial\eta}(y_i-\hat{\mu}_i), \quad \bar{s}_{x_{ij}>\delta} = \sum_{x_{ij}>\delta} \frac{1}{\hat{\sigma}_i^2}\frac{\partial h(\hat{\eta}_i)}{\partial\eta}(y_i-\hat{\mu}_i).$$

It is seen that for the normal model with identity link and $\lambda = 0$ estimation reduces to calculation of the mean of the residuals $(y_i - \hat{\mu}_i)$ for observations with $x_{ij} \leq \delta$ and observations with $x_{ij} > \delta$.

### 2.2.3 B-splines versus stumps: an empirical comparison

In the following we will use examples with simulated data where some of the $p$ predictors $x_{i1}, \ldots, x_{ip}$ carry information on the response variable $y_i$ (either from a Binomial or from a Poisson distribution) and some represent just noise. The following example uses three informative predictors drawn from a uniform distribution (values between $-1$ and 1) with the predictor given by

$$\eta_i = c_n(-0.7 + x_{i1} + 2x_{i3}^2 + \sin 5x_{i5}) \tag{11}$$

and the distribution of the response $y_i$ being

$$y_i \sim \text{Binomial}\left(1, \frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right)$$

for the binary case and

$$y_i \sim \text{Poisson}(\exp(\eta_i))$$

for the Poisson case. The constant $c_n$ determines the extent to which the informative predictors contribute to the response. With smaller values it will be harder to identify the shape of the predictor contribution as there is a smaller signal-to-noise ratio. When not noted otherwise we will use a training sample size of $n = 100$.

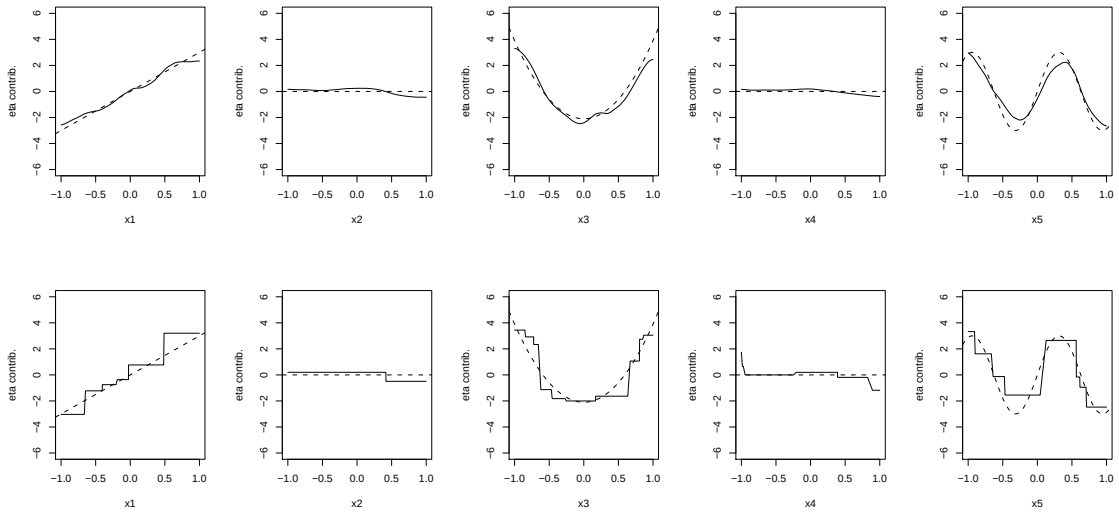


Figure 1: Boosting fit to simulated data with B-spline (upper panels) and tree stump (lower panels) learners. The broken lines indicate the true predictor influence functions and the solid lines indicate the model fit.

Figure 1 shows the fit to binary response data generated from model (11) with $c_n = 3$. The penalty parameters used were $\lambda_{splines} = 30$ and $\lambda_{stumps} = 2$ where for B-splines equidistant knots have been used throughout the paper. The upper panels show the results for penalised B-spline learners and the lower panels the results for penalized tree stumps. Although both fits (solid lines) track the true predictor functions (broken lines) closely (with the B-spline fit being slightly dampened compared to the true function) the fits based on B-splines are visually more satisfying.

The left panel of Figure 2 shows how for this example the corresponding mean squared error (MSE) of $\eta$ for GAMBoost with B-splines (solid line) and penalized stumps (broken line) respectively depends on the number of boosting iterations. The dash-dotted line shows the MSE for unpenalized stump learners. It is seen that overfitting occurs earlier when no penalty is used and the minimal value obtained during iterations is rather large. The situation becomes even more severe with a smaller signal-to-noise ratio (not shown here). This example demonstrates that penalization can effectively prevent early overfitting when tree stump learners are not weak enough. Penalizing stumps also result in a smaller minimum MSE compared to unpenalized stumps. The MSE for both types of tree stumps is consistently larger than for B-spline learners

The better performance of B-spline learners might be due to the special nature of the simulated data (smooth true functions and no categorical predictors). There-

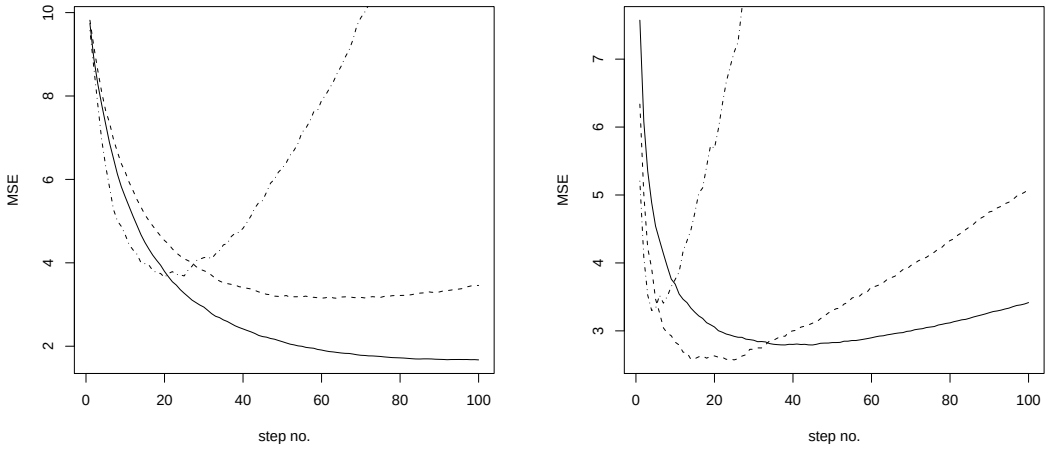


Figure 2: Mean curves of mean squared error (MSE) for 20 repetitions of data generation and model fit plotted against the number of boosting steps for data with smooth influence of predictors (left panel) and predictor influence via step functions (right panel) (GAMBoost with penalised B-splines: solid lines; with penalized stumps: broken lines; with unpenalized stumps: dash-dotted lines).

