Statistics Surveys
Vol. 13 (2019) 1–51
ISSN: 1935-7516
https://doi.org/10.1214/19-SS124

Additive monotone regression in high and lower dimensions

Solveig Engebretsen[*,†]

Oslo Centre for Biostatistics and Epidemiology
Department of Biostatistics
Institute of Basic Medical Sciences
University of Oslo
Postbox 1122 Blindern
0317 Oslo, Norway
Department of Infectious Disease Epidemiology and Modelling
Division for Infection Control and Environmental Health
Norwegian Institute of Public Health
Oslo, Norway
e-mail: solveig.engebretsen@medisin.uio.no

and

Ingrid K. Glad[†]

Department of Mathematics
University of Oslo
Postbox 1053 Blindern
0316 Oslo, Norway
e-mail: glad@math.uio.no

Abstract: In numerous problems where the aim is to estimate the effect of a predictor variable on a response, one can assume a monotone relationship. For example, dose-effect models in medicine are of this type. In a multiple regression setting, additive monotone regression models assume that each predictor has a monotone effect on the response. In this paper, we present an overview and comparison of very recent frequentist methods for fitting additive monotone regression models. Three of the methods we present can be used both in the high dimensional setting, where the number of parameters p exceeds the number of observations n, and in the classical multiple setting where $1 < p \leq n$. However, many of the most recent methods only apply to the classical setting. The methods are compared through simulation experiments in terms of efficiency, prediction error and variable selection properties in both settings, and they are applied to the Boston housing data. We conclude with some recommendations on when the various methods perform best.

MSC 2010 subject classifications: Primary 62G08.
Keywords and phrases: Monotone regression, shape constrained regression, regression splines, additive regression.

Received November 2018.

*Corresponding author.
†The authors acknowledge partial funding from the Norwegian Research Council centre Big Insight project 237718.

Contents

1. Introduction

The linear model is a simple model with strong restrictions. A model which is a lot more flexible is the general additive model [22], which assumes that the effect of each covariate is a general univariate function. Let p denote the number of

covariates, and n the number of observations. The general additive model with identity link is then

$$Y_i = \beta_0 + \sum_{j=1}^{p} g_j(x_{ij}) + \epsilon_i \quad (i = 1, \ldots, n), \tag{1.1}$$

where the g_js are unknown smooth functions to be estimated and ϵ_i are independent and identically distributed mean-zero normal random variables. A natural approach is to fit the functions g_j by splines, so that each g_j is a linear combination of spline basis functions. This brings us back to a linear problem, which there are methods for solving.

In many applications, and especially in the life sciences, effects are however often naturally subject to some shape restrictions, in particular monotonicity. In such situations, the g_js in (1.1) are assumed to be smooth and monotone functions. It is important to have methods which incorporate this restriction into the model estimation.

In this paper, methods for additive monotone regression in both high ($p > n$) and low ($1 < p \leq n$) dimensions are presented. Tutz and Leitenstorfer [44] write that "It is surprising that most of the literature on monotonic regression focuses on the case of unidimensional covariate x and metrically scaled, continuous response variable y". There has been some development since then, and we here give an overview and comparison of the available frequentist methods in the multidimensional setting, most of them developed very recently. We will especially consider the methods developed in [11, 39, 45, 44, 25]. These are all methods developed for the classical regression setting, but the method in [25] can also be used in the high dimensional case. We will also include two methods specifically designed for the high dimensional data setting. These two methods are the liso regression method [18] and the monotone splines lasso regression method [5]. Even though these are meant for $p > n$ situations, they might be applied also in the classical setting. We thus include these as possibilities also for $p \leq n$, but keep in mind that these methods per definition automatically perform variable selection. The methods developed for the classical setting require the monotonicity directions of the functions. However, such information is not always available. The high dimensional monotone regression methods can be used without prior information on the monotonicity direction, and can therefore potentially be a valuable resource also in the low dimensional setting.

The paper is organised as follows: in Section 2, a short review of methods for additive monotone regression for $p \leq n$ and $p > n$, respectively, is presented. In Section 3, a qualitative overview and comparison of the various methods is given. In Section 4, the methods are compared through simulation experiments in different classical settings. In Section 5, the $p > n$ methods are compared in a high dimensional setting. In Section 6, the robustness of the methods to violation of the monotonicity assumptions is studied and in Section 7, the various approaches are applied to the Boston housing data, which is a classical data set with house values and different explanatory variables. In Section 8, we have some additional remarks on monotone regression hypersurfaces, existing

Bayesian methods for monotone regression and comment on methods for the partially linear monotone model. In Section 9, we summarise our results with concluding remarks and recommendations.

2. Monotone regression methods

Often, the relationship between some explanatory variable and a response is monotonically increasing or decreasing. For example, it is common to assume that the relationship between some measure of cognitive performance of children and age is a monotonically increasing function, and it is not plausible that this relationship is linear [6]. In medicine, we often have monotone relationships between two variables, for example between the amount of exercise and serum cholesterol level [43]. It is often assumed that genetic effects on phenotypes are monotone, like in [34].

As mentioned, many of the methods developed for monotone regression are developed for the univariate case. For instance, [4] uses isotonic step functions to fit regression models in the one dimensional setting. He and Shi [23] present a method for univariate monotone regression using monotone B-spline smoothing and Meyer [36] develops a method for shape-restricted regression splines using I-splines, for the one-dimensional case. An alternative to splines is the wavelet based (univariate) monotone estimator [1].

It is more challenging, but of course more relevant, to consider multiple regression, as we rarely have only one predictor variable. Ramsay [40] develops a method for monotone regression using I-splines, which can also be used in the multivariate setting. Bachetti [2] develops a method for additive isotonic regression. In [2], each function is fitted by an isotonic step function, and the method is based on an iterative cyclic optimisation scheme, starting with an initial guess for all the functions. The functions are updated cyclically one by one, keeping the other functions at their currently best guess, and minimising the loss with respect to the current function, by a unidimensional isotonic regression method. This is repeated until convergence is obtained. Dette and Scheder [14] develop a method for monotone regression where the regression function is a monotone hypersurface of the covariates. However, in this paper, we focus on the (less general, but more widely studied,) *additive* monotone regression models. Tutz and Leitenstorfer [44] use the ideas of [40] in combination with monotone boosting, with optional monotonicity constraints on the functions. A very similar method is developed in [28], using monotone boosting and B-splines with constraints. Pya and Wood [39] use P-splines to fit a regression model where some of the functions are fitted by functions with shape constraints, and the rest have no shape constraint. Chen and Samworth [11] also develop a method for regression with different shape constraints on the functions. The method is based on using different basis functions with different constraints on the parameters, depending on what shape restriction is imposed. A similar method is developed in [37]. Hofner and others [25] combine boosting with P-splines. Wang and Xue [45] generalise the method in [23] to the multidimensional setting, where B-spline

smoothing is used to fit the monotone regression model. All these methods are developed for lower dimensional regression. When it comes to methods developed for the high dimensional ($p > n$) setting, to our knowledge, there are only three available methods, namely the liso regression method [18], the monotone splines lasso [5] and the monotone boosting method (mboost) developed in [25].

In this paper, we consider non-Bayesian methods for monotone regression. However, it should be noted that there exist also various Bayesian methods for multiple monotone regression models. We provide a short overview of Bayesian methods for monotone regression in Section 8.2.

We will focus mostly on the methods known as scar, CPS, scam, MonBoost, mboost, liso and monotone splines lasso, which are introduced in more detail in the following sections. Some of the methods we consider focus only on monotonically increasing functions. However, if g_j is assumed to be monotonically decreasing, the same method/algorithm can be used, but with reversed sign on the observed covariates.

2.1. Monotone regression methods for the $p \leq n$ setting

2.1.1. Scar – Chen and Samworth (2016)

The method developed in [11] estimates the model in equation (1.1), where each function g_j is assumed to satisfy one out of nine possible shape constraints. It is assumed that it is a priori known which shape constraint each function satisfies. Among these nine are monotonically increasing and monotonically decreasing constraints. All the functions are assumed to have zero mean, for unique identification.

To fit the monotone functions, step basis functions are used. Let $\mathbf{X}$ be the design matrix of the observations, and let $x_{(i)j}$, $i = 1, \ldots, n$, be the corresponding order statistics for each covariate j. The basis functions are given as

$$s_{ij}(x) = \begin{cases} I(x_{(i)j} \leq x) - I(x_{(i)j} \leq 0), & \text{if } g_j \text{ is monotonically increasing,} \\ I(x \leq x_{(i)j}) - I(0 < x_{(i)j}), & \text{if } g_j \text{ is monotonically decreasing,} \end{cases}$$

where I is the indicator function. We refer to [11] for the spline basis functions used for the other shape constraints. The spline approximation is given as $\tilde{g}_j(x_j) = \sum_{i=1}^{n} \beta_{ij} s_{ij}(x_{ij})$, where x_{ij}, $i = 1, \ldots, n$, are the observations of covariate j. The basis coefficients β_{ij} are all restricted to be positive. The solution is given by the (positive) $\boldsymbol{\beta}$ minimising

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left\| \mathbf{y} - \sum_{j=1}^{p} \sum_{i=1}^{n} \beta_{ij} s_{ij}(\mathbf{x}_j) \right\|_2^2,$$

where $\mathbf{y}$ is the observed response. To solve the optimisation problem, an active set algorithm is used. The detailed algorithm is given in [11]. In the active set algorithm, one basis function is added in the active set at the time, namely

the one maximising the local derivative of the likelihood, that is, the one with the largest slope. In every iteration, the $\boldsymbol{\beta}$ maximising the likelihood is found by iterative reweighted least squares. If there are any negative elements of $\boldsymbol{\beta}$, a single basis function is dropped from the active set (the minimiser of a specific non-linear drop function), and the iterative reweighted least squares solution for the updated set is found. This is repeated until all elements are positive. The algorithm stops when all the local derivatives of the likelihood are smaller than a threshold (smaller than or equal to zero for the Gaussian case, smaller than a small threshold for other exponential family distributions). Convergence of the algorithm is guaranteed in the Gaussian setting. The authors also state that they did not encounter convergence issues in the non-Gaussian setting [11]. The method was found to be uniformly consistent on compact intervals, under mild conditions.

The method is implemented in the R-package *scar*, and we will refer to it as scar. It should be noted that this method is not restricted to the setting with a normal distribution for the $\mathbf{y}$, but can be used also for other exponential family distributions.

2.1.2. Constrained polynomial splines – Wang and Xue (2015)

The method developed in [45] is a generalisation of the method developed by [23], to the multidimensional setting. This method is thus based on B-spline smoothing and fits the model given in (1.1), where the functions are assumed to be monotone and have zero mean. The monotonicity directions have to be known a priori with this method. The authors use a two-stage approach where they first fit the g_j functions using B-splines with no constraints on the parameters, so that the estimated functions are general smooth functions. Then these estimated functions are used in a one-step constrained backfitting approach. Let B_k denote the B-spline basis functions and γ_{jk} denote the corresponding basis coefficients. The spline approximations are then given by $\tilde{g}_j(x) = \sum_{k=1}^{m} \gamma_{jk} B_k(x)$, where m is the number of spline basis functions for each covariate. Let $\mathbf{Z}$ denote the matrix with the $\mathbf{x}$ observations represented in the B-spline basis and $\boldsymbol{\gamma}$ denote the corresponding vector of basis coefficients. The estimates of the basis coefficients are then given by

$$\hat{\boldsymbol{\gamma}} = \text{argmin}_{\boldsymbol{\gamma}} ||\mathbf{y} - \mathbf{Z}\boldsymbol{\gamma}||_2^2,$$

which can be solved by ordinary least squares (given a large enough n compared to p).

Let then $\mathbf{y}_{-j} = \mathbf{y} - \sum_{j' \neq j} \hat{\tilde{g}}_{j'}(\mathbf{x}_{j'})$, where $\hat{\tilde{g}}_j(\mathbf{x}_j) = \sum_{k=1}^{m} \hat{\gamma}_{jk} B_k(\mathbf{x}_j)$. The $\mathbf{y}_{-j}$ is then an approximation of g_j. A sufficient condition for $\tilde{g}_j$ to be monotonically increasing is that $\gamma_{jk} \geq \gamma_{jk-1}$. The estimated functions are then given by $\hat{g}_j(\mathbf{x}_j) = \sum_{k=1}^{m} \hat{\beta}_{jk} B_k(\mathbf{x}_j)$, where $\hat{\boldsymbol{\beta}}_j$ is given by

$$\hat{\boldsymbol{\beta}}_j = \text{argmin}_{\boldsymbol{\beta}_j \in C} ||\mathbf{y}_{-j} - \mathbf{Z}_j \boldsymbol{\beta}_j||_2^2,$$

where C is the set of vectors of length m satisfying the constraint $\beta_{jk} \geq \beta_{jk-1}$ and $\mathbf{Z}_j$ is the matrix with the observed values of covariate j, represented in the B-spline basis. This is a standard constrained optimisation problem, and can be solved with the R-function *constrOptim*.

The constrained fitting could have been done in a one-stage approach instead of using a two-stage approach, but in [45], they argue that fitting the model in two steps is numerically more stable. Hence, in the implementation of the method, we will use the two-stage approach. They also show asymptotical convergence and consistency of the method, under regularity conditions.

For the rest of the paper, we will refer to this method as CPS (constrained polynomial spline).

