Tutz, Leitenstorfer:

Generalized smooth monotonic regression

Sonderforschungsbereich 386, Paper 417 (2005)

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

Generalized smooth monotonic regression

Gerhard Tutz, Florian Leitenstorfer
Ludwig-Maximilians-Universität München
Akademiestraße 1, 80799 München
{tutz,leiten}@stat.uni-muenchen.de

15.03.2005

Abstract

Common approaches to monotonic regression focus on the case of a unidimensional covariate and continuous dependent variable. Here a general approach is proposed that allows for additive and multiplicative structures where one or more variables have monotone influence on the dependent variable. In addition the approach allows for dependent variables from an exponential family, including binary and Poisson distributed dependent variables. Flexibility of the smooth estimate is gained by expanding the unknown function in monotonic basis functions. For the estimation of coefficients and the selection of basis functions a likelihood based boosting algorithm is proposed which is simply to implement. Stopping criteria and inference are based on AIC-type measures. The method is applied to several data sets.

Keywords: monotonic regression, additive models, likelihood based boosting

1 Introduction

In classical monotonic regression it is assumed that $E(y|x)$ is nondecreasing (or nonincreasing) in x for an independent variable x and dependent variable y. The theory of isotonic regression yielding a nonparametric solution in the form of a step function is treated extensively in Robertson, Wright & Dykstra (1988). The most widely used estimate is based on the Pool Adjacent Violators Algorithm (PAVA) which minimizes the (weighted) sum of squares and yields a step function that may have n levels, where n is the number of observations. The resulting estimate tends to overfitting and is aesthetically not convincing, the latter being also a handicap when the method is recommended to practioners. There have been several suggestions to obtain smooth estimates of the underlying monotonic function by combining isotonic regression with smoothing in a sequential fashion.

Friedman & Tibshirani (1984) recommended first smoothing and then isotonizing the data; Mukerjee (1988) suggested the reverse sequence. Mammen (1991) derived theoretical results for both approaches. For theoretical background see also Mammen, Marron, Turlach & Wand (2001). Alternative approaches to smooth isotonic regression have been given by Ramsay (1998) based on differential equations and Ramsay (1988) based on monotone splines. From a Bayesian point of view, monotonic regression has been treated more recently by Holmes & Heard (2003), Neelon & Dunson (2004) or Brezger & Steiner (2004).

It is surprising that most of the literature on monotonic regression focusses on the case of unidimensional covariate x and metrically scaled, continuous dependent variable y. In applications one often has multiple covariates $\mathbf{x}' = (x_1, \ldots, x_p)$ and it is known that $E(y|\mathbf{x})$ depends isotonically on some of the variables (say $x_1, \ldots, x_s$) but the effect of all variables has to be modelled. In particular if dichotomous variables have to be included, monotonicity can refer only to part of the variables. The second restriction refers to the type of distribution that is assumed for y. While generalized linear models (GLMs) and generalized additive models (GAMs) are nowadays common tools for modelling linear and additive relationships between regressors and dependent variables, isotonic modelling for binomial or Poisson distributed variables is rarely found. Because of lack of adequate methods least squares approaches have been widely used for binary data (e.g. Kelly & Rice 1991), in spite of the deficiencies of least squares approaches concerning the implicit variance heterogenity of binary data.

In the following a likelihood based approach is suggested that allows to consider dependent variables from a simple exponential family (e.g. binomial, Poisson, normal) and allows to consider an isotonic relationship between y and one or more of the covariates. The approach is based on an idea of Ramsay (1988) to expand the monotone function in a sum of monotone basis functions $f = \sum_i \alpha_i B_i$ and restrict the coefficients by the condition $\alpha_i \geq 0$. While Ramsay uses very few basis functions (integrated splines, one to at most three interior knots) we allow for more flexibility by using many basis functions (say 30). This seems to imply heavy computational burden when using common algorithms which are able to handle inequality constraints. In addition one might fear that estimates become wiggly. Both effects are prevented by using boosting strategies. By using componentwise boosting, monotonicity restrictions are easily incorporated and by controlling the number of boosting iterations (which corresponds to controlling smoothness) the resulting estimate turns out to be very stable.

An illustration of the resulting fits is given in Figure 1, where an additive model has been fitted to investigate the effect of weight and engine displacement on gasoline consumption for 60 automobiles (to be described in Section 3). It is seen that the monotonic fit (solid lines), selected by a corrected AIC criterion, is rather smooth and stable also in ranges with few data. In contrast, the

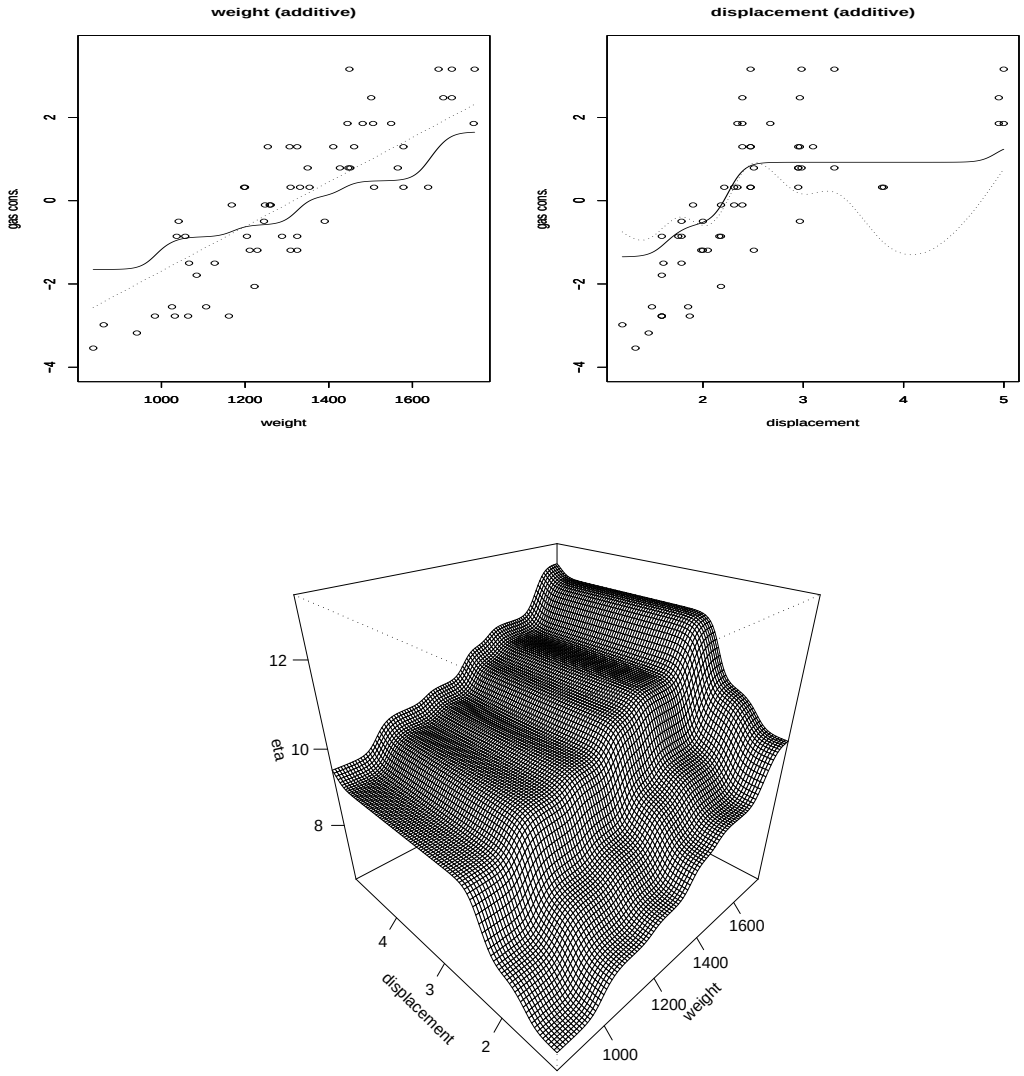

FIGURE 1: *Dependence of gasoline consumption on weight and displacement. Upper panels: solid lines show monotonic boosted regression, dashed lines show GAM. Lower panel: surface plot of monotonic boosted regression.*

fitting of an unconstrained additive model (dashed line, function `gam()` of the R library `mgcv`) yields a curve that is strongly detracked by single observations within ranges where few observations are found. The result is a strong dip in the displacement curve in the upper range of displacement.

In Section 2 we introduce the method of boosted monotonic estimates for the simpler case of continuous response. The properties of the method are investi-

gated in Section 3, where the automobile data are considered more extensively. In Section 4 the method is extended to more general response distributions by using likelihood based boosting techniques. Section 5 gives applications of the general distribution case. In Section 6 the method is extended to surface fitting where the response is assumed to depend monotonically on two covariates. Throughout the paper we consider monotonic regression to denote nondecreasing regression.

2 Monotonic regression by boosting techniques

We focus initially on conventional monotonic regression where y is a continuous variable. For a dependent variable y_i and covariates $x_i' = (x_{i1}, \ldots, x_{ip})$ an additive model is assumed,

$$y_i = \alpha_0 + \sum_{s=1}^{p} m_s(x_{is}) + \epsilon_i, \tag{1}$$

where α_0 is an intercept parameter, $m_s(.)$ is an unknown regression function for the sth covariate and $E(\epsilon_i) = 0$. Monotonic regression postulates that one or more of the regression functions are monotonic. If $m_s(.)$ is assumed to be monotonic one postulates

$$m_s(x) \geq m_s(z) \quad \text{if} \quad x > z. \tag{2}$$

Simple monotonic regression corresponds to the special case $p = 1$ with only one explanatory variable in the model. The additive model is an appealing way of structuring the influence of explanatory variables which allows to separate the (potentially monotonic) influence of single variables. Various procedures have been proposed for the estimation of non–monotonic additive models (see e.g. Hastie & Tibshirani (1990) for backfitting algorithms, Linton & Nielsen (1995) for the marginal integration approach, Marx & Eilers (1998) for direct estimates based on P–splines). In the latter approach, which has also been used by Ramsay (1988), flexibility of the predictor is obtained by expanding $m_s(x)$ in basis functions of the form

$$m_s(x) = \sum_{j=1}^{m} \alpha_j^{(s)} B_j^{(s)}(x)$$

for given basis functions $B_j^{(s)}$. In a regression spline approach, computationally convenient bases are B–splines which have been used recently by Eilers & Marx (1996) and Ruppert (2002). Ruppert (2002) mainly deals with the selection of the number of knots in spline regression. As will be demonstrated, the approach suggested here implies an automatic, data driven selection of knots.