fore we generated binary response data with the same setup as the previous example but this time using a step function

$$f_{step}(x) = \begin{cases} -1 & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$$

and the predictor given by

$$\eta_i = c_n(0.5 f_{step}(x_{i1}) + 0.25 f_{step}(x_{i3}) + f_{step}(x_{i5})).$$

The right panel of Figure 2 shows the MSE for GAMBoost with B-splines and penalized and unpenalized stumps. The penalty parameters used for B-splines and stumps are the same as in the last example. Again unpenalized stumps show early overfitting that can be prevented to some extent by penalization. The minimum MSE achieved by penalized tree stumps learners in this case is smaller than that resulting when B-spline learners are used. This does not come as a surprise as this second example is designed to favor tree learners compared to learners that use smooth functions like penalized B-splines.

In summary, at least if the underlying functions are smooth, B-splines seem to yield smaller values of MSE. In addition the fits are visually more pleasing.

## 2.3 Stopping criteria and complexity

Stopping of boosting iterations is often based on cross-validation (e.g. Dettling & Bühlmann, 2003). In particular for larger data sets cross-validation becomes very time consuming. An attractive alternative is to link the amount of flexibility to be used in boosting to some measure that specifies the model complexity at a specific boosting step. With such a measure the optimal number of boosting steps can be determined as a trade-off between complexity and goodness-of-fit. A well known measure for the trade-off between model complexity and goodness-of-fit is the information criterion given by Akaike (AIC). For this measure the degrees of freedom used by a model fit have to be known. Although the degrees of freedom are known for each iteration, the degrees of the iterative procedure do not have a simple additive form. Therefore the effective degrees of freedom (as introduced by Hastie & Tibshirani, 1990) which are given given by the trace of the hat matrix will be used. In the following an approximate hat matrix for GAMBoost with penalized B-splines will be developed, but the formulation can also be adapted to tree stumps.

When considering penalized regression splines the vectorized predictor $\eta^T = (\eta_1, \ldots, \eta_n)$ is given by $\eta = Z\beta$ where the total design matrix $Z = (Z_1, \ldots, Z_p)$ decomposes into the design matrices for single variables

$$Z_j^T = (z_{1j}, \ldots, z_{nj}) \quad \text{with} \quad z_{ij}^T = (B_1^{(j)}(x_{ij}), \ldots, B_m^{(j)}(x_{ij})), j = 1, \ldots, p.$$

Similarly the vector $\beta$ decomposes into $\beta^T = (\beta_1^T, \ldots, \beta_p^T)$ where $\beta_i$ denotes the parameters for the basis function values collected in $Z_j$. The additive structure is captured in

$$Z\beta = Z_1\beta_1 + \cdots + Z_p\beta_p = f_1 + \cdots + f_p.$$

In the following it is essential to distinguish between the estimates which evolve over iterations and the change over iterations. The $\gamma$-parameters represent the change over iterations whereas the evolving estimates are denoted by $\beta$.

Let $j$ denote the variable that is selected in the $(l+1)$th iteration. The linear predictor of the fitted model has the form

$$\hat{\eta}_{(l+1)} = \hat{\eta}_{(l)} + Z_j\gamma_j.$$

With estimate $\hat{\gamma}_j$ the total vector after the update is denoted by

$$\hat{\beta}_{(l+1)}^T = \hat{\beta}_{(l)}^T + (0^T, \ldots, \hat{\gamma}_j^T, \ldots, 0^T)$$

where $0^T = (0, \ldots, 0)$ is a vector of length $m$, containing only zeros. The predictor in the $(l+1)$th iteration has the form

$$\hat{\eta}_{(l+1)} = Z\hat{\beta}_{(l)} + Z_j\hat{\gamma}_j = Z\hat{\beta}_{(l+1)}.$$

With starting value $\hat{\gamma}_{j0}$ and $W_l = W(\hat{\gamma}_{j0})$, $\Sigma_l = \Sigma(\hat{\gamma}_{j0})$, $D_l = D(\hat{\gamma}_{j0})$ denoting evaluations at value $\hat{\eta} = \hat{\eta}_{(l)} + Z_j\hat{\gamma}_{j0}$ one step Fisher scoring (5) is given by

$$\begin{aligned}\hat{\gamma}_j &= F_p(\hat{\gamma}_{j0})^{-1}s_p(\hat{\gamma}_{j0}) \\ &= (Z_j^T W_l Z_j + \Lambda)^{-1}(Z_j W_l D_l^{-1}(y - \hat{\mu}_{(l)})),\end{aligned}$$

where $\hat{\mu}_{(l)} = h(\hat{\eta}_{(l)} + Z_j\gamma_{j0})$. One obtains

$$\begin{aligned}\hat{\eta}_{(l+1)} &= Z\hat{\beta}_{(l)} + Z_j\hat{\gamma}_j \\ \hat{\eta}_{(l+1)} - \hat{\eta}_{(l)} &= Z_j\hat{\gamma}_j \\ &= Z_j(Z_j^T W_l Z_j + \Lambda)^{-1} Z_j^T W_l D_l^{-1}(y - \hat{\mu}_{(l)}).\end{aligned}$$