2.1.3. Scam – Pya and Wood (2015)

The method developed in [39] estimates the model given in equation (1.1), where the functions have different optional shape constraints. The shape constraints on the functions have to be known a priori to use this method. Among these constraints are monotonically increasing and monotonically decreasing functions. The model is fitted using P-splines. P-splines are B-splines with a difference penalty on adjacent B-spline coefficients. See [17] for more details on P-splines.

Consider first the one dimensional setting, where

$$Y = g(x) + \epsilon.$$

The function $g(x)$ is approximated by a B-spline. Let B_k denote the spline basis functions and γ_k denote the basis coefficients. Then we have

$$\tilde{g}(x) = \sum_{k=1}^{m} \gamma_k B_k(x),$$

where m is the number of basis functions, and $\tilde{g}$ is the spline approximation of g. As mentioned, a sufficient condition for the function $\tilde{g}$ to be monotonically increasing is that $\gamma_k \geq \gamma_{k-1}$. A reparametrisation is used, so that

$$\boldsymbol{\gamma} = \boldsymbol{\Sigma}\tilde{\boldsymbol{\beta}},$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)'$, $\tilde{\boldsymbol{\beta}} = (\beta_1, \exp(\beta_2), \ldots, \exp(\beta_m))'$ and $\boldsymbol{\Sigma}$ is such that $\Sigma_{ij} = 0$ if $i < j$ and $\Sigma_{ij} = 1$ if $i \geq j$. This reparametrisation ensures that the fitted function is monotonically increasing. Let $\mathbf{Z}$ denote the matrix with the $\mathbf{x}$ observations represented in the B-spline basis. Then we have

$$\tilde{g}(\mathbf{x}) = \mathbf{Z}\boldsymbol{\Sigma}\tilde{\boldsymbol{\beta}}.$$

The reparametrisations necessary for other shape constraints are listed in Table 1 in [39].

To control the wiggliness of $\tilde{g}(x)$, a penalty term is introduced, penalising the squared differences between adjacent β_k. The penalty is given as $||\mathbf{D}\boldsymbol{\beta}||_2^2$,

where $\mathbf{D}$ is such that all elements are zero, except from $D_{i,i+1} = -D_{i,i+2} = 1$ for $i = 1, \ldots, m-2$. Note that the penalty is on the $\boldsymbol{\beta}$ and not on the $\tilde{\boldsymbol{\beta}}$.

In the multidimensional setting, it is assumed that all the functions have zero mean, for unique identification of the functions. Let each shape constrained function be represented by a model matrix on the form $\tilde{g}_j(\mathbf{x}_j) = \mathbf{Z}_j\boldsymbol{\Sigma}_j\tilde{\boldsymbol{\beta}}_j = \boldsymbol{M}_j\tilde{\boldsymbol{\beta}}_j$, where $\mathbf{x}_j$ are the observed values of covariate j. Let $\mathbf{M}$ denote the matrix with all the $\mathbf{M}_j$ and $\tilde{\boldsymbol{\beta}}$ the vector with all the $\tilde{\boldsymbol{\beta}}_j$. If there are linear covariates in addition, the design matrix with the linear covariates and the linear parameters are also included in $\mathbf{M}$ and the parameter vector $\tilde{\boldsymbol{\beta}}$. There are no penalties on the linear parameters. In a similar manner, functions with no shape constraints can also be added to the model, given as B-spline approximations, so that we have a design matrix with the observations represented in the B-spline basis and a parameter vector for the covariates with no shape constraints. The penalty for the covariates with no shape constraints is given in [47]. The penalty term is on the form $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_\lambda\boldsymbol{\beta}$, where $\mathbf{S}_\lambda = \sum_{j=1}^{p}\lambda_j\mathbf{S}_j$ and $\mathbf{S}_j = \mathbf{D}_j^{\mathrm{T}}\mathbf{D}_j$. The parameters λ_j are smoothing parameters. Given the λ_j, the solution, $\hat{\boldsymbol{\beta}}$, is given as the minimiser of

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}}||\mathbf{y} - \mathbf{M}\tilde{\boldsymbol{\beta}}||_2^2 + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_\lambda\boldsymbol{\beta}. \tag{2.1}$$

This is solved by a Newton-Raphson scheme. The smoothing parameters λ_j are estimated by the AIC criterion or the generalised cross-validation (GCV). See [39] for the algorithm for solving the problem and details on the GCV. As for scar, the method is not restricted to the setting with a normal distribution for the $\mathbf{y}$, but can be used also for other exponential family distributions.

This scheme is implemented in the R-package *scam*, and we will refer to this method as scam.

2.1.4. MonBoost – Tutz and Leitenstorfer (2007)

The method developed in [44] is called the MonBoost method. MonBoost also estimates the model given in equation (1.1), where some of the functions g_j are restricted to being monotone.

Let g_j be approximated by a basis expansion, where the basis functions are I-spline basis functions. These are monotonically increasing basis functions. A sufficient condition for monotonicity is then that all the basis coefficients are of the same sign. Another option for MonBoost is sigmoidal basis functions. Since we only consider monotonically increasing functions, we seek a solution where all the basis coefficients are nonnegative. The basis expansion is given by

$$\tilde{g}_j(x) = \sum_{k=1}^{m}\beta_{jk}I_k^{(l)}(x),$$

where $I_k^{(l)}$ are the basis functions, l is the order of the basis functions, $\tilde{g}_j(x)$ is the approximation of g_j, and m is the number of basis splines used. We will use

I-spline basis functions of order two. In [40] and [5], a small number of knots is used. In MonBoost, many interior knots are used, and boosting is used to avoid overfitting. Since MonBoost is based on boosting, it has an in-built variable selection property.

The concept of boosting is to combine many weak learners (in classification, a weak learner is one that is only slightly better than random guessing), to obtain a good predictor. Componentwise boosting is used, so that each weak learner only changes the contribution of one basis spline. The more iterations, the closer will the model be fitted to the training data. Thus, we need a stopping criterion for determining when to stop. In [44], both AIC and the g-prior minimum description length (gMDL) are suggested. gMDL is a hybrid between AIC and BIC, see [44] or [10] for more details. It is also possible to regularise by using a shrinkage parameter which shrinks the learner for each iteration. In MonBoost, this is done by using a ridge regression estimate as the weak learner, with a quite large value of the ridge penalty parameter λ.

The estimated functions are constructed by ensuring that all the estimated parameters are positive, so the estimated function will necessarily be monotone. If there are no shape constraints on the function, we do not need to consider only the subset of positive estimated parameters in the algorithm. An R-implementation of the algorithm was made available by the authors [44], but has recently been removed. We base our implementation on this R-code. The algorithm in the one dimensional case with Gaussian response is provided in [44], and we restate it here in Algorithm 1 in Appendix B.

It should be noted that even though [44] only considers applications in the classical setting, the algorithm would also work when $p > n$. However, MonBoost has to be provided the monotonicity directions for every covariate a priori, and in a high dimensional setting, it can be challenging to have an intuition about the monotonicity directions for all the covariates.

Just as scam and scar, MonBoost is not restricted to having normal response, but can be used with any response from an exponential family.

2.1.5. Mboost – Hofner and others (2016)

The method developed in [25] is a combination of scam and MonBoost, and thus combines boosting with P-splines. It estimates the model given in equation (1.1), with optional monotonicity constraints on the g_j. It also supports other shape constraints, for instance linearity and periodicity.

The functions are estimated by spline approximations, where P-splines are used to estimate the monotone functions. As with MonBoost, many interior knots are used, and boosting is used to avoid overfitting. The method is based on componentwise boosting, hence it performs variable selection intrinsically, like MonBoost.

Since componentwise boosting is used, the contribution of one basis spline is changed in each boosting iteration. As with MonBoost, the more iterations, the closer will the model be fitted to the training data, and cross-validation is

used as a stopping criterion for the number of iterations. Similar to MonBoost, regularisation is also obtained by using a shrinkage parameter which shrinks the learner for each iteration, with a difference penalty as in equation (2.1) in the scam method, and an additional penalty term to ensure monotonicity. The additional monotonicity penalty term ensures that the differences in adjacent coefficients are either all positive or all negative, due to a fixed, high penalty for solutions which do not fulfil this. For other shape constraints than monotonicity, other penalty terms are used. See [25] for details on the penalty terms. The penalty parameter for the P-splines difference penalty is found by fixing the degrees of freedom to a low number. The default value for the degrees of freedom is four, and we will use this in our applications of the method.

In [25], they point out that the method can be used in the high dimensional setting, however the method is not tried out in this setting. Note that the monotonicity directions must be provided a priori (as with MonBoost), which can be difficult, especially in the high dimensional setting. It can for instance be reasonable to assume a monotone relationship between a response variable and gene expressions, however it is not straightforward to know a priori the monotonicity directions for all the 20 000 different genes.

This method is not restricted to gaussian response. It is not even restricted to exponential family distributions, see [26] for details on the possible families of distributions that are implemented. The method is implemented in the R-package *mboost*, and we will refer to it as mboost.

2.2. *Monotone regression methods specifically designed for the high dimensional $p > n$ setting*

2.2.1. Liso – Fang and Meinshausen (2012)

The most common method for modelling monotone relationships is to use isotonic regression, which produces step functions instead of smooth functions. For high dimensional data, there has been developed a method, lasso isotone (liso) [18], which combines isotonic regression with lasso. It is defined as the minimisation of the liso loss, L_λ, with respect to $(g_1, g_2, \ldots, g_p)$. The liso loss L_λ is given by

$$L_\lambda(\beta_0, g_1, \ldots, g_p) = \frac{1}{2}\left\|\mathbf{y} - \beta_0 - \sum_{j=1}^{p} g_j(\mathbf{X}^{(j)})\right\|_2^2 + \lambda \sum_{j=1}^{p} \Delta(g_j),$$

where $\mathbf{X}^{(j)}$ is the jth column of $\mathbf{X}$, and the g_js are bounded, univariate and monotonically increasing functions. The $\Delta(g_j)$ denotes the total variation in g_j

$$\Delta(g_j) = \sup_{x \in \mathbb{R}} g_j(x) - \inf_{x \in \mathbb{R}} g_j(x).$$

The residual error only considers the value of g_j at the observed points. Thus for optimality, the bounds for the estimated g_j should be at the extremal

observed value of the covariate. Outside the interval between the smallest and the largest observed value of the covariate, the function should be flat. Any interpolation function between the points minimising L_λ will be an optimal solution. Therefore, for simplicity, right-continuous step functions are used, with knots at the observation points. To perform the fitting of the liso method, we need to know a priori whether the covariates are monotonically increasing or monotonically decreasing. The liso method can be improved by an adaptive procedure. This improved method is called adaptive liso, and it can be used without prior knowledge about the monotonicity directions of the functions. The adaptive liso thus has the advantage over the lower dimensional methods for monotone regression that it does not need to be provided the monotonicity directions. Let $\hat{g}_j^{\text{init}}$ for $j = 1, \ldots, p$ be initial liso fits for the functions. Let then

$$w_j = \begin{cases} \infty, & \text{if } \Delta\hat{g}_j^{\text{init}} = 0, \\ \frac{1}{\Delta\hat{g}_j^{\text{init}}}, & \text{otherwise,} \end{cases}$$

for $j = 1, \ldots, p$. The adaptive liso fit is then given by

$$(\hat{g}_1, \hat{g}_2, \ldots, \hat{g}_p) = \text{argmin}_{(g_1, g_2, \ldots, g_p)} \frac{1}{2} \left\| \mathbf{y} - \beta_0 - \sum_{j=1}^{p} g_j(\mathbf{X}^{(j)}) \right\|_2^2 + \lambda \sum_{j=1}^{p} w_j \Delta(g_j).$$

A univariate liso solution is found by a thresholded version of the pool adjacent violator algorithm (PAVA) [3], which fits a univariate isotone step function. The thresholds depend on the regularisation parameter λ, which can be chosen by cross-validation. The thresholded PAVA algorithm is then extended to multiple dimensions by an iterative backfitting algorithm. The authors show that with this algorithm, the liso loss converges to its global minimum [18]. However, if there is no unique solution, the backfitting algorithm is not guaranteed to converge. For the adaptive liso under unknown monotonicity directions, the function is decomposed into a sum of a monotonically increasing function and a monotonically decreasing function, by including both the original covariate and the sign-opposite covariate as covariates in the liso fit. The estimated effect of the covariate is then the combination of the estimated monotonically increasing function, and the estimated monotonically decreasing function.

Even though the adaptive liso does not have to be provided the monotonicity directions, it does have the disadvantage of not guaranteeing a monotone fit, but it shrinks the estimated functions towards monotone functions. Both liso and adaptive liso perform automatic variable selection, as opposed to most of the classical methods. The resulting estimated functions are step functions.

2.2.2. Monotone splines lasso – Bergersen and others (2014)

The monotone splines lasso method [5] is a recently developed method for monotone regression in high dimensions. With this method, the fitted functions are

smooth, monotone functions. In applications, it is often more reasonable to assume that the true underlying function is smooth (rather than a step function as in liso). To apply the monotone splines lasso method, the monotonicity directions do not need to be known a priori. Consider again the model in equation (1.1), and assume that the functions g_j can be approximated by m I-spline basis functions of order l, so that

$$\tilde{g}_j(x) = \sum_{k=1}^{m} \beta_{jk} I_k^{(l)}(x),$$

where $I_k^{(l)}$ are the basis functions, β_{jk}, $k = 1, \ldots, m$, are the basis coefficients for covariate j in the spline basis and $\tilde{g}_j$ is a spline approximation of g_j. As mentioned, since the I-spline basis functions are monotonically increasing, $\tilde{g}_j$ will be monotone as long as for each j, all the coefficients β_{jk}, $k = 1, \ldots, m$, have the same sign. So β_{jk}, $k = 1, 2, \ldots, m$, are either all nonnegative, all nonpositive or all zero. We will use the I-spline basis functions of order two. As in [5], the I-spline basis functions are centred so that $\mathrm{E}[\tilde{g}_j(x)] = 0$, to ensure unique identification of the functions.