In the following the monotonicity restriction is imposed by two modifications of the simple basis function approach. First (strictly) monotonically increasing basis functions are used and second the restriction

$$\alpha_j \geq 0, \quad j = 1, \ldots, m, \tag{3}$$

which is a sufficient condition for monotonicity is imposed. We will use two different choices of basis functions. The first set of basis functions consists of sigmoidal functions which are popular in the machine learning community in the fitting of hidden layer networks (e.g. Intrator & Intrator 2001). We consider the logistic type functions $B_j(x) = \{1/[1+\exp(c(x-t_j)]\} - 0.5$, where c specifies the steepness of the functions and $\{t_j\}$ is a given sequence of knots. Moreover, the functions are centered around zero. The second set of basis functions is based on splines. Following Ramsay (1988) we use integrated splines (I–splines), in particular I–splines of order 2, which have the closed form (centered around zero)

$$B_j(x) = \begin{cases} -0.5, & x < t_j, \\ \frac{(x-t_j)^2}{(t_{j+1}-t_j)(t_{j+2}-t_j)} - 0.5, & t_j \le x \le t_{j+1}, \\ 0.5 - \frac{(t_{j+1}-x)^2}{(t_{j+2}-t_j)(t_{j+2}-t_{j+1})}, & t_{j+1} \le x \le t_{j+2}, \\ 0.5, & x > t_{j+2}, \end{cases}$$

where $\{t_j\}$ is again a given sequence of knots.

In order to obtain estimates that fulfill restriction (3) we propose boosting techniques. Boosting has originally been developed in the machine learning community to improve classification procedures (e.g. Schapire 1990). With Friedman's (2001) gradient boosting machine it has been extended to regression modelling (Bühlmann & Yu 2003, Bühlmann 2004). The basis concept in boosting is to obtain a fitted function iteratively by fitting in each iteration a "weak" learner to the current residual. Componentwise boosting in the sense of Bühlmann & Yu (2003) means that in one iteration, only the contribution of one variable is updated. Boosting for monotonic fits uses a similar procedure, however componentwise does not refer to variables but to basis functions. Thus in each iteration only the contribution of one basis function is updated. This knotwise update makes it easy to control the monotonicity property (2). In addition, the procedure automatically selects a subset of basis functions (knots) which produce a proper fit. The weak learner that is used is ridge regression as proposed by Hoerl & Kennard (1970). For simplicity, we give the algorithm for the case $p = 1$ and therefore omit the index s in basis functions and the regression function $m(.)$. In matrix notation the data are given by $\mathbf{y} = (y_1, \ldots, y_n)'$, $\mathbf{x} = (x_1, \ldots, x_n)'$. The expansion into basis function yields the data set $(\mathbf{y}, \mathbf{B})$, where $\mathbf{B} = (B_1(\mathbf{x}), \ldots, B_m(\mathbf{x}))$, $B_j(\mathbf{x}) = (B_j(x_1), \ldots, B_j(x_n))'$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)'$.

MonBoost (continuous dependent variable)

Step 1 (Initialization)

Standardize $\mathbf{y}$ to zero mean, i.e. set $\hat{\alpha}_0 = \bar{y}$, $\hat{\boldsymbol{\alpha}}^{(0)} = (\bar{y}, 0, \ldots, 0)'$ and $\hat{\boldsymbol{\mu}}^{(0)} = (\bar{y}, \ldots, \bar{y})'$.

Step 2 (Iteration)

For $l = 1, 2, \ldots,$ compute the current residuals $\mathbf{u}^{(l)} = \mathbf{y} - \hat{\boldsymbol{\mu}}^{(l)}$.

1. *Fitting step*
   For $j = 1, \ldots, m$, compute the ridge regression estimator with tuning parameter λ for the linear regression model

   $$\mathbf{u}^{(l)} = \alpha_j B_j(\mathbf{x}) + \boldsymbol{\epsilon}.$$

   The resulting ridge estimate is given by $\hat{\alpha}_j = B_j(\mathbf{x})'\mathbf{u}^{(l)}/[B_j(\mathbf{x})'B_j(\mathbf{x}) + \lambda]$.

2. *Selection step*
   Choose from components $j \in \{1, \ldots, m\}$ the component $\hat{\gamma}^{(l)}$ such that $||\mathbf{u}^{(l)} - \hat{\alpha}_j B_j(\mathbf{x})||^2$ is minimized *and* the constraint $\hat{\alpha}_j^{(l+1)} = \hat{\alpha}_j^{(l)} + \hat{\alpha}_j \geq 0$ is satisfied, i.e. check if the potential update of component j is non–negative. If $\hat{\alpha}_j^{(l+1)} < 0$ for all j, stop. Otherwise, set $\hat{\gamma}^{(l)} = j$.

3. *Update*
   Set

   $$\hat{\alpha}_j^{(l+1)} = \begin{cases} \hat{\alpha}_j^{(l)} + \hat{\alpha}_j & j = \hat{\gamma}^{(l)} \\ \hat{\alpha}_j^{(l)} & \text{otherwise,} \end{cases}$$

   and

   $$\hat{\boldsymbol{\mu}}^{(l+1)} = \hat{\boldsymbol{\mu}}^{(l)} + \hat{\alpha}_{\hat{\gamma}^{(l)}} B_{\hat{\gamma}^{(l)}}(\mathbf{x}).$$

By construction, the fitted function

$$m(x) = \sum_{j=1}^{m} \hat{\alpha}_j^{(l)} B_j(x)$$

is monotonic for each iteration l. In order to prevent overfitting, it is necessary to include a stopping criterion. The often used cross-validation criterion is not recommended because it implies heavy computational effort. A much more appropriate criterion is the AIC criterion which balances goodness-of-fit with the degrees of freedom (for AIC in smoothing, see Hastie & Tibshirani 1990). In order to use the AIC criterion, the hat matrix of the smoother has to be given.

For the present procedure, it can be obtained in a similar way as for componentwise L2Boost in linear models. With $\mathbf{S}_l = B_{\hat{\gamma}^{(l)}}(\mathbf{x})B_{\hat{\gamma}^{(l)}}(\mathbf{x})'/[B_{\hat{\gamma}^{(l)}}(\mathbf{x})'B_{\hat{\gamma}^{(l)}}(\mathbf{x})+\lambda]$, $l = 1, 2, \ldots$ and $\mathbf{S}_0 = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'$, $\mathbf{1}_n = (1, \ldots, 1)'$, one has in the lth iteration

$$\hat{\boldsymbol{\mu}}^{(l+1)} = \hat{\boldsymbol{\mu}}^{(l)} + \mathbf{S}_l\mathbf{u}^{(l)} = \hat{\boldsymbol{\mu}}^{(l)} - \mathbf{S}_l(\hat{\boldsymbol{\mu}}^{(l)} - \mathbf{y}),$$

and therefore

$$\hat{\boldsymbol{\mu}}^{(l+1)} = \mathbf{H}_l\mathbf{y},$$

where

$$\mathbf{H}_l = \mathbf{I} - (\mathbf{I} - \mathbf{S}_0)(\mathbf{I} - \mathbf{S}_1)\cdots(\mathbf{I} - \mathbf{S}_l) = \sum_{j=0}^{l}\mathbf{S}_j\prod_{i=0}^{j-1}(\mathbf{I} - \mathbf{S}_i). \tag{4}$$

Since $\mathbf{H}_l$ corresponds to the hat matrix after the $(l+1)-$th iteration, $tr(\mathbf{H}_l)$ may be considered as degrees of freedom of the estimate. The suggested stopping rule for boosting iterations is based on the corrected AIC criterion proposed by Hurvich, Simonoff & Tsai (1998), given by

$$AIC_c(l) = \log(\hat{\sigma}^2) + \frac{1 + tr(\mathbf{H}_l)/n}{1 - (tr(\mathbf{H}_l) + 2)/n}, \tag{5}$$

where $\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y}-\hat{\boldsymbol{\mu}}^{(l)})'(\mathbf{y}-\hat{\boldsymbol{\mu}}^{(l)})$. Thus, the optimal number of boosting iterations, which in our framework plays the role of a smoothing parameter, is estimated by $l_{opt} = \arg\min_l AIC_c(l)$. The main objective of stopping the boosting procedure is to prevent overfitting. By applying the monotonicity restriction (3) in the selection step, a slightly higher resistance to overfitting is to be expected, when compared to unrestricted componentwise L2Boost (see Section 3).

In boosting procedures the number of iterations is the crucial tuning parameter which determines the amount of smoothing. The procedure is less sensitive to the choice of the shrinkage parameter λ which has to be large in order to obtain a weak learner. Since very large values of λ make more iterations necessary, in applications λ is chosen as large as possible but with the number of iterations until convergence being below 500.

Bühlmann (2004) suggests using a multiplicative shrinkage constant $\nu \in (0, 1]$ in the update step of componentwise L2Boost, rather than the application of ridge regression. It is easily seen that the two methods provide the same results if and only if

$$\nu = \frac{1}{1+\lambda} \quad \text{and} \quad B_j(\mathbf{x})'B_j(\mathbf{x}) = 1, \quad j = 1, \ldots, p. \tag{6}$$

In general, the Euclidean norm of basis function vectors is not equal to one, i.e. in our framework, shrinkage and ridge regression lead to different results, even if the first condition in (6) is fulfilled.