Taylor approximation of first order $h(\hat{\eta}) = h(\eta) + \frac{\partial h(\eta)}{\partial \eta^T}(\hat{\eta} - \eta)$ yields

$$\begin{aligned}\hat{\mu}_{(l+1)} &\approx \hat{\mu}_{(l)} + D_l(\hat{\eta}_{(l+1)} - \hat{\eta}_{(l)}) \\ \hat{\eta}_{(l+1)} - \hat{\eta}_{(l)} &\approx D_l^{-1}(\hat{\mu}_{(l+1)} - \hat{\mu}_{(l)})\end{aligned}$$

and therefore

$$W_l^{1/2} D_l^{-1}(\hat{\mu}_{(l+1)} - \hat{\mu}_{(l)}) \approx W_l^{1/2} Z_j (Z_j^T W_l Z_j + \Lambda)^{-1} Z_j^T W_l D_l^{-1}(y - \hat{\mu}_{(l)}).$$

Since $W(\gamma)^{1/2} D(\gamma)^{-1} = \Sigma(\gamma)^{-1/2}$ one has

$$\Sigma_l^{-1/2}(\hat{\mu}_{(l+1)} - \hat{\mu}_{(l)}) \approx \bar{H}_{l+1}\Sigma_l^{-1/2}(y - \hat{\mu}_{(l)})$$

where $\bar{H}_{l+1} = W_l^{1/2} Z_j (Z_j^T W_l Z_j + \Lambda)^{-1} Z_j^T W_l^{1/2}$. With

$$\hat{\mu}_{(l+1)} - \hat{\mu}_{(l)} \approx M_{l+1}(y - \hat{\mu}_{(l)})$$

where $M_{l+1} = \Sigma_l^{1/2}\bar{H}_l\Sigma_l^{-1/2}$ one obtains

$$\begin{aligned}\hat{\mu}_{(l+1)} &\approx \hat{\mu}_{(l)} + M_{l+1}(y - \hat{\mu}_{(l)}) \\ &= \hat{\mu}_{(l)} + M_{l+1}(y - \hat{\mu}_{(l-1)} - (\hat{\mu}_{(l)} - \hat{\mu}_{(l-1)})) \\ &\approx \hat{\mu}_{(l)} + M_{l+1}(y - \hat{\mu}_{(l-1)} - M_l(y - \hat{\mu}_{(l-1)})) \\ &= \hat{\mu}_{(l)} + M_{l+1}(I - M_l)(y - \hat{\mu}_{(l-1)}).\end{aligned}$$

With the starting value $\hat{\mu}_{(0)} = M_0 y$ one obtains

$$\begin{aligned}\hat{\mu}_{(1)} &\approx M_0 y + M_1(y - \hat{\mu}_{(0)}) \\ &= M_0 y + M_1(I - M_0)y\end{aligned}$$

and more general

$$\hat{\mu}_{(m)} \approx \sum_{j=0}^{m} M_j \prod_{i=0}^{j-1}(I - M_i)\ y.$$

The corresponding hat matrix is given by

$$H_m = \sum_{j=0}^{m} M_j \prod_{i=0}^{j-1} (I - M_i). \tag{12}$$

The initial estimate corresponds to the fitting of the intercept which yields $\hat{\mu}_0^T = (\bar{y}, \ldots, \bar{y})$. Thus $M_0$ is given by $M_0 = \frac{1}{n} 1_n 1_n^T$ where $1_n^T = (1, \ldots, 1)$.

In the special case where $M_l$ does not depend on the iteration, i.e. $M_l = S$ with linear smoother $S$, one obtains the form

$$H_m = \sum_{j=0}^{m} S(I - S)^{j-1} = I - (I - S)^{m+1}$$

which is equivalent to the form given in Bühlmann & Yu (2003). However, the use of the same smoother in each iteration in particular excludes componentwise boosting where only one variable is used in each iteration.

The hat matrix is used to obtain the complexity of the fit, which for the $m$th estimate is given by $df_m = tr(H_m)$. The corresponding AIC criterion is given by

$$\text{AIC} = D_m + 2 \cdot df_m \tag{13}$$

where $D_m$ is the deviance of the model $\hat{\mu}_{(m)} = h(\hat{\eta}_{(l)})$.

To investigate if the hat matrix approximation (12) leads to a useful criterion for selection of the number of boosting steps we compare the number of boosting steps chosen by approximate AIC (calculated from $tr(H_M)$ and (13)) to the number of steps chosen by cross-validation. Usually cross-validation is based on squared errors. But since the responses $y_i$ are not necessarily normally distributed a more general concept is needed. By use of Kullback-Leibler discrepancy one obtains the likelihood based cross-validation criterion which uses the log-likelihood $l(y_i, \eta_i)$ for observation $y_i$ in the form $l(y_i, \hat{\eta}_i^{-S})$ which is the log-likelihood for observation $y_i$ with the predictor $\hat{\eta}_i^{-S}$ obtained by leaving out the subsample S. Twenty data sets have been generated from model (11) with $c_n = 2.5$ for binary response and $c_n = 1$ for Poisson response, where the values of $c_n$ and the penalty $\lambda$ have been chosen such that the MSE curves have a similar shape for both examples. Figure 3 shows the number of boosting steps selected by the approximate AIC (circles) and 5-fold cross-validation (crosses) plotted on the mean MSE curve. For the binary response example (left panel) as well as for the Poisson response example (right panel) the number of steps chosen by approximate AIC is closer to the minimum of the MSE than for the cross-validation method. The tendency of cross-validation to stop too early often results in very high values of MSE. In particular for Poisson distributed responses the number of steps chosen by