Let $\mathbf{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_p)$ be the $n \times pm$ design matrix with the covariates represented in the I-spline basis, where $\mathbf{Z}_j$ is the $n \times m$ design matrix for covariate j, represented in the I-spline basis. Let $\boldsymbol{\beta}$ be the corresponding vector of basis coefficients. Then consider the minimisation problem

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_{\mathrm{coop}},$$

where λ controls the regularisation as before, and a cooperative lasso penalty is used to ensure that the estimated coefficients for each covariate are sign-coherent.

The cooperative lasso [13] is an enhancement of the group lasso, which can be used to obtain sign-coherent parameter estimates within a group, when fitting a linear regression model. The cooperative lasso penalty is given by

$$||\boldsymbol{\beta}||_{\mathrm{coop}} = \sum_{j=1}^{k} ||\boldsymbol{\beta}_{\mathscr{G}_j}^{+}||_2 + ||\boldsymbol{\beta}_{\mathscr{G}_j}^{-}||_2,$$

where $\boldsymbol{\beta}_{\mathscr{G}_j}^{+} = \max(\boldsymbol{\beta}_{\mathscr{G}_j}, 0)$, $\boldsymbol{\beta}_{\mathscr{G}_j}^{-} = \max(-\boldsymbol{\beta}_{\mathscr{G}_j}, 0)$ and $\mathscr{G}_j$ denotes the group.

This penalisation scheme favours sign-coherent solutions, in the sense that it penalises more on sign-incoherent solutions. When the penalty parameter goes to zero, sign-coherence is no longer guaranteed [13]. So if the regularisation parameter is small, the solution might be sign-incoherent, resulting in a non-monotone fit.

In [5], λ is chosen by cross-validation. Since the cooperative penalty has the variable selection property, the monotone splines lasso method can perform variable selection. If the covariate j is selected by the method, all the parameters within one group will be nonnegative or nonpositive, provided that the penalty parameter is large enough.

The monotone splines lasso can also be improved by an adaptive procedure. This improved method is called adaptive monotone splines lasso. Let $\hat{\boldsymbol{\beta}}_j^{\text{init}}$ be the initial fit for the basis coefficients for covariate j, for $j = 1, \ldots, p$. The adaptive monotone splines lasso estimates are then given by

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}||_2^2 + \lambda \sum_{j=1}^{p} w_j (||\boldsymbol{\beta}_j^+||_2 + ||\boldsymbol{\beta}_j^-||_2),$$

where $\boldsymbol{\beta}_j$ are the m basis coefficients corresponding to covariate j, and

$$w_j = \begin{cases} \infty, & \text{if } ||\hat{\boldsymbol{\beta}}_j^{\text{init}}||_2 = 0, \\ \frac{1}{||\hat{\boldsymbol{\beta}}_j^{\text{init}}||_2}, & \text{otherwise.} \end{cases}$$

The authors show that, conditional on some assumptions, the monotone splines lasso estimator is consistent and has the property of exact support recovery, that is, the set of selected variables is correct with probability converging to one. See [13] or [5] for details on the assumptions.

The algorithm for solving the cooperative lasso problem is an active set algorithm combined with a Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton with box constraints, or proximal methods, for solving the cooperative lasso optimisation problem in each step. An R-implementation of monotone splines lasso is provided by [5], see http://www.mn.uio.no/math/english/people/aca/glad/r-scripts/mslasso/. It is based on the R-package *scoop* for cooperative lasso [13], see http://julien.cremeriefamily.info/scoop.

3. Qualitative overview and comparison of monotone methods

An overview of properties of the various methods is given in Table 1. The methods are compared through the type of basis functions used, which settings they can be applied in ($p \leq n$ and/or $p > n$), whether they have to be provided the monotonicity directions for the functions, whether they support other constraints than monotonicity, whether there exists an R-package with an implementation of the method, whether the methods require choices or parameter values that have to be specified a priori, and the default options, if any, for the specifications required. If there is an R-package for the method, we have marked whether the implementation also can handle a generalised response. The different specifications for the methods are the number of knots for the spline methods, penalisation parameters and stopping criterion for the boosting methods.

TABLE 1. *Overview of properties for the different methods. "Needs direction" means that the method has to be provided the monotonicity direction for each covariate. "Other shapes" means that the method can also be used to fit functions with other shapes than monotone. "R-package" is whether or not the method is implemented in an R-package, and "(+ Generalised)" means that the implementation has the possibility for other families than normal response. "Required specifications" are choices or parameter values that have to be specified a priori. "Default option/method" are the corresponding default parameter values or methods for determining the parameters. "MS-lasso" is the monotone splines lasso.*

Method	Basis functions	Dimensions	Needs direction	Other shapes	R-package	Required specifications	Default option/ method
Scar	Step functions	$1 \leq p \leq n$	Yes	Yes	Yes (+ Generalised)	None	–
CPS	B-splines	$1 \leq p \leq n$	Yes	No	No	Number of knots	None
Scam	P-splines	$1 \leq p \leq n$	Yes	Yes	Yes (+ Generalised)	Number of knots $\boldsymbol{\lambda}$	None GCV
MonBoost	I-splines Sigmoidal functions	$1 \leq p \leq n$ (and $p > n$)	Yes	Yes	No	Number of knots Stopping criterion, λ	20 AIC 20
Mboost	P-splines	$1 \leq p \leq n$ and $p > n$	Yes	Yes	Yes (+ Generalised)	Number of knots Stopping criterion deg. of freedom	20 10-fold CV 4
Adaptive liso	Step functions	$1 \leq p \leq n$, and $p > n$	No	No	Yes	λ	10-fold CV
MS-lasso	I-splines	$1 \leq p \leq n$, and $p > n$	No	No	No	Number of knots λ	None 10-fold CV

4. Numerical comparison of monotone methods when $1 < p \leq n$

We here study the performances of all the methods in the classical multiple setting, where there are less parameters than observations. This is done by performing simulation experiments with n observations and p parameters. The methods are compared by estimation performance, prediction performance and variable selection performance. Though prediction and estimation errors might often be the primary interests in $p \leq n$ regression settings, there are also many settings where variable selection is important, especially when the number of predictors is large [49] (which can of course also be the case when $p \leq n$). Parsimonious models are easier to interpret and provide better understanding of the relationship between the response and the explanatory variables [49, 24]. Hence, in most (but not all) of the settings we consider, there are some noise covariates. The simulation set up is similar to [31]. We draw random variables $\mathbf{v} = (v_1, v_2, \ldots, v_n), \mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{w} = (w_{11}, w_{12}, \ldots, w_{1n}, \ldots, w_{p1}, \ldots, w_{pn})$, where u_i, v_i and w_{ij} are drawn from a normal distribution with mean 0.5 and standard deviation 1, truncated to $[0, 1]$.

We let

$$x_{ij} = \frac{w_{ij} + tu_i}{1+t} \text{ for } j \in \mathscr{A} \ ,$$

and

$$x_{ij} = \frac{w_{ij} + tv_i}{1+t} \text{ for } j \notin \mathscr{A} \ ,$$

where $\mathscr{A}$ is the set of true covariates. We let the set of true covariates be x_1, x_2, x_3 and x_4. The dependence between the covariates is controlled by t, and with $t = 0$, the covariates are independent. The covariance between two variables is then 0 if they are not in the same set, and $t^2/(1+t^2)$ if they are in the same set. We will for simplicity only consider $t = 0$ and $t = 1$, that is, independent covariates, and covariates with a within-set covariance of 0.5. We let the true additive regression model be

$$y_i = g_1(x_{i1}) + g_2(x_{i2}) + g_3(x_{i3}) + g_4(x_{i4}) + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$, and σ is chosen to control the signal-to-noise ratio (SNR), where SNR is the ratio between the standard deviation of the signal and σ. The functions are given as

$$g_1(x) = -\exp(x^2),$$

$$g_2(x) = -\log(x + 0.1),$$

$$g_3(x) = 2\tanh(20x^2) + 0.5\exp(x^3),$$

and

$$g_4(x) = \frac{2\exp(10x-5)}{1+\exp(10x-5)},$$

as in [5]. The functions are centred for the assumption of zero mean.

The methods that are compared are monotone splines lasso, adaptive monotone splines lasso, adaptive liso, scam, scar, CPS, MonBoost, mboost and classical linear regression using ordinary least squares. We do not use the liso method for comparison, since it needs prior knowledge about the monotonicity directions of the functions, while adaptive liso does not. Scam, scar, CPS, MonBoost and mboost also need to be provided the monotonicity directions, but since there are no alternative versions of these methods which do not need this, these methods will still be included in the comparison. In the classical linear regression setting, linear functions are fitted. For scam, MonBoost and mboost, the noise covariates are fitted without any monotonicity assumption. Scar has no option of no constraint, so we fit the noise covariates with a linear function. CPS has to be provided with monotonicity directions for all the covariates, so we fit the noise covariates by monotonically increasing functions. It should be kept in mind that the results for these rely on additional (and correct) information about the direction of the active variables, as opposed to the monotone splines lasso and adaptive liso.

To estimate the optimal penalisation parameter for monotone splines lasso and adaptive liso, a 10-fold cross-validation scheme is used. The smoothing parameters for scam are chosen by the default GCV option in the implementation. For MonBoost, AIC is used as a stopping criterion and the default value $\lambda = 20$ is used as a penalty parameter for the ridge estimate. For mboost, 10-fold cross-validation is used as a stopping criterion for the boosting iterations. We use B-splines of order three for CPS.

In addition, we have to specify the number of knots to use for the spline methods. MonBoost has an automatic, data-driven selection of basis functions and hence also the number of knots [44]. We use the default value of the maximum number of knots, $m = 20$. The same is true for mboost, and we also use the default value of the maximum number of knots, $m = 20$. For monotone splines lasso, scam and CPS, we select the number of knots as the minimiser of a prediction performance measure, as in [41]. For monotone splines lasso, we use 10-fold cross-validation, for CPS we use leave-one-out cross-validation and for scam we use the GCV. The reason why we use 10-fold cross-validation for monotone splines lasso and GCV for scam is because these are the default, implemented options. For CPS, we use leave-one-out cross-validation, since it is dependent on having a large n compared to p.

We simulate 500 times and a different design matrix is drawn in each simulation, to cover more situations and to give a fair comparison.

4.1. Estimation performance

We explore situations with large noise (SNR ≈ 2), with less noise (SNR ≈ 4), dependent covariates ($t = 1$) and independent covariates ($t = 0$). The specific settings we consider are given in Table 2. The true model is the same in all settings, except case 6, where one of the functions is replaced by a non-monotone function, to investigate robustness to the monotonicity assumptions. Cases 0a-c are the only cases with no noise covariates. The number of true covariates

selected, the number of false covariates selected and the mean squared estimation errors from the estimated functions to the true functions in the observed points are recorded for comparisons.

TABLE 2
The different simulation settings for the comparisons of the monotone regression methods.

Name	n	p	Noise covariates	SNR	t	Description
Case 0a	80	4	0	4	0	Strong signal, independent covariates, no noise covariates
Case 0b	150	4	0	4	0	As case 0a, more observations
Case 0c	50	4	0	4	0	As case 0a, fewer observations
Case 1a	80	7	3	4	0	Strong signal, independent covariates
Case 1b	150	7	3	4	0	As case 1a, more observations
Case 1c	150	7	3	4	0	As case 1a, fewer observations
Case 2a	80	7	3	4	1	Strong signal, dependent covariates
Case 2b	150	7	3	4	1	As case 2a, more observations
Case 3	80	7	3	2	0	Weak signal, independent covariates
Case 4	200	20	16	4	0	Many noise covariates
Case 5	50	1000	996	4	0	High dimensional setting
Case 6	80	7	3	4	0	Non-monotone setting

4.1.1. Case 0: The ideal case

In the first setting, we have a strong signal, independent covariates and no noise covariates. The mean squared estimation errors for the different methods are given in Table 3. Considering the estimation errors, we find that the scam method performs the best for all four functions. MonBoost and mboost perform second best and adaptive liso performs fourth best. CPS performs the worst. We also include, for completeness, the results of variable selection where appropriate. The methods which perform automatic variable selection do not have problems with selecting all four variables, as they should. Prediction results are also available in Table 3, but will be commented on separately in Section 4.2 on prediction performance for all the cases 0-4.

We investigate how sensitive the results are to the number of observations, by increasing the number of observations to $n = 150$ (case 0b) and decreasing the number of observations to $n = 50$ (case 0c). The relative ranking was quite robust to the number of observations, though as expected, all methods perform worse with less information (cf. Table A2 in Appendix A), and better with more information (cf. Table A1 in Appendix A). As before, scam performs best in estimation and CPS performs worst, for both settings. There were minor variations in the relative rankings of the other methods. With $n = 150$, all the true covariates are selected for all the methods which perform variable selection. When $n = 50$, the adaptive monotone splines lasso performs worse in selecting all the true covariates.