The extension to the additive model (1) with $p > 1$ is straightforward. Instead of m basis functions one has p sets of basis functions, one set for each variable.

The monotonicity constraint has to be fulfilled only for coefficients within one set of basis functions. The fitting step as well as the selection step include all of the basis functions with the effect that basis functions which refer to variables with stronger curvature are selected more often than basis functions which refer to flat functions. This automatic adaption to the curvature is an additional strength of the boosting approach. Although essentially only one tuning parameter, namely the number of iterations is needed, different amounts of smoothing are exerted for different variables.

3 Simulation and application

3.1 Simulations

In order to examine the performance of the suggested approach, we conduct a simulation study over a variety of data settings. Consider a regression model with continuous response, $y_i = f(x_i) + \epsilon_i$, where the ϵ_i are iid drawn from a $\mathcal{N}(0, \sigma^2)$–distribution and the x_i from a $U[0, 5]$–distribution, respectively. We investigate two types of monotonic functions:

$$f(x) = 3I(x > 2.5) \quad \text{(piecewise constant)}$$

and

$$f(x) = 3/[1 + \exp(10(x - 1))] + 2/[1 + \exp(5(x - 4))] \quad \text{(plateau)}.$$

For each setting, MonBoost is compared with GAM and PAVA, where the function `gam()` of the library `mgcv`, and the function `isoreg()` were used, respectively, both implemented in the statistical environment R (R Foundation for Statistical Computing 2004). In addition, two earlier approaches of smooth monotone regression are investigated. On the one hand, we implemented the algorithm of Friedman & Tibshirani (1984) (FT), where in a first step, y_i is fitted to x_i by an appropriate smoothing method, yielding y_i^* (in the present analyses, we again use `gam()`, which incorporates an automatic selection of the smoothing parameter by GCV). In the second step, PAVA is applied to the data set (y_i^*, x_i). Needless to say, this procedure does not yield continuous curves. On the other hand, in Mukerjee's (1988) approach, the data are isotonized first using PAVA, which results in the fitted data y_i^{**}. Then, $f(.)$ is estimated by

$$\hat{m}(x) = \frac{\sum_{i=1}^n k[(x - x_i)/h] y_i^{**}}{\sum_{i=1}^n k[(x - x_i)/h]},$$

i.e. the new data set (y_i^{**}, x_i) is piped into a kernel regression smoother. To ensure monotonicity of the estimate, a log–concave kernel $k(.)$ has to be used (for details, see Mukerjee 1988 or Mammen, Marron, Turlach & Wand 2001).

We applied the Gaussian kernel, which is a member of this class and select the optimal bandwidth by GCV. The method is referred to as MUK. Furthermore, we include a non–monotone version of componentwise boosting which works similar to MonBoost, but without any constraints on the estimated αs.

Componentwise boosting leads to an automatic, data driven choice of knots. That means, a sequence of m knots is available for selection by the boosting algorithm, but the actually chosen number of knots, i.e. the number of nonzero estimates of $\hat{\alpha}_j$, $j = 1, \ldots, m$, may be considerably lower than m. Note that before further proceedings, the predictor variable was always rescaled to $[0, 1]$. The two types of basis functions in the simulation study are: sigmoidal (scale parameter $c = 50$) and I–splines (degree 2). They have to be distinguished in terms of the pre–chosen sequence of knots. As simulations have shown, for sigmoidal functions the number of $m = [2n/3]$ knots placed at the $(j-1)/(m-1)$-quantiles, $j = 1, \ldots, m$, of the data x_i, $i = 1, \ldots, n$, performs quite reasonable. While the slope for sigmoidal functions is fixed, the slope of I–Splines depends on the number and placement of the knots. We followed Eilers & Marx (1996) and placed m equidistant interior knots in the domain [0,1], resulting in $m + 2$ basis components. The results of Ruppert (2002) show that functions similar to the ones investigated here can be well estimated by P–splines with 25 or less knots. Thus, in the simulations, 25 interior knots were chosen for I–spline basis functions.

The optimal number of boosting iterations was determined by the corrected AIC criterion from (5), where the maximal number of iterations was limited by $L = 500$. The choice of the ridge parameter λ is mainly guided by computational issues: Although a small value may lead to better prediction performance, it slows down the boosting algorithms, and then a large number of iterations is necessary to optimize the stopping criterion. In several experiments, $\lambda = 20$ turned out as a reasonable compromise between prediction performance and computational effort. We tried also shrinkage with factor $\nu = 0.1$: the difference compared to ridge regression was negligible.

Figure 2 shows typical data sets for the piecewise and plateau function, with $n = 30$, $\sigma = 1$. The true regression functions and several estimates are plotted. It is seen that especially for the piecewise function, MonBoost clearly outperforms the non–monotonic competitors and does also considerably better than PAVA, which might be thought of being appropriate in this discontinuous case. Also for the plateau example, MonBoost recovers best the typical shape of the underlying true function, while also Mukerjee's estimate performs well.

A strong criterion for the performance of an estimation method is out–of–sample prediction. Therefore, 1000 new observations $x_i^{(new)}, i = 1, \ldots, 1000$, were drawn from a $U[0, 5]$–distribution, and the averaged squared error (generalization

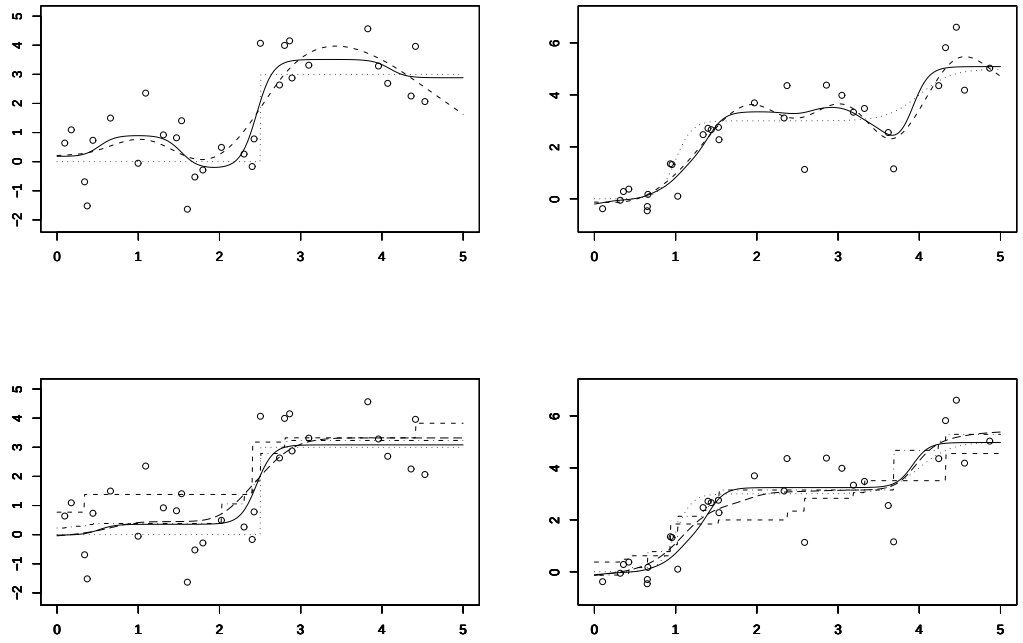

FIGURE 2: *Typical data set with $n = 30$, $\sigma = 1$, and estimates for the piecewise (left) and plateau (right) regression functions. True function (dotted), nonmonotonic estimators (above): componentwise L2Boost (solid), GAM (dashed). Monotonic estimators (below): MonBoost (solid), PAVA (dashed), FT (dotdashed), MUK (longdashed).*

error),

$$ASE = \frac{1}{1000}\sum_{i=1}^{1000}[\hat{f}(x_i^{(new)}) - f(x_i^{(new)})]^2,$$

was computed using the new observations. In Tables 1 and 2, for selected sample sizes and variances, the mean of the ASE is given over $S = 50$ simulated data sets for the various fitting methods. In each setting, the best two performers are given in bold numbers. It is seen from Table 1 that in the case of the piecewise constant function, MonBoost with either logistic basis functions or I-splines outperforms the other competitors in almost all of the experiments. Thereby, the I-spline approach does distinctively better for $n = 100$, whereas logistic basis functions seem to have some advantages for lower sample sizes. Among the alternative monotonic fitting methods, only MUK can compete with MonBoost in the low noise case $\sigma = 0.5$, while PAVA and FT in some cases do even worse than GAM. Interestingly, the unrestricted L2Boost estimators performs better than GAM in all experiments.