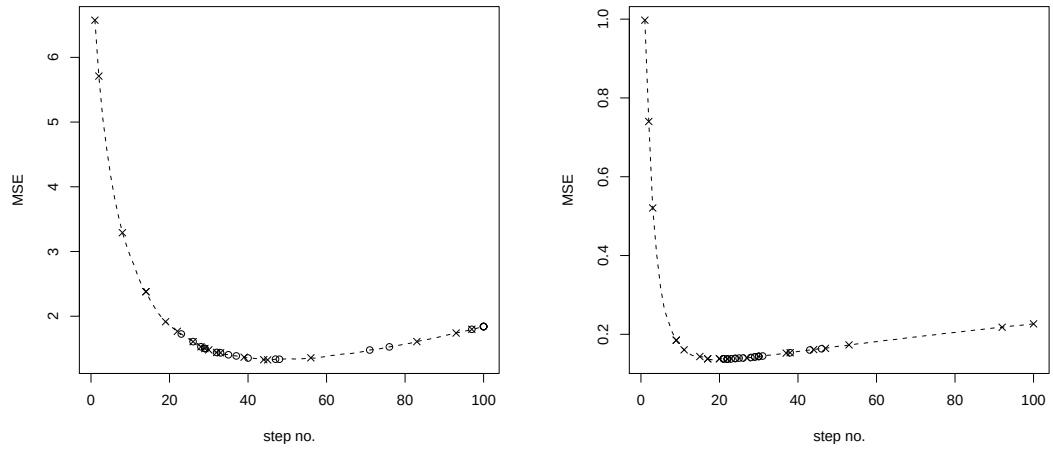


Figure 3: Number of boosting steps selected by AIC (circles) and cross-validation (crosses) for binary response data (left panel) and Poisson response data (right panel) plotted on the mean MSE curves for 20 repetitions.

AIC shows less variation than that chosen by cross-validation and is close to the minimum. The results indicate that $tr(H_m)$ yields a satisfying approximation of the degrees of freedom of the additive model fit by GAMBoost. An additional advantage of the AIC criterion are the low computational costs as five-fold cross-validation takes about five times as much time as the use of the approximate AIC.

## 2.4 Standard deviations

For the derivation of standard deviations of function estimates the estimate of the linear predictor has to be investigated. The linear predictor after $m$ iterations decomposes into

$$\hat{\eta}_{(m)} = \hat{f}_0 + Z_1\hat{\beta}_{(m),1} + \cdots + Z_p\hat{\beta}_{(m),p}$$

where $\hat{f}_0$ stands for the intercept vector. With $j_{l+1} \in \{1, \ldots, p\}$ denoting the variable chosen in the $(l+1)$th iteration one has

$$Z_j\hat{\beta}_{(l+1),j} = \begin{cases} Z_j\hat{\beta}_{(l),j} & j_{l+1} \neq j \\ Z_j\hat{\beta}_{(l),j} + R_{l+1}(y - \hat{\mu}_{(l)}) & j_{l+1} = j \end{cases}$$

where $R_{l+1} = Z_j(Z_j^T W_l Z_j + \Lambda)^{-1} Z_j W_l D_l^{-1}$.

The form reflects that only one component is updated at a time. In closed form one obtains for the $j$th component

$$Z_j\hat{\beta}_{(l+1),j} = Z_j\hat{\beta}_{(l),j} + I(j_{l+1} = j)R_{l+1}(y - \hat{\mu}_{(l)})$$

where $I$ is the indicator function with $I(a) = 1$ if $a$ is true and $I(a) = 0$ if $a$ does not hold. With approximation $\hat{\mu}_{(m)} \approx H_m y$ one obtains

$$Z_j\hat{\beta}_{(l+1),j} \approx Z_j\hat{\beta}_{(l),j} + I(j_{l+1} = j)R_{l+1}(I - H_l)y$$

and therefore

$$Z_j\hat{\beta}_{(m),j} \approx Q_{m,j}y$$

where

$$Q_{m,j} = \sum_{l=0}^{m-1} I(j_{l+1} = j)R_{l+1}(I - H_{(l)}).$$

From

$$\hat{cov}(Q_{m,j}y) = Q_{m,j}\hat{cov}(y)Q_{m,j}^T \tag{14}$$

where $\hat{cov}(y) = diag(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_n^2)$ approximate confidence intervals of $\hat{f}_{(m),j} = Z_j\hat{\beta}_{(m),j} \approx Q_{m,j}y$ are found. For the normal model with identity link this is equal to the formulation suggested in Hastie & Tibshirani (1990).

Figure 4 shows the mean of the pointwise empirical confidence bands ($\pm 2$ standard error) (broken lines) and and the mean of pointwise confidence bands (dash-dotted lines) based on (14). The upper panels show the bands for binary response data (with $c_n = 2$) and the lower panels for Poisson response data (with $c_n = 1$), both generated from model (11) with $n = 200$. For each of the 50 repetitions the number of boosting steps has been determined by AIC (see Section 2.3) and so the empirical bands already incorporate the variation that stems from the stopping criterion. The solid lines indicate the mean of the fitted functions, the true functions are shown as dotted lines.

In the Poisson examples the estimated and the empirical confidence bands almost coincide. In the binary example the estimated confidence bands are slightly narrower than the empirical bands. Given that confidence bands in additive modelling, including the selection of smoothing parameters and variables, is always approximative the approximation by means of the hat matrix is rather good. Alternative confidence intervals by bootstrap techniques fail because of the heavy computational burden. In both examples the true function is within the empirical as well as within the estimated pointwise confidence bands.

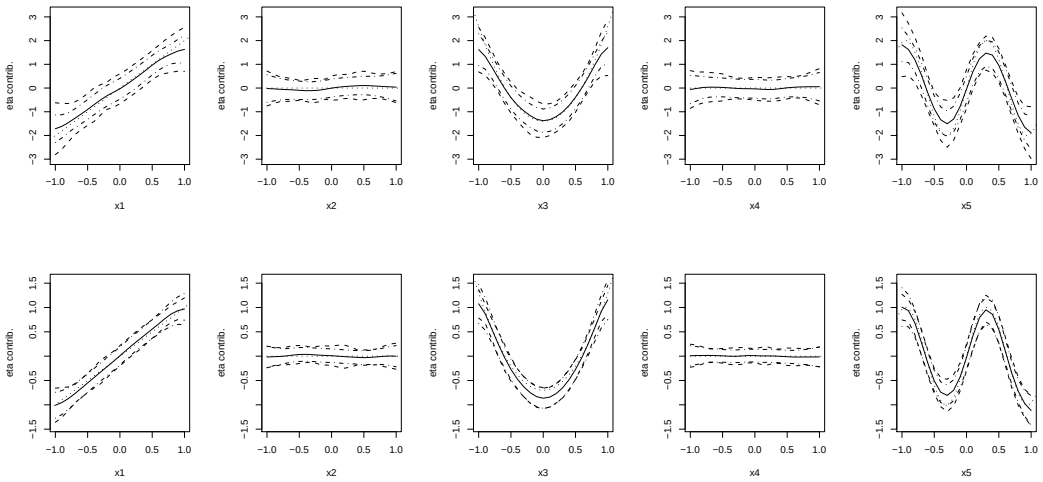


Figure 4: Mean empirical (broken lines) and estimated (dash-dotted lines) confidence bands for 50 fits to data from a binary response (upper panels) and a Poisson response model (lower panels). The solid lines indicate the estimates and the dotted lines the true functions.