TABLE 3

Case 0a. Average number of total true positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.1, where $n = 80$, $p = 4$, $SNR \approx 4$ and $t = 0$. Standard deviations are given in parenthesis. Monotone splines lasso selected 1 interior knot, scam selected 13 interior knots and CPS selected 9 interior knots. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection		Mean squared prediction error			
	TP				
Lin. mod	–	0.38 (0.043)			
MS-lasso	4.0 (0)	0.28 (0.047)			
Ad. MS-lasso	3.99 (0.10)	0.27 (0.052)			
Ad. liso	4.0 (0)	0.21 (0.050)			
Results with correct information on monotonicity direction provided for each covariate:					
Scam	–	0.16 (0.046)			
Scar	–	0.37 (0.73)			
CPS	–	0.60 (0.28)			
MonBoost	4.0 (0)	0.19 (0.044)			
Mboost	4.0 (0)	0.19 (0.037)			
Mean squared estimation error					
		g_1	g_2	g_3	g_4
Lin. mod		0.038 (0.012)	0.037 (0.013)	0.12 (0.019)	0.041 (0.011)
MS-lasso		0.024 (0.015)	0.023 (0.014)	0.046 (0.016)	0.052 (0.016)
Ad. MS-lasso		0.034 (0.025)	0.020 (0.016)	0.031 (0.014)	0.048 (0.018)
Ad. liso		0.015 (0.0073)	0.018 (0.0076)	0.019 (0.0066)	0.016 (0.0067)
Results with correct information on monotonicity direction provided for each covariate:					
Scam		0.0059 (0.0049)	0.0048 (0.0045)	0.0079 (0.0057)	0.0070 (0.0055)
Scar		0.018 (0.0082)	0.020 (0.0084)	0.022 (0.0091)	0.020 (0.0087)
CPS		0.063 (0.053)	0.084 (0.079)	0.13 (0.14)	0.16 (0.15)
MonBoost		0.012 (0.0060)	0.014 (0.0063)	0.015 (0.0068)	0.014 (0.0069)
Mboost		0.010 (0.0072)	0.011 (0.0077)	0.019 (0.0081)	0.015 (0.0088)

4.1.2. Case 1: Easy case

In the remaining situations, we will include noise covariates. We first consider the situation with a strong signal and independent covariates. The number of true positives, the number of false positives and the estimation errors for the estimated functions for case 1a are given in Table 4. Since the scar, scam, CPS and linear regression method do not perform variable selection, these methods are not compared through variable selection properties, but we compare them through mean squared estimation errors of the fitted functions. The 500 fitted functions for g_2 with all the different methods are given in Fig 1 and the 500 fitted functions for g_3 are given in Fig 2. A box plot of the mean squared estimation error for g_2 and g_3 is provided in Fig 3, corresponding to Table 4.

We see from Table 4 that all the methods perform well in selection of the true

covariates. The adaptive monotone splines lasso method outperforms the other methods in false covariates. Hence in variable selection, the adaptive monotone splines method seems to perform the best. Adaptive liso performed second best in terms of selection. The boosting based methods, MonBoost and mboost, select the most false covariates. Note that even though MonBoost and mboost perform variable selection, boosting methods are developed for good predictions and not necessarily capturing the true underlying effects of each covariate. There is no additional penalty for including a new covariate instead of changing one that is already included in the boosting algorithm. MonBoost and mboost thus do not directly penalise the number of covariates included in the model, and including small contributions of false covariates is not costly for the methods.

Considering the estimation error in terms of mean squared error, we find that the estimated functions with scam are a lot closer to the true functions than the estimated functions with any of the other methods. Second best are mboost and MonBoost, which perform equally well. Adaptive liso performs fourth best. As before, CPS has the largest estimation errors. We see from Figs 1 and 2 that all the monotone regression methods are good at recovering the true shapes of g_2 and g_3, except the CPS method, which clearly performs the worst among the monotone methods. We also observe that the estimated functions with scam are the most accurate. This can also be seen in Fig 3.

We investigate how sensitive the results are to the number of observations, by increasing the number of observations to $n = 150$ (case 1b) and decreasing the number of observations to $n = 50$ (case 1c). The estimation errors for $n = 150$ are smaller than the estimation errors for $n = 80$ (cf. Table A3 in Appendix A), and larger for $n = 50$ (cf. Table A2 in Appendix A). The relative ranking was again quite robust to the number of observations. Scam performs best in estimation and CPS performs the worst. There are minor variations in the relative ranking for the other methods. For $n = 150$, all the methods are good at selecting the true covariates, and adaptive monotone splines lasso selects fewest false covariates. For $n = 50$, adaptive monotone splines lasso performs worse in selection of true covariates, and hence adaptive liso performs the best in selection. MonBoost and mboost select the most false covariates.

4.1.3. Cases 2-4: Difficult cases

Case 2. Strong signal, dependent covariates The results for case 2a are given in Table A5 in Appendix A. When it comes to variable selection when we have SNR ≈ 4 and dependent covariates ($t = 1$), adaptive liso seems to be the best method. It selects all the true covariates, and has few false covariates. Adaptive monotone splines lasso selects no false covariates, but it has problems selecting all the true covariates. Monotone splines lasso performed better than adaptive monotone splines lasso, in that it was better at selecting the true covariates, while still selecting relatively few false covariates. Again, MonBoost selects the most false covariates, but as noted before, it is not really penalised for including more covariates in the same way as the other methods. Mboost

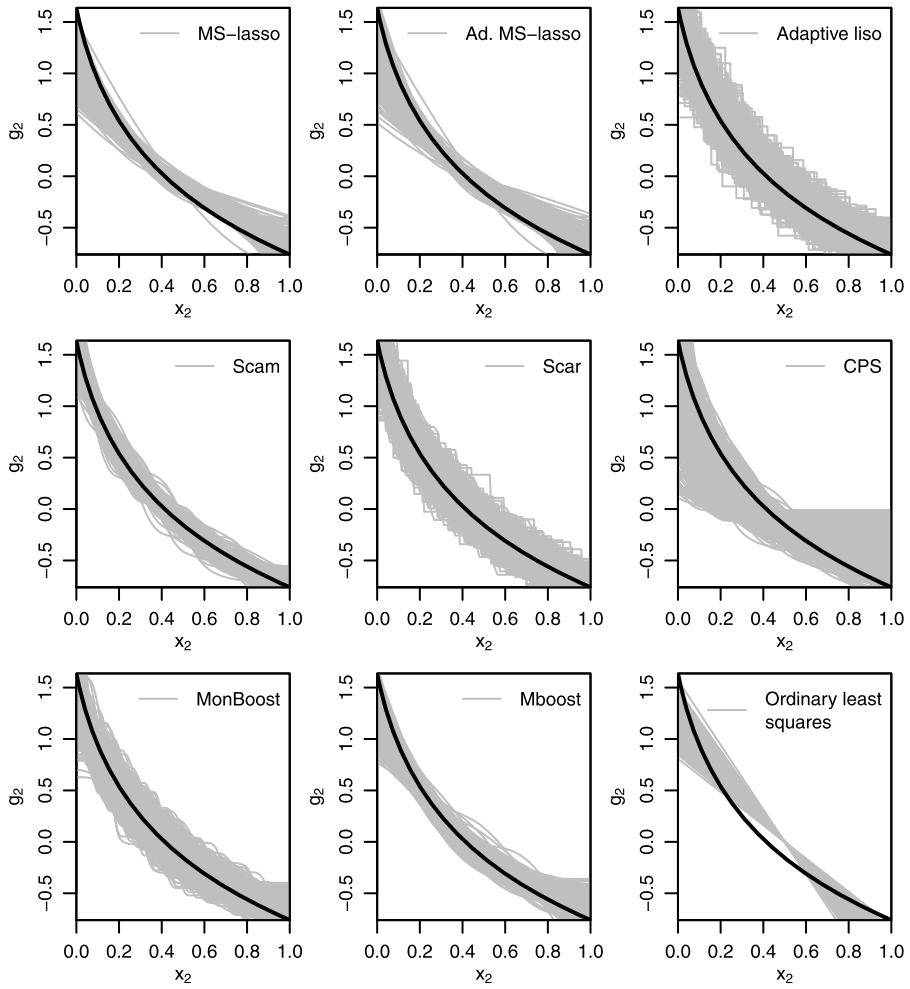

FIG 1. *Case 1a. Estimated functions for g_2 in the simulation considered in Section 4.1.2 with $n = 80$, $p = 7$, SNR ≈ 4 and $t = 0$, for all the different methods. The true function is given in black. Monotone splines lasso selected 1 interior knot, scam selected 12 interior knots and CPS selected 3 interior knots. "MS-lasso" is the monotone splines lasso and "Ad. MS-lasso" is the adaptive monotone splines lasso.*

also selects quite many false covariates, but performs better than MonBoost. When it comes to estimation error, scam outperforms all the other methods, MonBoost performs second best, while scar performs third best. CPS has the largest estimation errors. Note that the estimation errors are actually smaller here than in the setting with independent covariates, for all methods except mboost.

If n is increased to 150 (case 2b), we get the results in Table A6 in Appendix A. All the methods manage to capture the true model well here, except

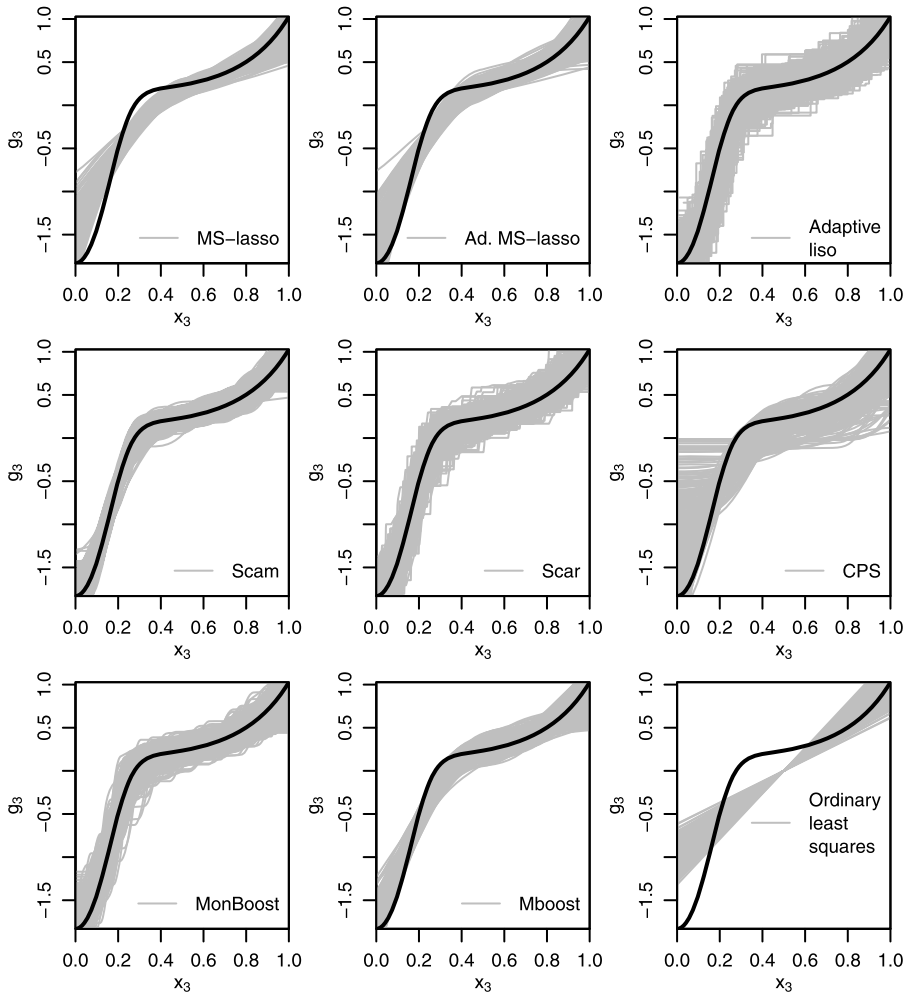

FIG 2. *Case 1a. Estimated functions for g_3 in the simulation considered in Section 4.1.2 with $n = 80$, $p = 7$, SNR ≈ 4 and $t = 0$, for all the different methods. The true function is given in black. Monotone splines lasso selected 1 interior knot, scam selected 12 interior knots and CPS selected 3 interior knots. "MS-lasso" is the monotone splines lasso and "Ad. MS-lasso" is the adaptive monotone splines lasso.*

from MonBoost, which selects too many false variables. Mboost does not have a problem with false covariates in this setting. Adaptive monotone splines lasso selects no false covariates, and adaptive liso also selects almost no false covariates. The relative estimation performance of the methods was quite robust to increased sample size.