		GAM	PAVA	FT	MUK	L2B (log.)	MonB (log.)	L2B (ISpl.)	MonB (ISpl.)
$\sigma = 0.5$	$n = 20$	0.304	0.446	0.416	**0.222**	0.247	0.227	0.270	**0.226**
	$n = 30$	0.248	0.282	0.282	0.184	0.173	**0.160**	0.177	**0.143**
	$n = 100$	0.138	0.099	0.140	0.090	0.093	0.092	**0.066**	**0.055**
$\sigma = 1$	$n = 20$	0.595	0.609	0.577	0.456	0.427	**0.367**	0.518	**0.395**
	$n = 30$	0.488	0.433	0.450	0.399	0.328	**0.265**	0.381	**0.268**
	$n = 100$	0.210	0.160	0.192	0.166	0.152	**0.123**	0.139	**0.091**
$\sigma = 1.5$	$n = 20$	0.934	0.911	0.793	0.678	0.731	**0.599**	0.922	**0.676**
	$n = 30$	0.751	0.713	0.642	0.580	0.580	**0.438**	0.725	**0.461**
	$n = 100$	0.309	0.271	0.270	0.260	0.237	**0.169**	0.253	**0.153**

TABLE 1: *Piecewise constant function, averaged squared error over 50 simulated datasets.*

		GAM	PAVA	FT	MUK	L2B (log.)	MonB (log.)	L2B (ISpl.)	MonB (ISpl.)
$\sigma = 0.5$	$n = 20$	0.209	0.356	0.336	**0.145**	0.178	0.174	0.180	**0.167**
	$n = 30$	0.163	0.253	0.250	**0.097**	0.108	**0.101**	0.124	0.106
	$n = 100$	0.040	0.054	0.047	0.031	0.033	**0.025**	0.039	**0.030**
$\sigma = 1$	$n = 20$	0.489	0.566	0.508	**0.351**	0.436	0.398	0.452	**0.389**
	$n = 30$	0.359	0.411	0.401	**0.246**	0.296	0.257	0.329	**0.249**
	$n = 100$	0.110	0.137	0.107	**0.088**	0.097	**0.080**	0.122	0.090
$\sigma = 1.5$	$n = 20$	0.798	0.878	0.711	**0.595**	0.762	**0.686**	0.848	0.708
	$n = 30$	0.558	0.661	0.570	**0.440**	0.542	**0.444**	0.672	0.452
	$n = 100$	0.209	0.254	0.193	**0.154**	0.194	**0.157**	0.247	0.178

TABLE 2: *Plateau function, averaged squared error over 50 simulated datasets.*

The results for the plateau function are given in Table 2. It is seen that Mukerjee's estimator performs very well, but the MonBoost results are by all means comparable. The logistic basis function approach does slightly better than the I-spline approach in the high noise case $\sigma = 1.5$. PAVA as well as FT in most cases perform worse than GAM, which might be explained by the non-continuous character of the corresponding curve fits. Again, unrestricted L2Boost with logistic basis functions performs consistently better than GAM, while the unrestricted I-splines perform worse than GAM for $\sigma = 1.5$.

In Figure 3 exemplarily, the logarithm of the ASE is shown for the piecewise constant function (left panels) and plateau function (right panels) is shown, for $n = 30$ and $\sigma = 1$ (upper panels), as well as for $n = 100$ and $\sigma = 1.5$ (lower panels). It is again seen that the MonBoost estimates clearly perform best for the piecewise continuous function, whereas in the alternative scenarios, also MUK yields comparable results. The advantage of MonBoost over the unrestricted versions becomes more distinct for higher noise when monotonicity is harder to

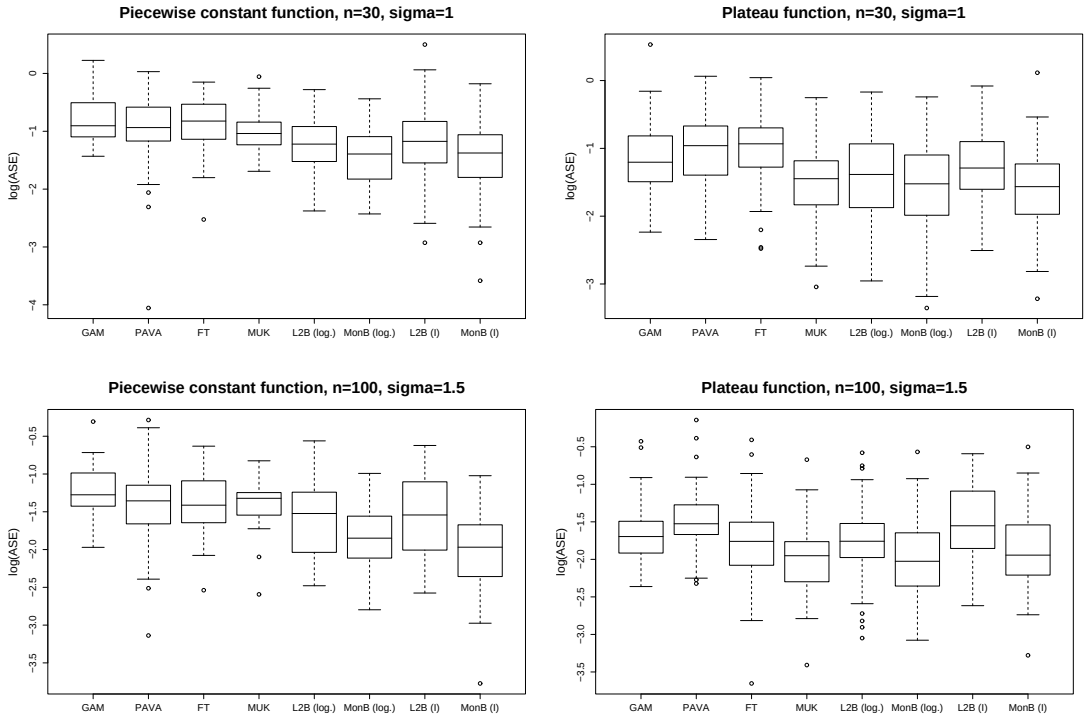

FIGURE 3: *Boxplots of* $\log(ASE)$ *for GAM, PAVA, FT, MUK, L2Boost with logistic basis functions, MonBoost with logistic basis functions, L2Boost with I-splines and MonBoost with I-splines for various models with continuous response.*

detect without using restricted estimators. In particular, Figure 3 shows that PAVA should no longer be used and that GAM is not a good solution for smooth monotone problems.

For the comparison of restricted and unrestricted boosting approaches it is interesting to look at the number of knots which have actually been chosen by the methods. Figure 4 shows the results for the piecewise constant (left panel) and the plateau function (right) panel, for a sample size of $n = 30$, with $m = 20$ logistic basis functions and noise $\sigma = 1$. It is seen how boosting adapts automatically to the complexity of the true underlying function: for estimation of the simpler structured piecewise function, on average fewer knots were chosen than for the plateau function. Furthermore, the monotonicity restriction results in sparser modeling. For example in the piecewise continuous case, in 10 out of 50 cases only one or two knots were actually chosen by MonBoost, while unrestricted L2Boost only in 2 cases chose such a small number of knots.

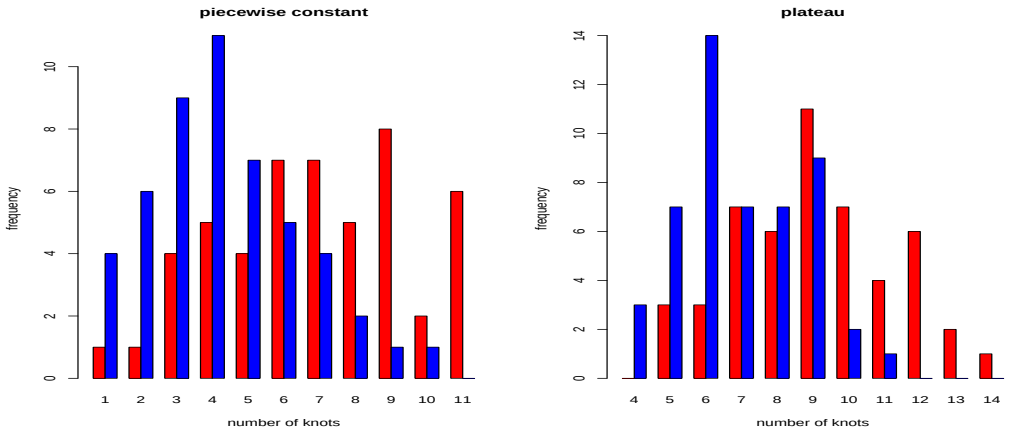

FIGURE 4: *Number of actually chosen knots by boosting algorithms (left bars: comp. L2Boost, right bars: MonBoost)over $S = 50$ simulations, piecewise continuous (left) and plateau function (right), $n = 30$, logistic basis functions with $m = 20$, $\sigma = 1$.*

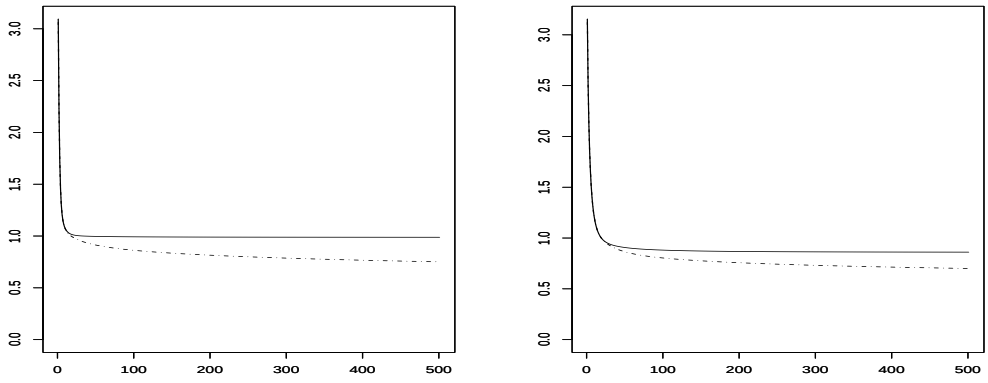

FIGURE 5: *$\hat{\sigma}^2$ vs. the number of boosting iterations, averaged over 50 simulations for MonBoost (solid) and componentwise L2Boost (dotdashed), piecewise constant (left panel) and plateau (right panel) function with $n = 30$ and $\sigma = 1$ with logistic basis functions.*

Additionally, Figure 5 shows the averaged estimated variance $\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l)})'(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l)})$ for MonBoost and unrestricted L2Boost plotted against the number of boosting iterations l, with sample size $n = 30$ and $\sigma = 1$. It is obvious that with increasing l, the variance of MonBoost remains fairly stable in both cases, indicating that there is resistance against overfitting. It is seen that in particular the monotonicity restriction is a strong tool to prevent overfitting.