# 3 Comparison to alternative fitting methods

## 3.1 GAMBoost and LogitBoost

For binary responses LogitBoost (Friedman et al., 2000) is a well established general boosting procedure. In contrast to GAMBoost where past boosting steps are represented by an intercept $\hat{\eta}_{(l)}$ LogitBoost uses the weighted least squares representation of Fisher scoring by considering the working response variables $z_i = y_i - p_{(l)}(x_i)/(p_{(l)}(x_i)(1-p_{(l)}(x_i)))$ with weights $w_i = p_{(l)}(x_i)(1-p_{(l)}(x))$ where $p_{(l)}(x_i)$ is the predicted probability of $y = 1$ based on the past boosting steps. With the resulting estimate $f_{(l+1)}$ the current function estimate $\eta_{(l)}(x)$ is updated by $\eta_{(l+1)}(x) = \eta_{(l)}(x) + \frac{1}{2}f_{(l+1)}(x)$ and $p_{(l+1)}$ is obtained by $p_{(l+1)}(x) = \exp(F_{(l+1)}(x))/(\exp(F_{(l+1)}(x)) + \exp(-F_{(l+1)}(x)))$. If LogitBoost is adapted to the present setting of componentwise boosting one obtains the estimation step for component $s$ as

$$f_s^{(l+1)} = Z_s(Z_s^T \hat{W}_L Z_s + \lambda\Lambda)^{-1} Z_s^T \hat{W}_L z_{(l)}$$

where $W_l = diag(p_{(l)}(x_1)(1-p_{(l)}(x_1), \ldots, p_{(l)}(x_n)(1-p_{(l)}(x_n))$ is the weight matrix corresponding to the working response $z_{(l)}^T = (z_1, \ldots, z_n)$ after step $l$. The corresponding

GAMBoost step for a binary response and the (canonical) logit link takes with

$\partial h(\eta_i)/\partial\eta = var(y_i)/\phi$ the form

$$f_{s,new} = Z_s(Z_s^T \hat{D} Z_s + \lambda\Lambda)^{-1} Z_s^T (y - \hat{\mu}).$$

Since for the canonical link $\hat{D} = \hat{W}_L$ and $W_L z_{(l)} = (y_1 - p_{(l)}(x_1), \ldots, y_n - p_{(l)}(x_n))^T$ it is seen that with the same starting point (i.e. $\hat{\mu}_i^{(l)} = p_{(l)}(x_i)$) the update is identical for LogitBoost and GAMBoost.

from GAMBoost because it uses a factor of 1/2 for the update and employs $\exp(F(x))/(\exp(F(x)) + \exp(-F(x)))$ instead of $\exp(F(x))/(1 + \exp(F(x)))$.

## 3.2 Empirical comparison

We present part of a simulation study in which the performance of GAMBoost is compared to other procedures for function estimation and variable selection in additive models. We used only procedures for comparison that explicitly allow for an exponential family response and give estimates for each model component (corresponding to a variable) separately together with standard deviations. The simplest approach is to use generalized linear models (GLM) which are restricted to linear predictors (McCullagh & Nelder, 1989). Generalized additive models (GAMs) weaken the linearity assumption and allow for smooth functions. For fitting Hastie & Tibshirani (1990) suggest to use the backfitting algorithm (to be denoted by "bfGAM") where each component function is iteratively re-estimated by a scatterplot smoother (e.g. smoothing splines). The amount of smoothness to be used is often specified by per-component degrees-of-freedom. An alternative approach is to use simultaneous estimation of all components with a design matrix that consist of basis function expansions of the predictors (Marx & Eilers, 1998). We will use the implementation of Wood (2000) (to be denoted by "wGAM") that incorporates automatic selection of smoothing parameters by generalized cross-validation (GCV). In contrast to GAMBoost which uses small updates of its estimates in each step, all other procedures use simultaneous estimation of all components or perform a separate re-estimation of components instead of an update. Therefore they are more prone to numerical problems with a large number of predictors and some kind of variable selection has to be performed. For GLM we use backward elimination as described e.g. by McCullagh & Nelder (1989) (denoted by "GLMse") and compare it to the full model. For generalized additive models with backward elimination (bfGAM) in addition to the variables to be used the degrees-of-freedom for each component have to be selected. We will use the step-wise approach that is suggested in Chambers & Hastie (1992). Starting from an initial model that includes all variables as linear predictors, for each predictor an upgrade and a downgrade by one level (with levels "not included","linear" and "smooth" with 4, 6 and 12 degrees of freedom) is evaluated and then that predictor is modified that results in a maximum decrease of the

AIC. This upgrade/downgrade procedure is repeated until no further decrease of AIC can be achieved. For wGAM we evaluate a similar procedure (denoted by "wGAMfb") that uses the levels "not included", "linear" and "smooth with parameter selected by GCV". Alternatively for a small number of predictors backward elimination is used (i.e. the initial model includes only smooth predictors and elimination of one variable is evaluated in each step). This will be denoted by "wGAMb". For GLM and the variants of GAM we used the implementations available for the statistical environment R (R Development Core Team, 2004) with parameters set to the default values.

The two flexible parameters of GAMBoost are the number of boosting steps and the penalty parameter $\lambda$. The first is chosen according to the approximate AIC given in (13). The second is chosen by an automatic procedure in such a way that the number of boosting steps chosen is greater or equal to 50. The reason for this rule is that despite being rather insensitive to the exact value of $\lambda$, often inadequate fits are obtained when the learner used is not weak enough and therefore only a small number of boosting steps is used.