Case 3. Weak signal, independent covariates In Table A7 in Appendix A, we see that when we have SNR ≈ 2 and $t = 0$, all the methods are good at

TABLE 4

Case 1a. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions in the simulation considered in Section 4.1.2, where $n = 80$, $p = 7$, SNR ≈ 4 and $t = 0$. Standard deviations are given in parenthesis. Monotone splines lasso selected 1 interior knot, scam selected 12 interior knots and CPS selected 3 interior knots. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection			Mean squared prediction error
	TP	FP	
Lin. mod	–	–	0.38 (0.045)
MS-lasso	4.0 (0)	0.15 (0.40)	0.28 (0.047)
Ad. MS-lasso	3.99 (0.1)	0.004 (0.089)	0.26 (0.048)
Ad. liso	4.0 (0)	0.098 (0.32)	0.21 (0.046)
Results with correct information on monotonicity direction provided for each covariate:			
Scam	–	–	0.17 (0.047)
Scar	–	–	0.31 (0.29)
CPS	–	–	0.46 (0.16)
MonBoost	4.0 (0)	2.97 (0.27)	0.21 (0.046)
Mboost	4.0 (0)	1.24 (0.80)	0.19 (0.038)

Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.038 (0.012)	0.038 (0.013)	0.12 (0.019)	0.040 (0.011)
MS-lasso	0.025 (0.016)	0.023 (0.014)	0.046 (0.016)	0.051 (0.016)
Ad. MS-lasso	0.034 (0.024)	0.021 (0.016)	0.031 (0.014)	0.047 (0.017)
Ad. liso	0.015 (0.0065)	0.018 (0.0070)	0.019 (0.0069)	0.016 (0.0065)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.0069 (0.0061)	0.0048 (0.0045)	0.0083 (0.0051)	0.0071 (0.0056)
Scar	0.019 (0.0090)	0.021 (0.0083)	0.023 (0.0087)	0.021 (0.0090)
CPS	0.045 (0.049)	0.081 (0.080)	0.074 (0.085)	0.12 (0.13)
MonBoost	0.013 (0.0063)	0.015 (0.0071)	0.016 (0.0081)	0.014 (0.0071)
Mboost	0.011 (0.0077)	0.012 (0.0080)	0.020 (0.0089)	0.016 (0.0091)

selecting the true covariates. Adaptive monotone splines lasso outperforms the other methods when it comes to false covariates (but also selected slightly fewer true covariates). Again, mboost and MonBoost select the most false covariates and a lot more than the other methods. When considering estimation error, we find again that scam outperforms all the other methods in estimation error, while mboost performs second best and MonBoost third best. CPS has the largest estimation error among all the methods.

Case 4. Many noise covariates We also consider a setting with more noise covariates, so $n = 200$, $p = 20$, SNR ≈ 4 and $t = 0$. The results are given in Table A8 in Appendix A. In this setting, MonBoost, mboost and monotone splines lasso have problems with false covariates. Adaptive liso performed the best in terms of selection. In terms of estimation error, scam again performs the

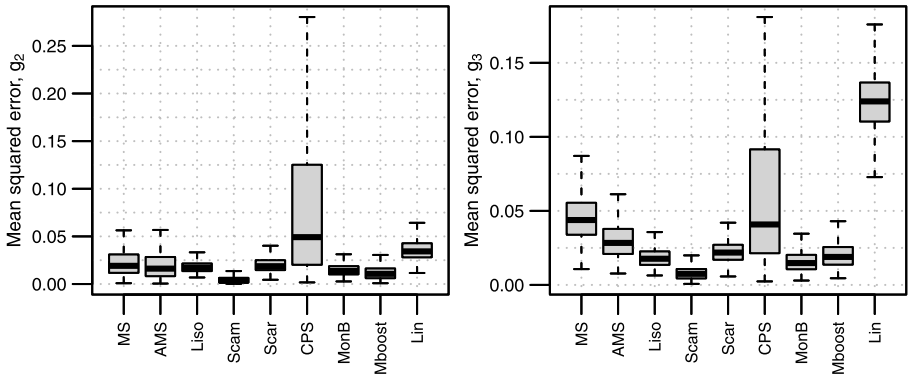

FIG 3. *Case 1a. Mean squared estimation error for g_2 and g_3, in the 500 simulations considered in Section 4.1.2 with $n = 80$, $p = 7$, SNR ≈ 4 and $t = 0$, for all the different methods. "MS" is the monotone splines lasso, "AMS" is the adaptive monotone splines lasso, "MonB" is the MonBoost method and "Lin" is the linear model.*

best, with MonBoost performing second best and adaptive monotone splines lasso third best. Adaptive liso performs better than monotone splines lasso, mboost, scar, the linear method and CPS.

4.2. Prediction performance

To compare the methods, their prediction performances are also studied. This is done by generating 500 new observations, $(\mathbf{X}^{\text{new}}, \mathbf{y}^{\text{new}})$, from the same distribution as the training data, and estimating the prediction error, PE, as

$$\text{PE} = \frac{1}{500}\sum_{i=1}^{500}(y_i^{\text{new}} - \hat{y}_i^{\text{new}})^2,$$

where $\hat{y}_i^{\text{new}}$ are the predicted values of y_i^{new}, using the fitted models. We draw 500 such sets of size 500, and estimate the mean prediction error over all the sets, refitting the model in each replication. The true underlying model is the same as before.

The prediction errors for all the settings are given in Tables 3, A1, A2, 4, A3, A4, A5, A6, A7 and A8. Box plots with the prediction errors for the different methods are given in Fig 4, for the case 1a (easy case) setting SNR ≈ 4, $t = 0$, $n = 80$ and $p = 7$. We see from Tables 3, A1, A2, 4, A4, A5, A6 and Fig 4 that in the settings with SNR ≈ 4, $t = 0$, $n = 80$ and $p = 4$ (case 0a), SNR ≈ 4, $t = 0$, $n = 150$ and $p = 4$ (case 0b), SNR ≈ 4, $t = 0$, $n = 50$ and $p = 4$ (case 0c), SNR ≈ 4, $t = 0$, $n = 80$ and $p = 7$ (case 1a), SNR $= 4$, $t = 0$, $p = 7$ and $n = 150$ (case 1b), SNR ≈ 4, $t = 0$, $n = 50$ and $p = 7$ (case 1c), SNR ≈ 4, $t = 1$, $n = 80$ and $p = 7$ (case 2a) and SNR ≈ 4, $t = 1$, $n = 150$ and $p = 7$ (case 2b), scam is best at prediction for all cases. In most of these settings, it is closely followed by mboost and MonBoost. Adaptive liso seems to

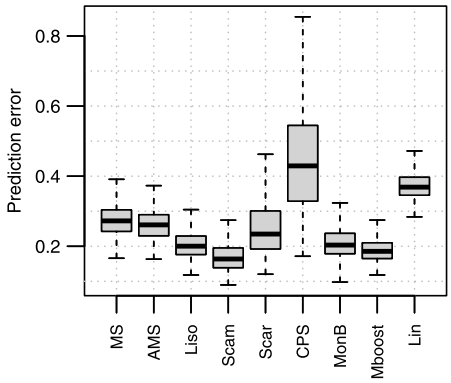

FIG 4. *Case 1a. Mean squared prediction error in the 500 simulations considered in Section 4.1.2 with $n = 80$, $p = 7$, SNR ≈ 4 and $t = 0$, for all the different methods. "MS" is the monotone splines lasso, "AMS" is the adaptive monotone splines lasso, "MonB" is the MonBoost method and "Lin" is the linear model.*

overall perform fourth best. Monotone splines lasso and its adaptive version also perform overall better than scar, the linear method and CPS. CPS performs the worst among the methods. In the setting with more noise given in Table A7, so SNR ≈ 2, $t = 0$, $n = 80$ and $p = 7$ (case 3), mboost performs best in terms of prediction, scam performs second best, while adaptive monotone splines lasso performs third best in this setting. Adaptive liso also outperforms MonBoost here. Scar performs the worst, and CPS performs second worst. From Table A8, we see that when there are more noise covariates (case 4), the prediction error is smallest for adaptive monotone splines lasso, closely followed by mboost, monotone splines lasso and adaptive liso. MonBoost and scam only have slightly larger prediction errors. So the prediction error is slightly smaller with adaptive monotone splines lasso when there are many noise covariates, but when there are few noise covariates, scam performs the best.

5. Case 5: The high dimensional case

The only methods available for additive monotone regression when $p > n$ are, as mentioned, liso, monotone splines lasso and mboost. An extensive comparison of liso and monotone splines lasso when $p > n$ is given in [5]. However, since the mboost method is more recent, it was not included in that comparison. We therefore perform a simulation study with a similar set-up as before (the easy case), with $n = 50$, $p = 1000$, SNR ≈ 4, $t = 0$. The true underlying model is the same as before, so we have four true covariates and 996 noise covariates.

The results in this setting are given in Table 5. We see that in selection, monotone splines lasso selects the most true positives, followed by adaptive monotone splines lasso. Adaptive liso selects fewest true positives. In false positives, adaptive monotone splines lasso outperforms all the other methods, with adaptive liso performing second best. Mboost and monotone splines lasso select a lot more

TABLE 5

Case 5. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 5, where $n = 50$, $p = 1000$, $SNR \approx 4$ and $t = 0$. Standard deviations are given in parenthesis. The number of interior knots selected by the monotone splines lasso method was one. "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection			Mean squared prediction error	
	TP	FP		
MS-lasso	3.83 (0.51)	15.88 (10.38)	0.65 (0.24)	
Ad. MS-lasso	3.65 (0.63)	3.55 (6.18)	0.49 (0.25)	
Ad. liso	3.07 (0.95)	5.21 (3.51)	0.79 (0.43)	
Results with correct information on monotonicity direction provided for each covariate:				
Mboost	3.30 (1.06)	15.24 (20.70)	0.87 (0.42)	
Mean squared estimation error				
	g_1	g_2	g_3	g_4
MS-lasso	0.090 (0.061)	0.094 (0.073)	0.13 (0.079)	0.13 (0.083)
Ad. MS-lasso	0.085 (0.074)	0.064 (0.077)	0.068 (0.064)	0.075 (0.070)
Ad. liso	0.13 (0.085)	0.13 (0.12)	0.14 (0.14)	0.091 (0.12)
Results with correct information on monotonicity direction provided for each covariate:				
Mboost	0.14 (0.076)	0.17 (0.12)	0.20 (0.15)	0.19 (0.16)

false covariates. The method performing best in selection is adaptive monotone splines lasso. Considering the mean squared errors of the estimated functions, adaptive monotone splines lasso performs the best, and mboost the worst. When considering prediction errors, adaptive monotone splines lasso also performs the best, while mboost performs worst. Hence, the mboost method performs worse than the other high dimensional methods, and we refer to the thorough comparison of monotone splines lasso and adaptive liso in [5] for performance of the methods in other settings, where it is concluded that the adaptive monotone splines lasso performs the best in most high dimensional settings.

6. Case 6: Robustness to monotonicity assumptions

To test the robustness of the methods to violation of the monotonicity assumptions, we perform a simulation where the simulation set-up and model are the same as in the previous settings, but g_4 is replaced by a non-monotonous function g_{4b}, such that,

$$y_i = g_1(x_{i1}) + g_2(x_{i2}) + g_3(x_{i3}) + g_{4b}(x_{i4}) + \epsilon_i,$$

where g_1, g_2 and g_3 are the same as before, and

$$g_{4b}(x) = 10(x - 0.5)^2.$$

For scam, scar, MonBoost, mboost and CPS, we assume a positive monotonicity direction for g_{4b}. The results are given in Table A9 in Appendix A. All the methods perform well in selecting the true covariates. As before, MonBoost and mboost are inferior in the selection of false covariates. Adaptive monotone splines lasso selects no false covariates, and adaptive liso and monotone splines lasso also select few false covariates, however more than in the setting with only monotone effects (case 1a).

Considering estimation error for the three monotone functions, adaptive liso performs the best, mboost performs second best and CPS performs third best. The linear method has the largest estimation errors and second worst is scar. Considering the estimation error for g_{4b}, adaptive liso again performs best, followed by adaptive monotone splines lasso and monotone splines lasso. These three methods are clearly best at estimating g_{4b}, and hence most robust to violation of the monotonicity assumption. This is also seen in Fig A1 in Appendix A, where the estimated functions for all the methods are given. The estimated functions with monotone splines lasso, adaptive monotone splines lasso and adaptive liso are not monotone, while the other methods have fitted monotone functions. Hence, the fact that these methods do not guarantee a monotone fit can also be beneficial, since it makes them more robust to violation of the monotonicity assumptions.

Comparing the prediction errors in this setting, we find that adaptive liso and (adaptive) monotone splines lasso clearly perform much better than the other methods when the monotonicity assumption is violated. Adaptive liso had the smallest prediction error.

7. Boston housing data ($p < n$)

7.1. Data description

We now try out the different methods for additive monotone regression using the well known Boston housing data set. The data consists of a response variable which is the house value, and different explanatory variables. This classical data set is from [21] and is available in the R-library *MASS*. It consists of $n = 506$ observations, 13 covariates and the house value (response). We will consider the explanatory variables crime (crime rate by town), zn (proportion of a town's residential land zoned into lots greater than 25 000 square feet), indus (proportion of non-retail business acres per town, serves as a measure of amount of industry), NOX (a pollution variable representing air quality as the concentration of nitrogen oxides), the mean number of rooms, age (the proportion of owner units built prior to 1940), distance to employment centres, rad (index of accessibility to radial highways), tax (the cost of public services), pupil-teacher ratio and the proportion of the population that is considered as lower status. We will not consider the covariate which is the proportion of the population being black, since this covariate is expected to have a parabolic effect [21] (and we focus only on monotone effects). In addition, an indicator

variable for whether or not it is a riverside location is given in the data set, but we will only consider numerical covariates. We will thus consider eleven predictors.

The methods that are used are monotone splines lasso, adaptive monotone splines lasso, adaptive liso, scam, scar, CPS, MonBoost and mboost. The data set was also used with adaptive liso in [18].