	AIC_c	
covariates	MonBoost	GAM
s(WGT)	0.842	0.886
s(DPL)	0.958	1.089
s(WGT)+s(DPL)	0.657	0.744

TABLE 3: *AIC_c values for the various models for the automobile data set.*

3.2 Application to automobile data

The analysis uses a data set reported in the April 1990 issue of *Consumer Reports*, which is included in the R library `rpart` as `car.test.frame`. The data set contains several measures on 60 automobiles. Ramsay (1988) investigated an earlier version of the data set, where the measure of gasoline consumption [CON] was considered as dependent variable, and weight [WGT] and engine displacement [DPL] were explanatory variables. An isotonic relationship between weight as well as displacement and gasoline consumption is expected, as large or highly motorized cars are supposed to use more gas. Note that the variables of interests were converted: CON to liters per 100 km, WGT to kg and DPL to liters.

Three models were fitted: first WGT (model 1) and DPL (model 2) as single covariates using MonBoost as described in the simulation study, using 40 logistic basis function. In a second step, the additive model with WGT and DPL from Figure 1 was fitted (model 3), where the monotonicity restriction was set on both covariates. The other settings were the same as in the simulation studies. For all models, boosting was stopped by the corrected AIC from (5). The optimal numbers of iterations were $l_{opt} = 56$ (model 1), $l_{opt} = 55$ (model 2) and $l_{opt} = 64$ (model 3), respectively. The results for the corrected AIC criterion from (5), along with the corresponding results from GAM are given in Table 3.2. Obviously, the MonBoost approach leads to distinguishably more appropriate models, especially for the additive model.

4 Likelihood based boosting in generalized monotonic regression

An advantage of the proposed boosting approach to monotonic regression is that it may be extended to non-normal response variables. As in generalized linear models (e.g. McCullagh & Nelder 1989) it is assumed that $y_i|x_i$ has 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 θ_i is the canonical parameter and ϕ denotes the dispersion parameter. The link

between the mean $\mu_i = E(y_i|x_i)$ and the explanatory variable x_i is specified by

$$\mu_i = h(\eta_i),$$

where h is a given (strictly monotone) response function (the inverse of the link function $g = h^{-1}$), and the predictor $\eta_i = \eta(x_i)$ is a function of x. In contrast to generalized linear models where $\eta(x)$ is a linear predictor, unidimensional monotonic regression postulates that

$$\eta(x) = \sum_{s=1}^{p} m_s(x)$$

and the monotonicity condition (2) is fulfilled for one or more than one regression function $m_s(.)$. Monotonicity in η immediately transforms into monotonicity in the means.

In the unidimensional case the choice of h is arbitrary and may be based on computational convenience. In the following the canonical link is used, thus h is the logistic distribution function for binomial models, $h = \log$ for Poisson models and $h = id$ for normally distributed y (for alternative links see e.g. McCullagh & Nelder 1989).

We develop a componentwise likelihood based boosting algorithm for *generalized monotonic regression* (GMonBoost), which is similar to the GAMBoost algorithm given by Tutz & Binder (2004). We again consider $p = 1$, and the same notation as in Section 2 is used for the representation of the data set. In contrast to monotonic L2Boost, in a generalized context, the estimation of the intercept cannot be done in the simple way by setting $\alpha_0 = \bar{y}$. Therefore, we propose in each boosting iteration one step Fisher scoring based on generalized ridge regression for one selected component *and* the unpenalized intercept. In detail, that means for basis function B_j, $j = 1, \ldots, m$, maximizing the log-likelihood

$$l_p(\boldsymbol{\alpha}_j) = \sum_{i=1}^{n} l_i(\boldsymbol{\alpha}_j) - \frac{\lambda}{2}\boldsymbol{\alpha}_j'\boldsymbol{\Lambda}\boldsymbol{\alpha}_j,$$

where $l_i(\boldsymbol{\alpha}_j) = l_i(h(\mathbf{B}_j\boldsymbol{\alpha}_j))$ is the likelihood contribution of the ith observation with design matrix $\mathbf{B}_j = (\mathbf{1}, B_j(\mathbf{x}))$ and penalty matrix $\boldsymbol{\Lambda} = \text{diag}(0, 1)$, $\lambda > 0$ representing the ridge parameter. Derivation leads to the penalized score function

$$s_p(\boldsymbol{\alpha}_j) = \mathbf{B}_j'\mathbf{W}(\boldsymbol{\eta})\mathbf{D}(\boldsymbol{\eta})^{-1}(\mathbf{y} - h(\boldsymbol{\eta})) - \lambda\boldsymbol{\Lambda}\boldsymbol{\alpha}_j, \tag{7}$$

with $\mathbf{W}(\boldsymbol{\eta}) = \mathbf{D}(\boldsymbol{\eta})\boldsymbol{\Sigma}(\boldsymbol{\eta})^{-1}$, $\mathbf{D}(\boldsymbol{\eta}) = diag\{\partial h(\eta_1)/\partial\eta, \ldots, h(\eta_n)/\partial\eta\}$, $\boldsymbol{\Sigma}(\boldsymbol{\eta}) = diag\{\sigma_1^2, \ldots, \sigma_n^2\}$, $\sigma_i^2 = var(y_i)$, all of them evaluated at the current value of η.

GMonBoost (Generalized Monotonic Regression Boost)

Step 1 (Initialization)

Set $\hat{\alpha}_0^{(0)} = \bar{y}, \hat{\boldsymbol{\alpha}}^{(0)} = (\bar{y}, 0, \ldots, 0)', \hat{\boldsymbol{\eta}}^{(0)} = (\bar{y}, \ldots, \bar{y})'$ and $\hat{\boldsymbol{\mu}}^{(0)} = (h(\bar{y}), \ldots, h(\bar{y}))'$.

Step 2 (Iteration)

For $l = 1, 2 \ldots,$

1. *Fitting step*
   For $j = 1, \ldots, m$, compute the estimate from (7) based on one step Fisher scoring,
   $$\hat{\boldsymbol{\alpha}}_j = (\mathbf{B}_j' \mathbf{W}_l \mathbf{B}_j + \lambda \boldsymbol{\Lambda})^{-1} \mathbf{B}_j' \mathbf{W}_l \mathbf{D}_l{}^{-1} (\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l)}), \tag{8}$$
   where $\hat{\boldsymbol{\alpha}}_j = (\hat{\alpha}_{j,0}, \hat{\alpha}_{j,1})'$, $\mathbf{W}_l = \mathbf{W}(\hat{\boldsymbol{\eta}}^{(l)})$, $\mathbf{D}_l = \mathbf{D}(\hat{\boldsymbol{\eta}}^{(l)})$, and $\hat{\boldsymbol{\mu}}^{(l)} = h(\hat{\boldsymbol{\eta}}^{(l)})$.

2. *Selection step*
   For each component $j \in \{1, \ldots, m\}$ compute the potential update of the linear predictor, $\hat{\boldsymbol{\eta}}_{j,new} = \hat{\boldsymbol{\eta}}^{(l)} + \mathbf{B}_j \hat{\boldsymbol{\alpha}}_j$, where $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)'$. Choose the component such that the deviance $Dev(\hat{\boldsymbol{\eta}}_{j,new})$ is minimized *and* the constraint $\hat{\alpha}_j^{(l+1)} = \hat{\alpha}_j^{(l)} + \hat{\alpha}_{j,1} \geq 0$ is satisfied, i.e. check if the virtual update of component j is non–negative. If $\hat{\alpha}_j^{(m+1)} < 0$ for all j, break. Else, set $\hat{\gamma}^{(l)} = j$.

3. *Update*
   Set
   $$\begin{aligned} \hat{\alpha}_0^{(l+1)} &= \hat{\alpha}_0^{(l)} + \hat{\alpha}_{\hat{\gamma}^{(l)},0}, \\ \hat{\alpha}_j^{(l+1)} &= \begin{cases} \hat{\alpha}_j^{(l)} + \hat{\alpha}_{j,1} & j = \hat{\gamma}^{(l)} \\ \hat{\alpha}_j^{(l)} & \text{else,} \end{cases} \end{aligned}$$
   $$\hat{\boldsymbol{\eta}}^{(l+1)} = \hat{\boldsymbol{\eta}}^{(l)} + \mathbf{B}_{\hat{\gamma}^{(l)}} \hat{\boldsymbol{\alpha}}_{\hat{\gamma}^{(l)}} \quad \text{and} \quad \hat{\boldsymbol{\mu}}^{(l+1)} = h(\hat{\boldsymbol{\eta}}^{(l+1)}).$$

The derivation of a stopping criterion based on AIC is not as straightforward as in the case of a continuous dependent variable. However an approximation to the hat-matrix may be derived which shows satisfying properties. Let $\mathbf{M}_0 = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n'$ and $\mathbf{M}_l = \boldsymbol{\Sigma}_l^{1/2} \mathbf{W}_l^{1/2} \mathbf{B}_{\hat{\gamma}^{(l)}} (\mathbf{B}_{\hat{\gamma}^{(l)}}' \mathbf{W}_l \mathbf{B}_{\hat{\gamma}^{(l)}} + \lambda \boldsymbol{\Lambda})^{-1} \mathbf{B}_{\hat{\gamma}^{(l)}}' \mathbf{W}_l^{1/2} \boldsymbol{\Sigma}_l^{1/2}$, where $\mathbf{W}_l =$

$\mathbf{W}(\hat{\boldsymbol{\eta}}^{(l-1)})$, $l = 1, 2, \ldots,$ and $\boldsymbol{\Sigma}_l = \boldsymbol{\Sigma}(\hat{\boldsymbol{\eta}}^{(l-1)})$. As shown in the Appendix, an approximate hat-matrix is given by

$$\mathbf{H}_l = \mathbf{I} - (\mathbf{I} - \mathbf{M}_0)(\mathbf{I} - \mathbf{M}_1) \cdots (\mathbf{I} - \mathbf{M}_l) = \sum_{j=0}^{l} \mathbf{M}_j \prod_{i=0}^{j-1} (\mathbf{I} - \mathbf{M}_i), \tag{9}$$

with $\hat{\boldsymbol{\mu}}^{(l)} \approx \mathbf{H}_l \mathbf{y}$.