The performance baseline that can for example be used to identify extreme overfitting of real models is defined by fitting a generalized linear model that uses only an intercept.

The simulation study is in two parts. First a GLM is assumed to hold and then an additive model is the data generating model. The assumed GLM is given by

$$\eta_i = c_n(x_{i1} + 0.7x_{i3} + 1.5x_{i5}). \tag{15}$$

with the response being drawn from a binomial or a Poisson distribution with parameter $\eta_i$. The variables $x_{ij}, j = 1, \ldots, p$, are drawn from a uniform distribution (with values between -1 to 1) where the influence of the three informative predictors on the respone (similar to model (11)) is controlled by the parameter $c_n$. The training sample size is $n = 100$. As a measure of goodness-of-fit we used the Kullback-Leibler distance on future observations as it is both adequate for the binary and the Poisson case. We evaluated it on a test sample of size 1000.

Figure 5 shows the Kullback-Leibler distances obtained from 50 repetitions for binary response data (upper panels) and for Poisson response data (lower panels) for $p = 5$ (left panels) and $p = 10$ right panels. The signal-to-noise ratio is chosen in such a way that generalized linear models, which are adequate for this data, can improve over the baseline intercept model. It is seen that all of the more advanced procedures that offer too much flexibility for the underlying model also outperform the baseline model. It comes as a surprise that the performance of GAMBoost is well comparable to the performance of GLM. Although the data generating model is a generalized linear model the added flexibility by estimating an generalized additive model does not affect the performance. In the contrary, considering the variability of estimates, GAMBoost is even more stable than

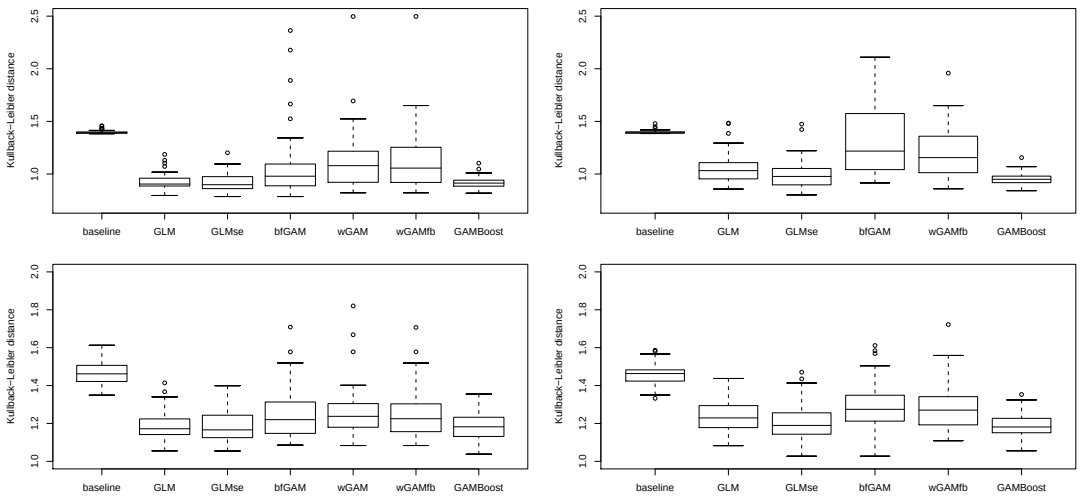


Figure 5: Kullback-Leibler distances for various procedures for binary (upper panels) and Poisson (lower panels) response data generated from model (15) ($c_n = 2$ for the binary and $c_n = 0.5$ for the Poisson case) with 5 (left panels) and 10 (right panels) variables (baseline: GLM with only intercept; GLM: GLM with backward elimination; bfGAM: backfitting GAM with forward/backward; wGAMb: Wood GAM with backward elimination; wGAMfb: Wood GAM with forward/backward; GAMBoost: GAMBoost with AIC stopping criterion).

GLM. The performance of the other GAM procedures is worse, where bfGAM performs better than wGAM procedures. In particular when the noise increases ($p = 10$) these procedures perform distinctly worse than GLM.

In the second part of the simulation study we generated data from the additive (non-linear) model (11) with binary and Poisson responses. The total number of predictors is varied ($p \in \{5, 10, 20, 50\}$) but the number of informative predictors is always three. The amount of influence of the latter on the response is controlled by the parameter $c_n$ which takes three values (1, 2 and 3 for the binary response examples and 0.5, 0.75 and 1 for the Poisson response examples) corresponding to "high noise", "some noise" and "low noise". The size of the training sample is kept fixed at $n = 100$. The training and test procedure is repeated for 20 times for each combination of $p$ and $c_n$.

Table 1 gives the mean Kullback-Leibler distances for the binary response examples. The number of instances where no fit could be obtained is given in parentheses. So the mean Kullback-Leibler distance given in these cells is based on a smaller number of repetitions (having possibly less complicated data structure). An empty cell indicates that a procedure could not be used at all. Given that there is a linear predictor in the data generating model, GLM performs rather good for small number of variables whereas for $p > 20$ its performance is very

Table 1: Mean Kullback-Leibler distance for binary response data generated from model (11) for varying numbers of predictors $p$ and varying amount of predictor influence $c_n$ (base: GLM with only intercept; GLM: GLM with full model; GLMse: GLM with backward elimination; bfGAM: backfitting GAM with forward/backward; wGAMb: Wood GAM with backward elimination; wGAMfb: Wood GAM with forward/backward; GB: GAMBoost; opt: GAMBoost with optimal number of boosting steps). The number of instances where no fit could be obtained is given in parentheses.