7.2. Monotonicity directions and parameter choices

For scam, scar, CPS, MonBoost and mboost, we have to provide the monotonicity directions. In [21], they propose that crime should have a negative effect on the house value, zn should have a positive effect, indus should have a negative effect, the mean number of rooms should have a positive effect, the distance should have a negative effect, rad should have a positive effect, tax should have a negative effect and the pupil-teacher ratio should have a negative effect. In addition, we assume that NOX and the proportion of the population having lower status will have a negative effect on the house value. For age, we do not know the monotonicity direction (it may well be that the relationship is not even monotone), so with scam, MonBoost and mboost we do not use any shape constraint on age, while we set a linear shape constraint on age with scar, and we assume a decreasing effect for CPS (since the univariate correlation between age and house value is negative, and we have to make a choice). CPS would not estimate the effect of rad, because there were not enough unique observations. Rad is thus left out as a covariate for the CPS method.

The optimal tuning parameters for monotone splines lasso, adaptive monotone splines lasso and adaptive liso are estimated by 10-fold cross-validation. The smoothing parameter for scam is chosen by the default GCV option. For monotone splines lasso, scam and CPS, the number of knots minimising the 10-fold cross-validation, GCV and leave-one-out cross-validation is selected, respectively. This results in 14 interior knots for monotone splines lasso, 22 interior knots for scam and 3 interior knots for CPS. For MonBoost, I-splines of order two with 20 knots are used to estimate the functions, $\lambda = 20$ is used as a penalty parameter for the ridge estimate and AIC is used as a stopping criterion (all default options). For mboost, 20 knots are used to estimate the functions and 10-fold cross-validation is used as a stopping criterion (default options).

7.3. Results

The estimated effects of the different covariates are given in Figs 5 and 6. All the variables are centred. The estimated effects of crime and NOX with adaptive liso are not monotone, which is also noted for crime in [18].

The estimated effects of crime, the mean number of rooms, the distance, tax and the proportion of lower status clearly deviate from linear functions, while the pupil-teacher ratio seems to be well approximated by a linear function.

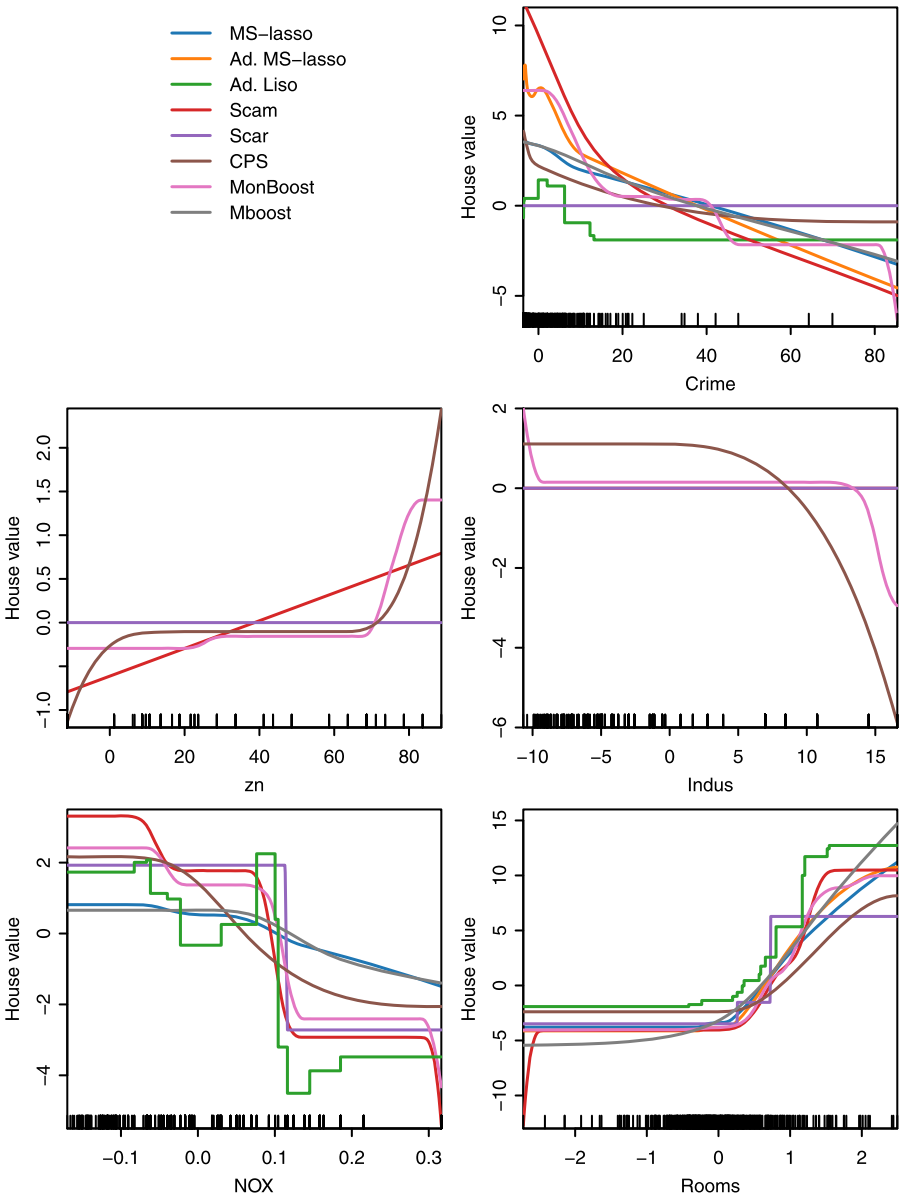

FIG 5. *Estimated functions for the Boston housing data, considered in Section 7. The observed values of the covariates are given at the bottom of the plots. "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.*

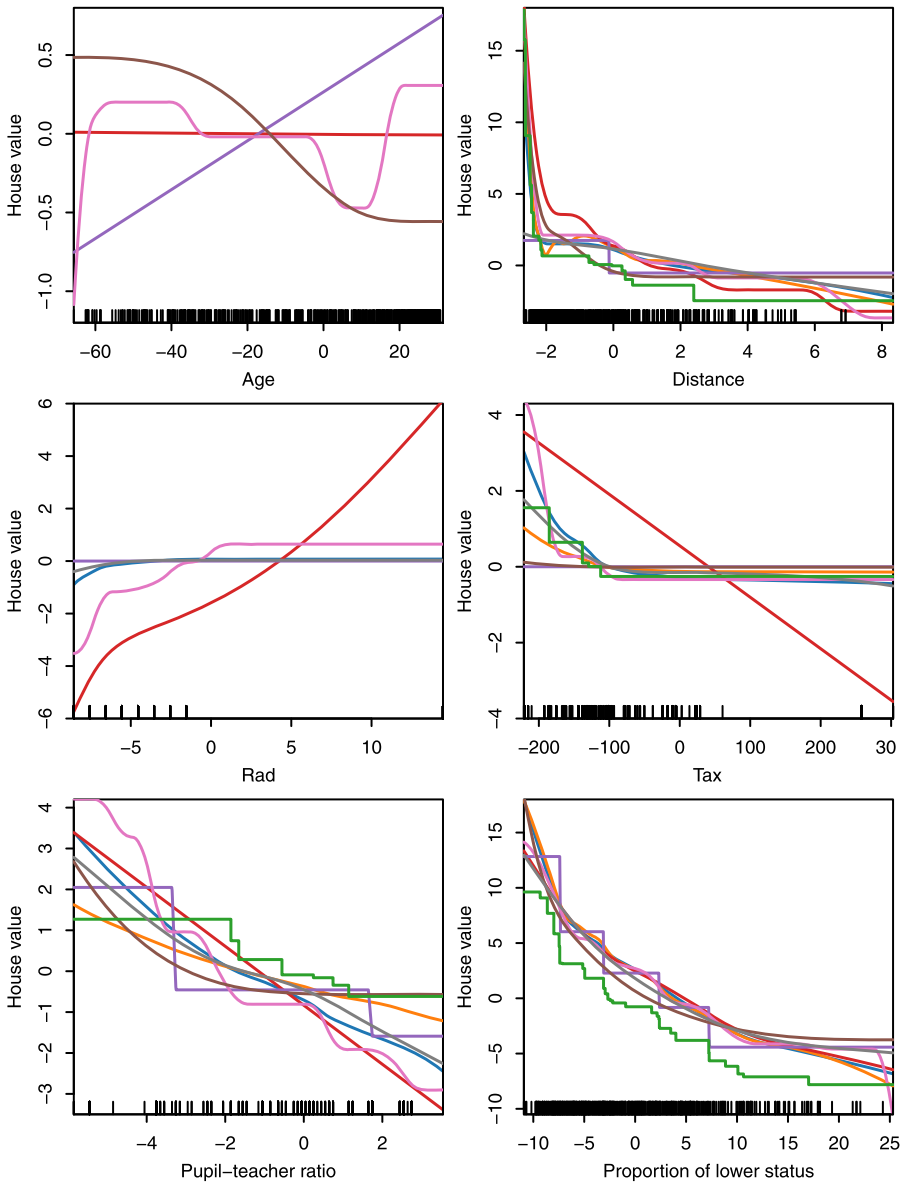

FIG 6. *Estimated functions for the Boston housing data, considered in Section 7. The observed values of the covariates are given at the bottom of the plots. "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.*

The estimated functions with the different methods are quite similar for the variables that were selected by a majority of the methods. The estimated effects of scam deviate most from the other methods, and scam seems to be more sensitive to extreme/influential observations. This is especially prominent in the estimated effects of NOX, mean number of rooms, rad and tax. CPS and MonBoost also seem to be affected by influential observations, which can be seen in the estimated effects of for instance crime, indus, NOX and proportion of lower status for MonBoost and zn and indus for CPS.

We know from the previous simulation experiments that the adaptive monotone splines lasso is good at not selecting false covariates. We therefore trust that all covariates selected by the adaptive monotone splines lasso method are important covariates for explaining the house value. We thus believe that crime, the mean number of rooms, distance, tax, pupil-teacher ratio and proportion of lower status should be kept in the final model. Adaptive liso only selects one more variable than adaptive monotone splines lasso, NOX, which is selected by all the methods except adaptive monotone splines lasso. However, the estimated function with adaptive liso is very non-monotone and not very reasonable, so we do not believe that this covariate is important for predicting the house value. Mboost and monotone splines lasso selects one additional variable, which is rad. However, the estimated effects of rad with mboost and monotone splines lasso are very small. MonBoost selects all the covariates, which is in accordance with our simulation experiments, where it was found that MonBoost selected more false positives than the other methods.

7.4. Prediction performance

In order to assess prediction performance of the additive monotone regression methods on the Boston housing data, we randomly split the data into a test set and a training set, and use the model fitted on the training set to predict the housing values in the test set. We let the training set consist of two thirds of the data (339 observations) and the test set consist of the remaining one third (167 observations). The splitting into a training set and a test set is repeated 100 times, to evaluate different alternative subsets of the data, in order to avoid sensitivity to the particular split. For monotone splines lasso, scam and CPS, the number of knots minimising the 10-fold cross-validation, GCV and leave-one-out cross-validation, respectively, is selected for every training set partition. The prediction variables, assumed monotonicity directions and different parameter and model choices are the same as before. Monotone splines lasso selected 14.9 (3.6) knots on average, scam selected 21.5 (4.6) knots on average and CPS selected 2.2 (0.6) knots on average, where the standard deviations are given in parentheses.

The prediction errors for the various methods are given in Table 6. For scar, approximately half of the time the model fitting resulted in an error message, so we had to run more iterations in order to obtain 100 training/test set replicates. We also encountered some issues with scam, with some fits (nine out of the 100

repetitions) resulting in very large prediction errors, possibly due to convergence issues. In these situations, we refitted the model by restricting the number of knots to be maximum 10, solving the problem of extreme predictions. Scam thus seems more robust and stable with fewer number of knots. MonBoost obtained the smallest prediction errors, scam performed second best, adaptive liso performed third best and monotone splines lasso performed fourth best. As in the simulation settings, scar and CPS performed the worst. Note however that the comparison is not completely fair, since we have removed the extreme results for scam. If we had included these, scam would have been judged to perform the worst among the methods, with a mean prediction error of 31.6 and a standard deviation of 84.9.

TABLE 6

Prediction errors in the Boston housing data example in Section 7. "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Method	Mean squared prediction error	Standard deviation
MS-lasso	17.4	4.5
Ad. MS-lasso	19.9	4.2
Ad. liso	16.1	3.7
Results with information on monotonicity direction provided for each covariate:		
Scam	14.8	3.3
Scar	20.5	4.7
CPS	26.4	6.2
MonBoost	14.0	3.3
Mboost	18.2	4.7

8. Additional remarks

8.1. Monotone regression hypersurfaces

A more general monotone regression model is the model

$$Y_i = m(x_{i1}, \ldots, x_{ip}) + \epsilon_i \quad (i = 1, \ldots, n),$$

where m is a smooth, monotone function of the p predictors. The function m is then a monotone hypersurface of the predictors. The additive monotone regression model considered in this study is a special case of this model. This model has been less studied than the additive monotone regression model, but some methods for fitting this more general, monotone model have been proposed. Dette and Scheder [14] develop a method for the general monotone regression model, where they first fit a non-parametric unconstrained estimate for m, and then use successive one-dimensional isotonisation procedures, resulting in a (strictly) monotone hypersurface of the predictors. An even more general method for fitting the general monotone regression model, which can also be

used for other shape constraints than monotonicity, is developed in [16], building on non-parametric kernel regression. There are also some Bayesian methods for the general monotone model, mentioned in the following section.