Similar to MonBoost, we consider $tr(\mathbf{H}_l)$ as measure for the complexity of the fit, and use the corresponding AIC criterion

$$AIC(l) = Dev_l + 2tr(\mathbf{H}_l), \tag{10}$$

where $Dev_l = 2\sum_{i=1}^{n}[l_i(y_i) - l_i(\hat{\eta}_i^{(l)})]$ denotes the deviance of the fitted model in the lth boosting step, with l_i denoting the contribution of the ith observation to the log-likelihood. The optimal number of boosting iterations is defined by $l_{opt} = \arg\min_l AIC(l)$.

It should be noted that for binary variables the fitting step corresponds to LogitBoost as proposed by Friedman, Hastie & Tibshirani (2000) with the modification that it is used in a componentwise way with the components being given by basis functions. In the general case of distributions from an exponential family it represents a likelihood based boosting approach. It may be seen as generalization of the L2Boost which refers to normally distributed responses.

5 Applications of generalized monotonic regression

5.1 Simulation results

It is worthwhile to investigate the performance of GMonBoost since usually less information is available from binary or Poisson distributed responses than from normally distributed responses. In our simulation study, a binary regression model with canonical link is considered, i.e. we draw binary response from $y_i \sim \mathcal{B}(1, 1/[1+\exp(-\eta_i)])$, where $\eta_i = \eta(x_i)$ is specified by a monotone function. The two types of investigated functions are similar to the ones used before: $\eta(x) = \gamma[-I(x \leq 2.5) + I(x > 2.5)]$ (piecewise constant) as well as $\eta(x) = \gamma\{3/[1 + \exp(10(x-1))] + 3/[1+\exp(5(x-4))] - 3\}$ (plateau). The constant γ controls the influence of the linear predictor on the response. Small values for γ correspond to smaller signal-to-noise ratio, whereas high values correspond to higher signal-to-noise ratio. We again compare GMonBoost with GAM and PAVA. In addition, we apply a generalization of the Friedman-Tibshirani approach (GFT). Therefore, in step one, y_i is fitted on x_i with GAM using the canonical link, yielding estimates η_i^* for the linear predictors. Accordingly, the data set (η_i^*, x_i) is isotonized using PAVA. To our knowledge this generalization, which explicitly takes into account the binary character of the data, has not been investigated before. Furthermore,

		GAM	PAVA	GFT	GenB (log.)	GMonB (log.)	GenB (ISpl.)	GMonB (ISpl.)
$\gamma = 1$	$n = 20$	0.140	0.148	**0.101**	0.210	**0.107**	0.282	0.120
	$n = 30$	0.111	0.112	**0.088**	0.159	**0.079**	0.215	0.089
	$n = 100$	0.039	0.046	0.034	0.062	**0.027**	0.085	**0.031**
$\gamma = 2$	$n = 20$	0.144	0.225	0.216	0.145	**0.130**	0.144	**0.121**
	$n = 30$	0.104	0.149	0.151	0.092	**0.078**	0.102	**0.083**
	$n = 100$	0.083	0.064	0.081	0.062	**0.052**	0.064	**0.042**
$\gamma = 3$	$n = 20$	0.170	0.208	0.195	0.167	**0.131**	0.187	**0.136**
	$n = 30$	0.127	0.137	0.135	0.123	**0.084**	0.144	**0.089**
	$n = 100$	0.097	0.060	0.095	0.064	**0.047**	0.072	**0.040**

TABLE 4: *Piecewise constant function, Kullback–Leibler error over 50 simulated datasets.*

the non–monotonic version of GMonBoost is included, where no restrictions on the estimates are imposed.

Basis functions and the number and location of knots were chosen in the same way as in Section 3. Boosting was stopped by using the AIC criterion, as described in Section 4, with a maximum number of $L = 500$ iterations. In this generalized context, it turned out that a fairly small ridge parameter of $\lambda = 3$ suffices for a reasonable trade–off between computational convenience and minimizing prediction error. Experiments with higher values for λ suggested that especially in the non–monotone case, the AIC does not accomplish a distinct minimum within 500 boosting iterations for numerous data sets.

The out–of–sample prediction performance of the various methods is measured by drawing new observations $x_i^{(new)}, i = 1, \ldots, 1000$, from a $U[0, 5]$–distribution and computing the averaged Kullback-Leibler distance,

$$AKL = \frac{1}{1000} \sum_{i=1}^{1000} KL(\hat{\pi}_i, \pi_i),$$

where $KL(\hat{\pi}_i, \pi_i) = \hat{\pi}_i \log(\frac{\hat{\pi}_i}{\pi_i}) + (1 - \hat{\pi}_i) \log(\frac{1-\hat{\pi}_i}{1-\pi_i})$, with $\hat{\pi}_i = h[\hat{\eta}(x_i^{(new)})]$ and $\pi_i = h[\eta(x_i^{(new)})]$. In Table 4, the results for the piecewise constant function are given. It is seen that GMonBoost with logistic basis functions performs well in almost all settings, while also the I-spline approach yields comparable results. PAVA is clearly outperformed, and GFT yields good estimates only in the case of low signed strength and small sample size, but deteriorates in the low noise case of $\gamma = 3$.

From Table 5, the superiority of GMonBoost for the plateau function, compared to the other competitors, becomes apparent. GMonBoost with logistic basis functions is among the two best performers in all of the 9 settings. GFT does slightly better than the I-spline approach in the case $\gamma = 1$ but yields considerably worse results for lower noise problems. Interestingly, for small γ, GAM

		GAM	PAVA	GFT	GenB (log.)	GMonB (log.)	GenB (ISpl.)	GMonB (ISpl.)
$\gamma = 1$	$n = 20$	0.154	0.152	**0.121**	0.216	**0.112**	0.272	0.129
	$n = 30$	0.097	0.122	**0.093**	0.142	**0.083**	0.190	0.094
	$n = 100$	0.038	0.044	**0.032**	0.053	**0.031**	0.078	0.034
$\gamma = 2$	$n = 20$	0.158	0.150	0.167	0.188	**0.112**	0.221	**0.118**
	$n = 30$	0.108	0.120	0.123	0.127	**0.081**	0.167	**0.088**
	$n = 100$	0.037	0.041	0.034	0.043	**0.025**	0.060	**0.029**
$\gamma = 3$	$n = 20$	0.207	0.166	0.227	0.221	**0.138**	0.239	**0.127**
	$n = 30$	0.136	0.125	0.162	0.132	**0.086**	0.178	**0.093**
	$n = 100$	0.054	0.040	0.052	0.050	**0.030**	0.063	**0.032**

TABLE 5: *Plateau function, Kullback–Leibler error over 50 simulated datasets.*

yields better estimates than GenBoost without restrictions. This might be caused by some data sets where a distinct minimum of the AIC has not been reached after the maximum number of 500 boosting iterations.

5.2 Bronchitis data

As an illustration of the method, we consider the bronchitis data, previously analyzed by Küchenhoff & Ulm (1997) and Küchenhoff & Carroll (1997). The data were collected in a dust burdened mechanical engineering plant in Munich between 1960 and 1977. The binary response variable is the presence ($y = 1$) or absence ($y = 0$) of a chronic bronchitic reaction [CBR], measured on 1246 workers of the factory. For the following analysis, we follow Küchenhoff & Carroll (1997) and consider only the subpopulation of $n = 921$ smokers, from which 241 (26.2%) were diagnosed a CBR. The regressor variables of interest are the average dust concentration in the working area over the period of time in question [dust], measured in mg/m^3, and the duration of exposure [expo] in years. The data are visualized in Figure 6 on the log scale. Since dust may take the value zero, log(dust+1) is used rather than log(dust), compare also Küchenhoff & Carroll (1997). Since the probability for occurrence of CBR is supposed to increase with dust concentration and duration of exposure on dust, we assume an isotonic relationship between the two covariates and CBR.

We investigated several models, first log(dust+1) (model 1) and log(expo) (model 2), respectively, were considered as single covariates. Then, an additive model with log(dust+1) and log(expo) as independent variables has been fitted. We used GMonBoost with I-splines and $m = 25$ interior knots. For the additive model, 25 interior knots were taken for each of the covariates. All other settings were the same as in the simulations. The values of the AIC criterion (10) for the various models are given in table 6, along with the corresponding results of GAM. Boosting stopped after 56 (model 1), 15 (model 2) and 14 (model 3) iterations. It

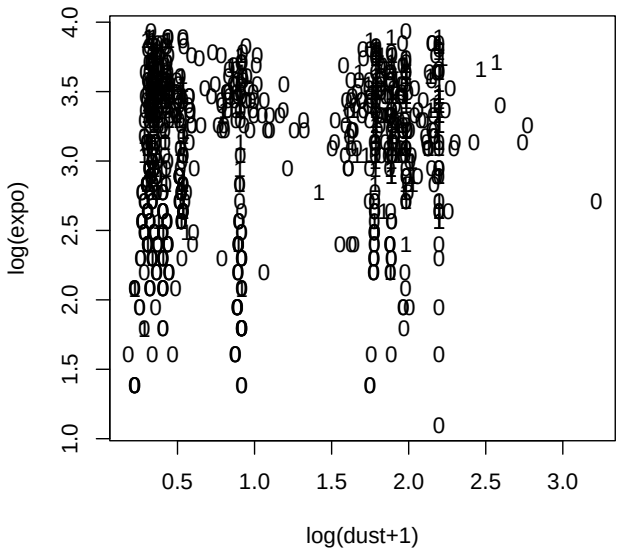

FIGURE 6: *Scatterplot drawn from the subpopulation of $n = 921$ smokers of the Bronchitis data set (logarithmic scaling).*

	AIC	
covariates	GMonBoost	GAM
s(log(dust+1))	1027.64	1032.76
s(log(expo))	1007.96	1014.70
s(log(dust+1))+s(log(expo))	982.52	995.79

TABLE 6: *AIC values for Bronchitis data set.*

is seen that the fits obtained by GMonBoost dominate the GAM results for each of the three considered models, and that the monotonic additive model yields the best fit.