| $c_n$ | $p$ | base | GLM | GLMse | bfGAM | wGAM | wGAMfb | GB | opt |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 1.400 | 1.396 | 1.386 | 1.527 | 1.373 (10) | 1.384 | 1.314 | 1.270 |
| | 10 | 1.398 | 1.492 | 1.452 | 2.190 | - | 1.534 | 1.329 | 1.305 |
| | 20 | 1.399 | 1.707 | 1.538 | - | - | 1.887 (1) | 1.365 | 1.320 |
| | 50 | 1.395 | 4.554 (38) | 2.564 (35) | - | - | - | 1.471 | 1.356 |
| 2 | 5 | 1.400 | 1.333 | 1.325 | 1.441 (4) | 1.087 (20) | 1.140 | 1.041 | 0.985 |
| | 10 | 1.400 | 1.424 | 1.378 | 1.618 (14) | - | 1.353 | 1.058 | 1.043 |
| | 20 | 1.397 | 1.646 | 1.496 | - | - | 1.812 | 1.103 | 1.089 |
| | 50 | 1.395 | 4.836 (43) | 2.617 (40) | - | - | - | 1.233 | 1.182 |
| 3 | 5 | 1.400 | 1.298 | 1.286 | 1.046 (15) | 0.884 (27) | 0.960 | 0.842 | 0.790 |
| | 10 | 1.396 | 1.372 | 1.333 | 1.548 (31) | - | 1.210 | 0.852 | 0.839 |
| | 20 | 1.397 | 1.641 | 1.471 | - | - | 1.780 | 0.912 | 0.893 |
| | 50 | 1.397 | 6.233 (44) | 2.762 (42) | - | - | - | 1.002 | 0.975 |

bad. The use of backward elimination results in an improvement but for $p > 20$ it is still worse than the baseline . For small signal-to-noise ratio the performance of all GAM procedures (except GAMBoost) is very disappointing. For $p > 5$ the performance is worse than the fitting of an intercept model (base), for $p = 5$ it is within the range of the intercept model. The procedures are simply unable to draw information from the data. For bfGAM the effect still is present for a medium signal-to-noise ratio. wGAM shows better performance for $p = 5$ but for more than 5 variables wGAMb cannot be applied and wGAMfb is mostly outperformed by the intercept model. This is strongly contrasted by the GAMBoost procedure which (with one exception) strictly performs better than the simple intercept model and is very robust against the inclusion of noise variables. The performance is rather the same up to 20 variables, an increase in Kullback-Leibler distance is seen only for 50 variables where all of the other GAM procedures fail totally. So GAMBoost seems to be very resistant to overfitting. In addition the difference between GAMBoost (with the number of steps selected by approximate AIC) and the results that can be obtained with the optimal number of boosting steps is rather small. Selection by approximate AIC seems to work rather well.

Table 2 shows the mean Kullback-Leibler distances for the Poisson response data. Compared to the binary response examples bfGAM performs much better and

Table 2: Mean Kullback-Leibler distance for Poisson response data generated from model (11) for varying numbers of predictors $p$ and varying amount of predictor influence $c_n$ (base: GLM with only intercept; GLM: GLM with full model; GLMse: GLM with backward elimination; bfGAM: backfitting GAM with forward/backward; wGAMb: Wood GAM with backward elimination; wGAMfb: Wood GAM with forward/backward; GB: GAMBoost; opt: GAMBoost with optimal number of boosting steps). The number of instances where no fit could be obtained is given in parentheses.

| $c_n$ | $p$ | base | GLM | GLMse | bfGAM | wGAM | wGAMfb | GB | opt |
|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 5 | 1.484 | 1.449 | 1.442 | 1.352 | 1.299 (2) | 1.308 | 1.261 | 1.241 |
| | 10 | 1.479 | 1.525 | 1.490 | 1.433 | - | 1.466 | 1.309 | 1.283 |
| | 20 | 1.471 | 1.695 | 1.587 | - | - | 1.527 | 1.352 | 1.324 |
| | 50 | 1.449 | 4.332 | 2.538 | - | - | - | 1.435 | 1.370 |
| 0.75 | 5 | 2.027 | 1.849 | 1.840 | 1.500 | 1.276 (3) | 1.261 | 1.250 | 1.227 |
| | 10 | 2.013 | 1.968 | 1.928 | 1.561 | - | 1.390 | 1.344 | 1.317 |
| | 20 | 2.024 | 2.272 | 2.188 | - | - | 1.572 | 1.436 | 1.413 |
| | 50 | 1.971 | 6.589 | 3.861 | - | - | - | 1.579 | 1.549 |
| 1 | 5 | 3.11 | 2.703 | 2.694 | 2.069 | 1.196 (1) | 1.196 | 1.255 | 1.223 |
| | 10 | 3.1 | 2.907 | 2.874 | 2.186 | - | 1.322 | 1.381 | 1.353 |
| | 20 | 3.117 | 3.379 | 3.283 | - | - | 1.556 | 1.563 | 1.521 |
| | 50 | 3.117 | 3.079 | 12.297 | - | - | - | 1.917 | 1.885 |

outperforms the baseline and GLM for all examples where it can be fit, but still the wGAM procedures perform better. For a small number of variables and a large signal to noise-ratio the latter procedures are close to the performance of GAMBoost and even dominate it in three examples. In contrast for a large number of predictors and/or a small-signal-to-noise ratio GAMBoost clearly outperforms the GAM procedures. Again in most examples the performance of GAMBoost with approximate AIC is close to that with the optimal number of boosting steps.

# 4 Application

When analyzing patterns of admission/discharge of patients at a psychiatric hospital one important parameter is the number of readmissions after an initial stay at the hospital. For planning and optimization of and after the initial treatment variables are needed that are connected to the number of future admissions. To identify and explore such relations techniques that provide a graphical representation and do not impose too many restrictions (e.g. linearity) are wanted. In addition the focus is on homogeneous and therefore often small groups of patients which raises problems with many complex procedures. GAMBoost with Poisson