8.2. Bayesian methods for monotone regression

The monotone regression methods we have considered in this overview are all non-Bayesian methods. However, it should be noted that there exist also various Bayesian methods for multiple monotone regression models. A multivariate monotone regression method using Bayesian Additive Regression Trees is proposed in [12], where the relationship between one or more predictors and the outcome is assumed to be a monotone function of the predictor(s). A Bayesian monotone regression method for estimating monotone surfaces of predictors using Gaussian process projections is proposed in [30]. In [42], they use a Bayesian estimation procedure to fit a monotone regression model where the regression function is a general monotone function of the covariates. The function is estimated by a piecewise constant regression surface. These three Bayesian regression methods estimate monotone regression hypersurfaces, without any additivity assumption. Bornkamp and Ickstadt [7] develop a Bayesian method for univariate monotone regression, and the method is generalised to the multivariate setting in [8], for an additive monotone regression model. Meyer and others [38] propose a Bayesian approach for fitting partially linear models with restrictions on the non-linear covariates, for instance monotonicity. This is done by fitting shape-restricted splines. In [9], a method for additive monotone regression using Bayesian P-splines is proposed. More examples of Bayesian methods for monotone regression can be found in [42] and [38], but as also noted in [42], most of those methods seem to be developed for the univariate case.

8.3. Partially linear monotone models

In additive partially linear models, some covariates are assumed to have a linear effect on the response, and the rest are assumed to have a non-linear effect on the response. Let $\mathbf{y}$ be the observations of the response. Let $\mathbf{X}$ denote the design matrix for the covariates assumed to have a linear effect on the response, and $\mathbf{Z}$ the design matrix for the covariates assumed to have a non-linear effect on the response. Let d_1 be the number of covariates with linear effect, and d_2 the number of covariates with non-linear effect. Then the additive partially linear model is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \sum_{j=1}^{d_2} g_j(\mathbf{Z}^{(j)}) + \boldsymbol{\epsilon},$$

where $\mathbf{Z}^{(j)}$ is the jth column of $\mathbf{Z}$ and $g_1, g_2, \ldots, g_{d_2}$ are unknown, smooth functions. Again it is assumed that $\mathrm{E}[g_j(x)] = 0$ for all j, for unique identification.

In the $p < n$ setting, there are many methods for estimating partially linear models with no shape constraints on the non-linear effects, see for instance [27]

for an overview. There are also some existing methods which can be used to fit additive partially linear models in the high dimensional setting. Methods for fitting additive partially linear models with variable selection for the linear covariates are developed in [32] and [20]. In [15] and [35], methods for fitting the additive partially linear model with variable selection in both the linear and the non-linear covariates are proposed. A method for additive partially linear models with grouped linear covariates performing automatic variable selection in the linear covariates is proposed in [46].

In some situations, it might be reasonable to assume that the non-linear functions in an additive partially linear model are monotone. Out of the methods we have considered in this paper, scam, scar and mboost can be used to fit additive partially linear monotone models. As before, mboost performs automatic variable selection in both the linear and the non-linear covariates.

The ideas of monotone splines lasso can be extended to the partially linear monotone model. The method performs variable selection simultaneously in both the linear and the non-linear variables. I-splines of order l are used to represent the monotone functions, with m basis splines for each function, so g_j is approximated by

$$\tilde{g}_j(x) = \sum_{k=1}^{m} \gamma_{jk} I_k^{(l)}(x),$$

where γ_{jk} are the coefficients for covariate j in the spline basis and $\tilde{g}_j$ is a spline approximation of g_j. Let now $\mathbf{X}' = (\mathbf{X}, \mathbf{Z}')$ be the design matrix with the linear covariates as the first d_1 columns, and the monotone non-linear covariates represented in the I-spline basis in the last $d_2 \cdot m$ columns ($\mathbf{Z}'$). Let $\boldsymbol{\phi} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$, where $\boldsymbol{\beta}$ is the vector with linear parameters, and $\boldsymbol{\gamma}$ is the vector of basis coefficients. Let $\mathscr{G}_j$, $j = 1, \ldots, d_1, d_1 + 1, \ldots, d_1 + d_2$, denote the groups of the covariates. Then $\boldsymbol{\phi}_{\mathscr{G}_j} = \beta_j$ for $j = 1, \ldots, d_1$ and $\boldsymbol{\phi}_{\mathscr{G}_j} = (\gamma_{j1}, \ldots, \gamma_{jm})$ for $j = d_1 + 1, \ldots, d_1 + d_2$. Consider the problem

$$\hat{\boldsymbol{\phi}} = \text{argmin}_{\boldsymbol{\phi}} ||\mathbf{y} - \mathbf{X}'\boldsymbol{\phi}||_2^2 + \lambda ||\boldsymbol{\phi}||_{\text{coop}}.$$

Note that there is one, common penalty parameter, λ, for the linear and non-linear components. The penalty term is

$$||\boldsymbol{\phi}||_{\text{coop}} = \sum_{j=1}^{d_1+d_2} m_j ||\boldsymbol{\phi}_{\mathscr{G}_j}^+||_2 + m_j ||\boldsymbol{\phi}_{\mathscr{G}_j}^-||_2,$$

where, as in Section 2.2.2, $\boldsymbol{\phi}_{\mathscr{G}_j}^+ = \max(\boldsymbol{\phi}_{\mathscr{G}_j}, 0)$ and $\boldsymbol{\phi}_{\mathscr{G}_j}^- = \max(-\boldsymbol{\phi}_{\mathscr{G}_j}, 0)$. Weights on the penalty terms are now used, since the group sizes are not equal. The standard group lasso weights can be used, namely square roots of group sizes. Note that for the linear terms, the penalty reduces to an L_1 penalty (as in lasso). As before, λ is a tuning parameter controlling the regularisation, and can be chosen by for instance cross-validation. It follows by properties of the cooperative lasso that, under appropriate assumptions, the estimated parameters are consistent, and the method will, with probability converging to one, select the true model. See [13] for details on the assumptions.

In the extension of the monotone splines lasso to the partially linear setting, and for scam, scar and mboost, we assume that we know a priori which covariates have a linear effect on the response, and which covariates have a monotone non-linear effect. However, as noted in [29], such knowledge is rarely available, especially in a high dimensional setting. In [48], a method for separating the covariates into covariates with linear effects and general non-linear effects is developed, with a focus on the $p < n$ case. Methods which separate the covariates into linear and non-linear effects which can be used for the high dimensional case are proposed in [29] and [33]. An idea for future work is thus to combine this with a monotone shape restriction on the non-linear functions.

This extension of the monotone splines lasso method to the partially linear monotone model and mboost are then, to our knowledge, the only two methods which can be used in the $p > n$ case, to fit a partially linear monotone model. Since genetic effects on phenotypes are often assumed to be monotone (as in [34]), one application for the model is the setting where the predictors are both gene expressions and clinical covariates. The gene expressions could enter the non-linear, monotone component, and the clinical covariates could enter the linear component. Note that mboost would have to be provided the monotonicity directions for the different genes a priori.

9. Discussion and recommendations

We conclude that among the methods developed for the classical regression setting ($p < n$), scam performed the overall best in our simulation experiments. Scam had the smallest estimation errors, and the smallest prediction errors (but mboost and MonBoost only performed slightly worse in prediction). However, in the Boston housing data example, we found that scam seemed to be sensitive to influential observations, and there were also some tendencies for this with MonBoost and CPS. In the Boston housing data application, even though scam performed well in prediction on most of the training/test splits, scam also sometimes resulted in extremely large prediction errors.

Among the methods developed for the high dimensional data setting, the adaptive monotone splines lasso performed the best in selection in the classical setting. Adaptive liso outperformed monotone splines lasso in estimation. However, the (adaptive) monotone splines lasso method does have the advantage over adaptive liso that the estimated functions are smooth. MonBoost and mboost performed worse than adaptive liso and (adaptive) monotone splines lasso in variable selection, selecting too many false covariates.

We found that scam outperformed all the other methods in estimation, and also performed well in prediction. It is not surprising that scam outperforms the adaptive liso and monotone splines lasso, since scam is designed for the $p < n$ setting (and, in addition, it is provided with the monotonicity directions for the functions). Since MonBoost and mboost are designed for good predictions, it is also not surprising that the estimation errors are smaller for scam than for the boosting methods. It is, however, slightly surprising that the boosting methods

do not outperform scam in prediction. The reason why scam performs better than scar and CPS might be due to the additional smoothing penalty term for scam, which is not present for scar and CPS.

We have seen that scam outperforms the adaptive monotone splines lasso and the adaptive liso in the classical low dimensional setting, given the true monotonicity directions of each variable. Monotone splines lasso and adaptive liso have the advantage of performing automatic variable selection, which scam does not do. They also have the advantage that they do not need to be provided the monotonicity directions of the functions, as opposed to scam. This also makes the comparisons unfair for monotone splines lasso and adaptive liso, since scam is provided with more information about the functions. However, scam and all the other methods except CPS, monotone splines lasso and liso, are able to fit models where some of the functions are monotone, while the rest can have other or no shape constraints, making them more flexible. We also tried providing scam with the wrong monotonicity direction for x_1 (for the case 1a and the case 2a settings). The estimation errors for the other functions were much larger when scam was provided with wrong information, but still smaller than the estimation errors for monotone splines lasso. The prediction errors were larger (results not shown here). Scam outperformed the adaptive monotone splines lasso and adaptive liso in estimation error in all the simulation experiments, except the non-monotone setting (case 6). Monotone splines lasso and adaptive liso have the disadvantage that they do not guarantee a monotonic fit, while the estimated functions with scam will always be monotone. This does however make the methods more robust to violation of the monotonicity assumption. In the Boston housing data example, we found that adaptive monotone splines lasso and adaptive liso are more robust to influential observations than scam. When considering prediction error, scam performed better than adaptive monotone splines lasso and adaptive liso when we did not have many noise covariates, but when we increased the number of noise covariates, (adaptive) monotone splines lasso (and adaptive liso) performed slightly better than scam.

So even though the monotone splines lasso and liso methods did not outperform the classical methods in the lower dimensional setting, they are still a contribution also to the tools for performing analysis when $p < n$. To our knowledge, there are no other methods for monotone regression than monotone splines lasso and adaptive liso which properly perform variable selection. The mboost and the MonBoost methods do have the variable selection property, but as we have seen, they often include contributions from false covariates. Though prediction error and estimation error might often be the primary interests in classical regression settings, there are many situations where variable selection is also of importance and parsimony is often desirable. The simpler the model, the easier it is to interpret and understand the relationship between the response and the explanatory variables [49, 24]. Parsimony is especially important when the number of predictors is large (for instance p in the order of 100 and n in the order of 1000) [49]. In addition, measuring variables can be both time and resource demanding, and variable selection can be used to help inform clinicians about which variables they have to collect [19]. If a variable can be omitted, it

should therefore not be included. Hence, when variable selection is of interest, we recommend using adaptive monotone splines lasso.

Most of the methods considered require the monotonicity directions of the functions. However, we do not always have such information. Monotone splines lasso and adaptive liso can be applied without knowing the monotonicity directions a priori, as opposed to all the classical methods. This is particularly important for instance in settings with potential confounding effects, or when testing a hypothesis. If the aim of a study is to test the effect of a new treatment on an outcome, assuming a positive or negative effect is a source of bias. In addition, the treatment might interact with other (measured or unmeasured) covariates, resulting in uncertainty about the direction of the effect. Hence, when the direction of the effect is unknown, or one does not want to provide prior information about the direction, (adaptive) monotone splines lasso or adaptive liso should be used.

In the high dimensional setting, we found that adaptive monotone splines lasso performed the best in terms of selection, estimation and prediction in the setting we studied. Mboost was clearly inferior, and we thus refer to the thorough comparison of monotone splines lasso and adaptive liso in [5], where they conclude that adaptive monotone splines lasso is the best method for monotone regression in most high dimensional settings. Note however that mboost is more flexible, since it can fit other shape constraints than monotone functions. Hence, in settings where some of the variables are assumed to have a monotone effect and other variables are assumed to have other shapes, mboost might be a good alternative. Mboost does however need directions for the various monotone variables, which can be difficult in settings with many predictor variables.

Some of the spline based methods we have presented, require the number of knots to be specified a priori (e.g. scam, monotone splines lasso and CPS). In the comparisons in this paper, we selected the number of knots based on optimisation of a prediction error measure. This was done to make the comparison of the methods more fair. In practice, one might however want to choose the number of knots without using information in the data. A greater flexibility for the function is obtained the more knots used to fit it. However, the more data points there are to estimate each spline, the better will each spline estimate be [40]. In addition, if too many knots are used, the estimated functions might overfit the data. Both scam and monotone splines lasso include penalty terms, which make them less sensitive to the number of knots. In the case of I-splines, Ramsay [40] argues that it is more important to fit each curve well, since there is little to gain in having many knots if the function is poorly estimated between the knots. He also claims that in practice, there is often enough flexibility in the curve with just a single knot, and thus not many interior knots are needed. This is in accordance with monotone splines lasso often selecting one interior knot in our simulations. Meyer [36] argues that when there are shape constraints on the functions (such as monotonicity), then the restricted regression splines are robust to the number of knots.