Figure 7 shows the the estimates of the nonparametric functions s(log(dust+1)) and s(log(expo)) for the additive model (lower panel). In addition, Figure 7 shows in the upper panel the curves for the model s(dust)+s(expo), where the original scaling of dust and expo has been used. The solid lines show the GMonBoost fits, compared to GAM (dashed lines). It is seen that in the original scaling of covariates, one extreme observation pulls the GAM curve downwards in the range where dust concentration is greater than 12. GMonBoost yields a nearly

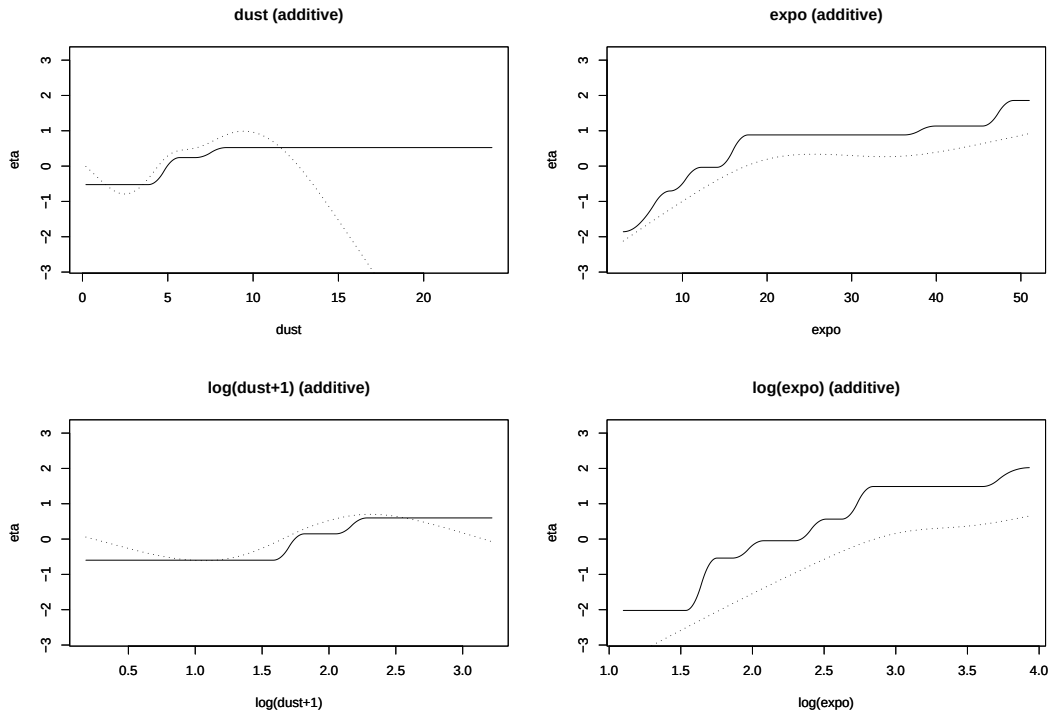

FIGURE 7: *Estimated curves for model 3. Upper panels: original scaling, lower panels: logarithmic scaling. Solid lines: GMonBoost, Dashed lines: GAM.*

constant, nondecreasing curve in this critical region. This effect is attenuated by using the transformation log(dust+1). Nevertheless also for logarithmic scaling, the GAM estimate is not monotonic.

6 Monotonic surface fitting

The given framework may be easily extended to incorporate monotonic surface smoothing. For simplicity, consider the case of two variables with the data given by (y_i, x_{i1}, x_{i2}), $i = 1, \ldots, n$, and assume that the predictor $\eta(x_{i1}, x_{i2})$ is monotone in both components but not necessarily additive.

In order to allow for interaction effects one considers the knots $t_{ij} = (t_i^{(1)}, t_j^{(2)}) \in \mathbb{R}^2$ with ordering $t_1^{(1)} \leq t_2^{(1)} \leq \ldots$, $t_1^{(2)} \leq t_2^{(2)} \leq \ldots$, and corresponding centered basis functions $B_{ij}(x_1, x_2) = \phi_i^{(1)}(x_1)\phi_j^{(2)}(x_2) - 0.5$, where $\phi_i^{(1)}(.)$, $\phi_j^{(2)}(.)$ are monotonic basis functions, e.g. sigmoidal function or I-splines, with values in [0,1], which are linked to knots t_i, t_j, respectively. Instead of fitting the model directly we propose to fit it in two stages. Following the hierarchical order, first the additive model is fitted and then the model which has an additional interaction

term. More concrete, in the first stage the monotonic additive model

$$\eta_a(x_{i1}, x_{i2}) = \alpha_0 + \sum_{j=1}^{m} \alpha_j^{(1)} B_j^{(1)}(x_{i1}) + \sum_{j=1}^{m} \alpha_j^{(1)} B_j^{(2)}(x_{i2})$$

is fitted, yielding $\hat{\eta}_a(x_{i1}, x_{i2})$. In the second stage, the model

$$\eta(x_{i1}, x_{i2}) = \hat{\eta}_a(x_{i1}, x_{i2}) + \sum_{i,j} \alpha_{ij} B_{ij}(x_{i1}, x_{i2})$$

is fitted, where $\hat{\eta}_a$ is treated as an offset. The essential modification that is needed in the second stage concerns the constraints in the selection step. Only updates are taken into consideration for which

$$\alpha_{ij} \geq 0$$

holds. In addition, the basis functions in the fitting step have to be replaced by $B_{ij}(\mathbf{x}) = (B_{ij}(x_{11}, x_{12}), \ldots, B_{ij}(x_{n1}, x_{n2}))$. By computing the corresponding AIC criterion the algorithm decides if the additional interaction term is necessary for an appropriate fit or not.

We applied monotonic surface fitting to the Bronchitis data. Therefore, two dimensional I-spline basis functions were used, specified by a grid of 10×10 equidistant interior knots. Boosting stopped after 63 additional iterations, yielding a slightly improved AIC of 981.89 (logarithmic scaling). In Figure 8 surface plots for the additive model without interaction (upper panel), and with interaction (lower panel), fitted by GMonBoost, are given. It is seen that the additional interaction leaves the overall surface unchanged. Only in extreme ranges of dust and expo where only a few data have been observed, a deviation from the additive model is fitted. However, the small difference in AIC is hardly supporting the necessity of an interaction effect. For the automobile data the case for the additive model is even stronger. When trying to improve the model by including an interaction effect, no further two-dimensional basis function is selected. Corrected AIC cannot be improved by including interaction effects.

7 Concluding remarks

The proposed framework is very flexible with regard to the handling of monotonic components. While some of the additive components may be assumed to be monotonic, others can be fitted without the assumption of monotonicity. Both types of components are estimated by the same algorithm, which slowly fits by selection of basis functions and ridging. The only difference is that under the assumption of monotonicity the constraints on coefficients are taken into account whereas they are ignored in unconstrained fitting.

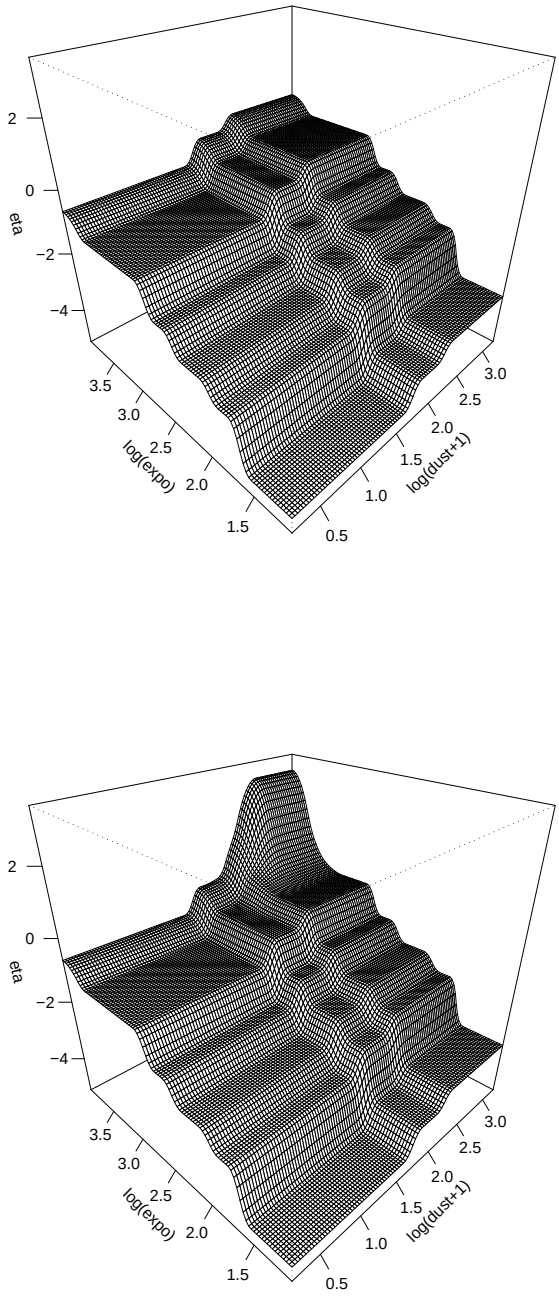

FIGURE 8: *Surface plots of the additive model without (above) and with (below) interaction for the Bronchitis data, both fitted by GMonBoost (logarithmic scaling).*

The inclusion of categorical variables and parametrically specified variables is straightforward. In each fitting step all the parametric terms *and* the basis functions under investigation are taken into the fitted model. In the selection step it is determined which update is performed. Alternatively one could treat the parametric terms in the same way as basis functions and select among the set of parametric terms and basis functions. The latter approach is not recommended since we found some bias in favor of smooth continuous variables over the less informative categorical variables which are rarely selected for an update.

Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (SFB 386, "Statistical Analysis of Discrete Structures").

References

BÜHLMANN, P. (2004). Boosting for high–dimensional linear models. Technical Report, ETH Zürich.

BÜHLMANN, P. AND YU, B. (2003). Boosting with the L_2loss: regression and classification. *Journal of the American Statistical Association 98*, 324–339.

BREZGER, A. AND STEINER, W. J. (2004). Monotonic regression based on bayesian p-splines: an application to estimating price response functions from store-level scanner data. SFB Discussion Paper 331, LMU München.

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

FRIEDMAN, J. H. (2001). Greedy function approximation: a gradient boosting machine. *The Annals of Statistics 29*, 337–407.

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

FRIEDMAN, J. H. AND TIBSHIRANI, R. (1984). The monotone smoothing of scatterplots. *Technometrics 26*, 243–250.

HASTIE, T. AND TIBSHIRANI, R. (1990). *Generalized Additive Models*. London: Chapman & Hall.

HOERL, A. E. AND KENNARD, R. W. (1970). Ridge regression: Bias estimation for nonorthogonal problems. *Technometrics 12*, 55–67.

HOLMES, C. C. AND HEARD, N. A. (2003). Generalized monotonic regression using random change points. *Statistics in Medicine 22*, 623–638.

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

INTRATOR, O. AND INTRATOR, N. (2001). Interpreting neural–network results: a simulation study. *Computational Statistics & Data Analysis 37*, 373–393.

KÜCHENHOFF, H. AND ULM, K. (1997). Comparison of statistical methods for assesing threshold limiting values in occupational epidemiology. *Computational Statistics 12*, 249–264.

KELLY, C. AND RICE, J. (1991). Monotone smoothing with application to dose–response curves and the assessment of synergism. *Biometrics 46*, 1071–1085.

KÜCHENHOFF, H. AND CARROLL, R. J. (1997). Segmented regression with errors in predictors: Semi-parametric and parametric methods. *Statistics in Medicine 16*, 169–188.

LINTON, O. B. AND NIELSEN, J. B. (1995). A kernel method of estimating structured nonparametric regression based on marginal integration. *Biometrika 82*, 93–100.

MAMMEN, E. (1991). Estimating a smooth monotone regression function. *The Annals of Statistics 19*, 724–740.

MAMMEN, E., MARRON, J., TURLACH, B., AND WAND, M. (2001). A general projection framework for constrained smoothing. *Statistical Science 16*(3), 232–248.

MARX, D. B. AND EILERS, P. H. C. (1998). Direct generalized additive modelling with penalized likelihood. *Comp. Stat. & Data Analysis 28*, 193–209.

MCCULLAGH, P. AND NELDER, J. A. (1989). *Generalized Linear Models* (2nd ed.). New York: Chapman & Hall.

MUKERJEE, H. (1988). Monotone nonparametric regression. *The Annals of Statistics 16*, 741–750.

NEELON, B. AND DUNSON, D. B. (2004). Bayesian isotonic regression and trend analysis. *Biometrics 60*, 398–406.

R Foundation for Statistical Computing (2004). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.

RAMSAY, J. O. (1988). Monotone regression splines in action. *Statistical Science 3*, 425–461.

RAMSAY, J. O. (1998). Estimating smooth monotone functions. *Journal of the Royal Statistical Society B 60*(2), 365–375.

ROBERTSON, T., WRIGHT, F. T., AND DYKSTRA, R. L. (1988). *Order–Restricted Statistical Inference*. New York: Wiley.

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.

TUTZ, G. AND BINDER, H. (2004). Generalized additive modelling with implicit variable selection by likelihood based boosting. SFB Discussion Paper 401, LMU München.

Appendix

Approximate hat-matrix for GMonBoost:
In the lth iteration of GMonBoost, after the selection of $\hat{\gamma}^{(l)}$, the update is given by

$$\hat{\boldsymbol{\alpha}}_{\hat{\gamma}^{(l)}} = (\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l\mathbf{B}_{\hat{\gamma}^{(l)}} + \lambda\boldsymbol{\Lambda})^{-1}\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l{\mathbf{D}_l}^{-1}(\mathbf{y} - \boldsymbol{\mu}^{(l)}),$$

where $\mathbf{W}_l = \mathbf{W}(\hat{\boldsymbol{\eta}}^{(l-1)})$ and $\mathbf{D}_l = \mathbf{D}(\hat{\boldsymbol{\eta}}^{(l-1)})$. From the update step of the algorithm, one has

$$\begin{aligned}
\hat{\boldsymbol{\eta}}^{(l)} - \hat{\boldsymbol{\eta}}^{(l-1)} &= \hat{\boldsymbol{\eta}}^{(l-1)} + \mathbf{B}_{\hat{\gamma}^{(l)}}\hat{\boldsymbol{\alpha}}_{\hat{\gamma}^{(l)}} - \hat{\boldsymbol{\eta}}^{(l-1)} \\
&= \mathbf{B}_{\hat{\gamma}^{(l)}}\hat{\boldsymbol{\alpha}}_{\hat{\gamma}^{(l)}} \\
&= \mathbf{B}_{\hat{\gamma}^{(l)}}(\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l\mathbf{B}_{\hat{\gamma}^{(l)}} + \lambda\boldsymbol{\Lambda})^{-1}\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l{\mathbf{D}_l}^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l)}).
\end{aligned}$$

By using a first order Taylor approximation, $h(\hat{\eta}) \approx h(\eta) + (\partial h(\eta)/\partial\eta^T)(\hat{\eta} - \eta)$, one obtains

$$\begin{aligned}
\hat{\boldsymbol{\mu}}^{(l)} &= h(\hat{\boldsymbol{\eta}}^{(l-1)} + \mathbf{B}_{\hat{\gamma}^{(l)}}\hat{\boldsymbol{\alpha}}_{\hat{\gamma}^{(l)}}) \\
&\approx \boldsymbol{\mu}^{(l-1)}\mathbf{D}_l(\hat{\boldsymbol{\eta}}^{(l)} - \hat{\boldsymbol{\eta}}^{(l-1)})
\end{aligned}$$

and therefore

$$\hat{\boldsymbol{\eta}}^{(l)} - \hat{\boldsymbol{\eta}}^{(l-1)} \approx \mathbf{D}_l^{-1}(\hat{\boldsymbol{\mu}}^{(l)} - \hat{\boldsymbol{\mu}}^{(l-1)}),$$

and

$$\mathbf{D}_l^{-1}(\hat{\boldsymbol{\mu}}^{(l)} - \hat{\boldsymbol{\mu}}^{(l-1)}) \approx \mathbf{B}_{\hat{\gamma}^{(l)}}(\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l\mathbf{B}_{\hat{\gamma}^{(l)}} + \lambda\boldsymbol{\Lambda})^{-1}\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l{\mathbf{D}_l}^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l)}).$$

Multiplication with $\mathbf{W}_l^{1/2}$ and using $\mathbf{W}_l^{1/2}\mathbf{D}_l^{-1} = \mathbf{\Sigma}_l^{-1/2}$ yields

$$\mathbf{\Sigma}_l^{-1/2}(\hat{\boldsymbol{\mu}}^{(l)} - \hat{\boldsymbol{\mu}}^{(l-1)}) \approx \tilde{\mathbf{H}}_l \mathbf{\Sigma}_l^{-1/2}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}),$$

where $\tilde{\mathbf{H}}_l = \mathbf{W}_l^{1/2}\mathbf{B}_{\hat{\gamma}^{(l)}}(\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l\mathbf{B}_{\hat{\gamma}^{(l)}} + \lambda\mathbf{\Lambda})^{-1}\mathbf{B}'_{\hat{\gamma}^{(l)}}\mathbf{W}_l^{1/2}$ denotes the usual generalized ridge regression hat-matrix. Defining $\mathbf{M}_l = \mathbf{\Sigma}_l^{1/2}\tilde{\mathbf{H}}_l\mathbf{\Sigma}_l^{-1/2}$ yields the approximation

$$\begin{aligned}
\hat{\boldsymbol{\mu}}^{(l)} &\approx \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{M}_l(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-1)}) \\
&= \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{M}_l[(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-2)}) - (\hat{\boldsymbol{\mu}}^{(l-1)} - \hat{\boldsymbol{\mu}}^{(l-2)})] \\
&\approx \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{M}_l[(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-2)}) - \mathbf{M}_{l-1}(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-2)})] \\
&= \hat{\boldsymbol{\mu}}^{(l-1)} + \mathbf{M}_l(\mathbf{I} - \mathbf{M}_{l-1})(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(l-2)}).
\end{aligned}$$

With starting value $\hat{\boldsymbol{\mu}}^{(0)} = \mathbf{M}_0\mathbf{y}$, $\mathbf{M}_0 = \frac{1}{n}\mathbf{1}_n\mathbf{1}'_n$, one obtains

$$\begin{aligned}
\hat{\boldsymbol{\mu}}^{(1)} &\approx \hat{\boldsymbol{\mu}}^{(0)} + \mathbf{M}_1(\mathbf{y} - \hat{\boldsymbol{\mu}}^{(0)}) \\
&= \mathbf{M}_0\mathbf{y} + \mathbf{M}_1(\mathbf{I} - \mathbf{M}_0)\mathbf{y},
\end{aligned}$$

and further, in a recursive manner,

$$\hat{\boldsymbol{\mu}}^{(l)} \approx \mathbf{H}_l\mathbf{y},$$

where

$$\mathbf{H}_l = \mathbf{I} - (\mathbf{I} - \mathbf{M}_0)(\mathbf{I} - \mathbf{M}_1)\cdots(\mathbf{I} - \mathbf{M}_l) = \sum_{j=0}^{l}\mathbf{M}_j\prod_{i=0}^{j-1}(\mathbf{I} - \mathbf{M}_i).$$