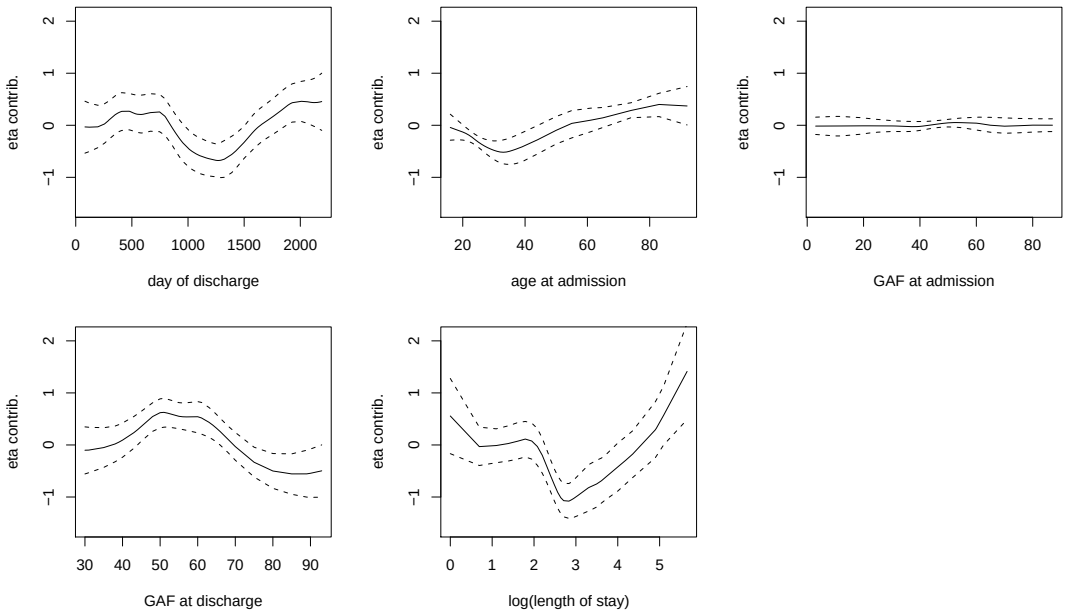


Figure 6: Estimated effects of several patient variables collected at the first stay at a psychiatric hospital on the number of readmissions within three years after discharge.

response in this situation offers flexible modelling for small datasets and helps in the selection of useful variables. Its ability to cope with a small signal-to-noise ratio is an additional benefit in this situation as the variables available for predicting the number of readmissions may not contain very much information.

To illustrate this application of GAMBoost it is applied to data from a German psychiatric hospital. There are 94 patients with a specific kind of substance abuse that have been treated in the hospital within a time of seven years. The response variable to be investigated is the number of readmissions within a time of three years after discharge. The metric variables available that are likely to carry information with respect to readmissions are "age at admission", (logarithm of) "length of stay (days)" and the "level of functioning" of the patient (GAF for "Global Assessment of Functioning") at admission and at discharge (with larger values indicating a better state of health). Because the patient data have been collected over a very long period the time of discharge is also included. We also included the binary variables "gender" (1:female, 0:male), "compulsory hospitalization" (1:yes, 0:no) and "comorbidity" (1:yes, 0:no). They are included into the per-step selection as predictors with one basis function and zero penalty.

Figure 6 shows the GAMBoost estimates for number of readmissions being modelled as a Poisson response. The estimate for the variable "age" suggests that

there may be two groups of patients, age $\leq 30$ and age $> 30$, where in the first group increasing age decreases the tendency to readmission whereas in the second group this relation is reversed.

The estimate for "length of stay" also indicates that there may be two groups of patients, the first corresponding to short interventions (one week or less) and a medium level of readmissions and the second corresponding to comprehensive treatment where the patients that receive more treatment may be the more problematic and therefore also have more readmissions. Looking at the "level of functioning" variables it comes as a surprise that the GAF at admission does not seem to be of high relevance with respect to the number of readmissions. Nevertheless there seems to be a reduction of the number of readmissions with increasing GAF score (at discharge) if it is above 50. The estimate for the variable "compulsory hospitalization" is 0.634 (standard deviation: 0.213), indicating that this type of patient is distinct with respect to the number of readmissions. For the other categorical variables there does not seem to be a significant effect ("sex": not selected in any boosting iteration; "comorbidity": estimate -0.074, standard deviation 0.054). It should be mentioned that bfGAM failed to find estimates for this data sets. Starting from a linear model problems with convergence made it impossible to increase the degrees of freedom of the components above a certain level and so a final model could not be obtained.

# 5 Concluding remarks

GAMBoost offers a procedure for fitting generalized additive models that overcomes deficiencies of commonly used fitting procedures. The selection of smoothing parameters, usually a task in $p$ dimensional space, is reduced to the selection of just one tuning parameter, the number of iterations. The amount of smoothing which varies across predictors is automatically adapted by the selection of variables to be updated in each iteration. The componentwise fitting procedure allows to use it in high dimensions where other GAM procedures fail. The derived approximation of the hat matrix turns out to be useful for the construction of an appropriate stopping criterion as well as for the derivation of standard errors.

It is straightforward to extend the method to multivariate generalized additive models like the proportional odds model with additive predictors (see Hastie & Tibshirani (1990) for traditional methods to estimate ordinal response models). After representing the smooth functions as weighted sum of basis functions one only has to adapt design matrices, score functions and Fisher matrices to the multivariate response case.

# References


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

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.

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

Chambers, J. M. & Hastie, T. J. (1992). *Statistical Models in S*. Pacific Grove, California: Wadsworth.

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

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

Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proc. Thirteenth International Conference*. Morgan Kaufman.

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

Hastie, T. J. & Tibshirani, R. J. (1990). *Generalized Additive Models*. London: Chapman & Hall.

Linton, O. B. & Nielsen, J. P. (1995). A kernel method of estimating structured nonparametric regression based on marginal integration. *Biometrika* **82**, 93–100.

Marx, B. D. & Eilers, P. H. C. (1998). Direct generalized additive modelling with penalized likelihood. *Computational Statistics and Data Analysis* **28**, 193–209.

McCullagh, P. & Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall, 2nd ed.

R Development Core Team (2004). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3.

Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics* **11**, 735–757.

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

Schimek, M. G. & Turlach, B. A. (2000). Additive and generalized additive models. In *Smoothing and Regression. Approaches, Computation and Application*, M. G. Schimek, ed. New York: Wiley, pp. 277–327.

Wood, S. N. (2000). Modelling and smoothing parameter estimation with multiple quadratic penalties. *Jounal of the Royal Statistical Society B* **62**, 413–428.