References

[1] Antoniadis, A., Bigot, J., and Gijbels, I. (2007). Penalized wavelet monotone regression. *Statistics & Probability Letters*, 77(16):1608–1621. MR2393599

[2] Bacchetti, P. (1989). Additive isotonic models. *Journal of the American Statistical Association*, 84(405):289–294. MR0999691

[3] Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. (1972). Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression. *Hoboken, NJ: Wiley.* MR0326887

[4] Barlow, R. E. and Brunk, H. D. (1972). The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147. MR0314205

[5] Bergersen, L. C., Tharmaratnam, K., and Glad, I. K. (2014). Monotone splines lasso. *Computational Statistics & Data Analysis*, 77:336–351. MR3210067

[6] Bollaerts, K., Eilers, P. H., and Mechelen, I. (2006). Simple and multiple P-splines regression with shape constraints. *British Journal of Mathematical and Statistical Psychology*, 59(2):451–469. MR2282224

[7] Bornkamp, B. and Ickstadt, K. (2008). Bayesian nonparametric estimation of continuous monotone functions with applications to dose-response analysis. *Biometrics*, 65(1):198–205. MR2665861

[8] Bornkamp, B., Ickstadt, K., and Dunson, D. (2010). Stochastically ordered multiple regression. *Biostatistics*, 11(3):419–431.

[9] Brezger, A. and Steiner, W.J. (2008). Monotonic regression based on Bayesian P-splines: an application to estimating price response functions from store-level scanner data. *Journal of business & economic statistics*, 26(1):90 –104. MR2422064

[10] Bühlmann, P. and Yu, B. (2006). Sparse boosting. *The Journal of Machine Learning Research*, 7:1001–1024. MR2274395

[11] Chen, Y. and Samworth, R. J. (2016). Generalized additive and index models with shape constraints. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(4):729 – 754. MR3534348

[12] Chipman, H. A., George, E., I, McCulloch, R. E. and Shively, T. S. (2016). High-dimensional nonparametric monotone function estimation using BART. *arXiv preprint* arXiv:1612.01619. MR2758172

[13] Chiquet, J., Grandvalet, Y., and Charbonnier, C. (2012). Sparsity with sign-coherent groups of variables via the cooperative-lasso. *The Annals of Applied Statistics*, 6(2):795–830. MR2976492

[14] Dette, H. and Scheder, R. (2006). Strictly monotone and smooth nonparametric regression for two or more variables. *Canadian Journal of Statistics*, 34(4):535–561. MR2345035

[15] Du, P., Cheng, G. and Liang, H. (2012). Semiparametric regression models with additive nonparametric components and high dimensional parametric components. *Computational Statistics & Data Analysis*, 56(6):2006–2017. MR2892394

[16] Du, P., Parmeter, C. F. and Racine, J. S. (2013). Nonparametric kernel regression with multiple predictors and multiple shape constraints *Statistica Sinica*, 23(3):1347–1371. MR3114717

[17] Eilers, P. H. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11:89–102. MR1435485

[18] Fang, Z. and Meinshausen, N. (2012). Lasso isotone for high-dimensional additive isotonic regression. *Journal of Computational and Graphical Statistics*, 21(1):72–91. MR2913357

[19] Gunter, L., Zhu, J. and Murphy, S. (2007). Variable selection for optimal decision making. *Conference on Artificial Intelligence in Medicine in Europe*, 2007:149–154. Springer.

[20] Guo, J., Tang, M., Tian, M. and Zhu, K. (2013). Variable selection in high-dimensional partially linear additive models for composite quantile regression *Computational Statistics & Data Analysis*, 65:56–67. MR3064943

[21] Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102.

[22] Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1(3):297–310. MR0858512

[23] He, X. and Shi, P. (1998). Monotone B-spline smoothing. *Journal of the American statistical Association*, 93(442):643–650. MR1631345

[24] Hesterberg, T., Choi, N. H., Meier, L., and Fraley, C. (2008). Least angle and L_1 penalized regression: A review. *Statistics Surveys*, 2:61–93. MR2520981

[25] Hofner, B., Kneib, T., and Hothorn, T. (2016). A unified framework of constrained regression. *Statistics and Computing*, 26(1-2):1–14. MR3439355

[26] Hothorn, T., Buehlmann, P., Kneib, T., Schmid, M., and Hofner, B. (2017). *mboost: Model-Based Boosting*. R package version 2.8-1.

[27] Härdle, W. and Liang, H. (2007). Partially linear models. In *Statistical methods for biostatistics and related fields*, 87–103. Springer, Berlin, Heidelberg. MR2376405

[28] Leitenstorfer, F. and Tutz, G. (2006). Generalized monotonic regression based on B-splines with an application to air pollution data. *Biostatistics*, 8(3):654–673.

[29] Lian, H., Liang, H. and Ruppert, D. (2015). Separation of covariates into nonparametric and parametric parts in high-dimensional partially linear additive models. *Statistica Sinica*, 25(2):591–607. MR3379090

[30] Lin, L. and Dunson, D. B. (2014). Bayesian monotone regression using Gaussian process projection *Biometrika*, 101(2):303–317. MR3215349

[31] Lin, Y. and Zhang, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297. MR2291500

[32] Liu, X., Wang, L. and Liang, H. (2011). Estimation and variable selection for semiparametric additive partial linear models *Statistica Sinica*, 21(3):1225–1248. MR2827522

[33] Lou, Y., Bien, J., Caruana, R. and Gehrke, J. (2016). Sparse partially linear

additive models. *Journal of Computational and Graphical Statistics*,25(4): 1126–1140. MR3572032

[34] Luss, R., Rosset, S., and Shahar, M. (2012). Efficient regularized isotonic regression with application to gene–gene interaction search. *The Annals of Applied Statistics*, 6(1):253–283. MR2951537

[35] Lv, J., Yang, H. and Guo, C. (2017). Variable selection in partially linear additive models for modal regression *Communications in Statistics-Simulation and Computation* 46(7): 5646 – 5665. MR3698548

[36] Meyer, M. C. (2008). Inference using shape-restricted regression splines. *The Annals of Applied Statistics*, 2(3):1013–1033. MR2516802

[37] Meyer, M. C. (2013). Semi-parametric additive constrained regression. *Journal of Nonparametric Statistics*, 25(3):715–730. MR3174293

[38] Meyer, M. C., Hackstadt, A. J., and Hoeting, J. A. (2011). Bayesian estimation and inference for generalised partial linear models using shape-restricted splines. *Journal of Nonparametric Statistics*, 23(4):867–884. MR2854243

[39] Pya, N. and Wood, S. N. (2015). Shape constrained additive models. *Statistics and Computing*, 25(3):543–559. MR3334416

[40] Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science*, 3(4):425–441.

[41] Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics*, 11(4):735–757. MR1944261

[42] Saarela, O. and Arjas, E. (2011). A method for Bayesian monotonic multiple regression. *Scandinavian Journal of Statistics*, 38(3):499–513. MR2833843

[43] Schell, M. J. and Singh, B. (1997). The reduced monotonic regression method. *Journal of the American Statistical Association*, 92(437):128–135.

[44] Tutz, G. and Leitenstorfer, F. (2007). Generalized smooth monotonic regression in additive modeling. *Journal of Computational and Graphical Statistics*, 16(1):165–188. MR2345751

[45] Wang, L. and Xue, L. (2015). Constrained polynomial spline estimation of monotone additive models. *Journal of Statistical Planning and Inference*, 167:27–40. MR3383234

[46] Wei, F. (2012). Group selection in high-dimensional partially linear additive models *Brazilian Journal of Probability and Statistics*, 26(3): 219–243. MR2911703

[47] Wood, S. N. (2006). Generalized additive models: an introduction with R. *Chapman and Hall/CRC*. MR2206355

[48] Zhang, H. H., Cheng, G. and Liu, Y. (2011). Linear or nonlinear? Automatic structure discovery for partially linear models *Journal of the American Statistical Association*, 106(495):1099–1112. MR2894767

[49] Zou H, Hastie T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–20. MR2137327

Appendix A: Tables and figures

In this appendix, various tables from different simulation experiments are reported. Table A1 contains the results for the setting with no noise covariates and $n = 150$ (case 0b), while Table A2 contains the results for the setting with no noise covariates and $n = 50$ (case 0c), belonging to Section 4.1.1. Table A3 contains the results for the setting with strong signal and independent covariates with $n = 150$ (case 1b), while Table A4 contains the results for the setting with strong signal, independent covariates and $n = 50$ (case 1c), belonging to Section 4.1.2. Table A5 contains the results from the simulations with dependent covariates when $n = 80$ (case 2a). Table A6 contains the results from the simulations with dependent covariates when $n = 150$ (case 2b). Table A7 contains the results from the simulations with independent covariates and large noise (case 3). Table A8 contains the results from the simulation setting with many noise covariates (case 4). These tables belong to Section 4.1.3. Table A9 contains the results from the simulation setting with a non-monotone covariate effect from Section 6, and Fig A1 contains the estimated non-monotone function, g_{4b}, for all the methods.

TABLE A1

Case 0b. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.1, with $n = 150$, $p = 4$, $SNR \approx 4$ and $t = 0$. Standard deviations are given in parenthesis. Monotone splines lasso selected 1 interior knot, scam selected 13 and CPS selected 17. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection		Mean squared prediction error			
	TP				
Lin. mod	–	0.37 (0.032)			
MS-lasso	4.0 (0)	0.17 (0.028)			
Ad. MS-lasso	4.0 (0)	0.17 (0.028)			
Ad. liso	4.0 (0)	0.17 (0.028)			
Results with correct information on monotonicity direction provided for each covariate:					
Scam	–	0.14 (0.026)			
Scar	–	0.21 (0.19)			
CPS	–	0.47 (0.36)			
MonBoost	4.0 (0)	0.16 (0.026)			
Mboost	4.0 (0)	0.16 (0.023)			
Mean squared estimation error					
		g_1	g_2	g_3	g_4
Lin. mod		0.037 (0.0082)	0.037 (0.0086)	0.12 (0.013)	0.040 (0.0079)
MS-lasso		0.0066 (0.0053)	0.0061 (0.0041)	0.021 (0.0075)	0.024 (0.011)
Ad. MS-lasso		0.0081 (0.0073)	0.0055 (0.0041)	0.015 (0.0056)	0.025 (0.011)
Ad. liso		0.0087 (0.0032)	0.011 (0.0034)	0.011 (0.0038)	0.0094 (0.0032)
Results with correct information on monotonicity direction provided for each covariate:					
Scam		0.0031 (0.0029)	0.0024 (0.0021)	0.0042 (0.0024)	0.0036 (0.0024)
Scar		0.0099 (0.0037)	0.012 (0.0043)	0.013 (0.0042)	0.012 (0.0043)
CPS		0.041 (0.027)	0.057 (0.043)	0.097 (0.096)	0.12 (0.096)
MonBoost		0.0068 (0.0030)	0.0082 (0.0032)	0.0084 (0.0034)	0.0079 (0.0034)
Mboost		0.0069 (0.0045)	0.0076 (0.0045)	0.016 (0.0053)	0.011 (0.0055)

TABLE A2

Case 0c. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.1, with $n = 50$, $p = 4$, SNR ≈ 4 and $t = 0$. Standard deviations are given in parenthesis. Monotone splines lasso selected 1 interior knot, scam selected 12 and CPS selected 4. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection		Mean squared prediction error		
	TP			
Lin. mod	–	0.40 (0.067)		
MS-lasso	3.99 (0.089)	0.40 (0.078)		
Ad. MS-lasso	3.83 (0.39)	0.40 (0.13)		
Ad. liso	4.0 (0)	0.27 (0.076)		
Results with correct information on monotonicity direction provided for each covariate:				
Scam	–	0.20 (0.073)		
Scar	–	0.41 (0.43)		
CPS	–	0.76 (0.17)		
MonBoost	4.0 (0)	0.24 (0.074)		
Mboost	4.0 (0)	0.22 (0.056)		
Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.039 (0.014)	0.038 (0.015)	0.12 (0.025)	0.042 (0.014)
MS-lasso	0.045 (0.030)	0.044 (0.027)	0.074 (0.031)	0.074 (0.027)
Ad. MS-lasso	0.073 (0.060)	0.046 (0.043)	0.055 (0.035)	0.066 (0.037)
Ad. liso	0.025 (0.015)	0.028 (0.013)	0.030 (0.014)	0.026 (0.014)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.0099 (0.0082)	0.0080 (0.0076)	0.012 (0.0080)	0.012 (0.0091)
Scar	0.028 (0.015)	0.032 (0.016)	0.033 (0.015)	0.033 (0.017)
CPS	0.086 (0.073)	0.12 (0.11)	0.16 (0.16)	0.22 (0.20)
MonBoost	0.019 (0.011)	0.022 (0.0099)	0.023 (0.012)	0.021 (0.012)
Mboost	0.014 (0.011)	0.015 (0.011)	0.024 (0.013)	0.020 (0.013)

TABLE A3

Case 1b. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.2, with $n = 150$, $p = 7$, $SNR \approx 4$ and $t = 0$. Standard deviations are given in parenthesis. Monotone splines lasso selected 1 interior knot, scam selected 16 and CPS selected 6. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection			Mean squared prediction error
	TP	FP	
Lin. mod	–	–	0.36 (0.031)
MS-lasso	4.0 (0)	0.43 (0.71)	0.18 (0.027)
Ad. MS-lasso	4.0 (0)	0.010 (0.12)	0.17 (0.028)
Ad. liso	4.0 (0)	0.056 (0.24)	0.17 (0.030)
Results with correct information on monotonicity direction provided for each covariate:			
Scam	–	–	0.14 (0.026)
Scar	–	–	0.20 (0.13)
CPS	–	–	0.37 (0.12)
MonBoost	4.0 (0)	2.98 (0.21)	0.16 (0.027)
Mboost	4.0 (0)	0.53 (0.66)	0.16 (0.024)

Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.036 (0.0075)	0.036 (0.0088)	0.12 (0.013)	0.039 (0.0078)
MS-lasso	0.0070 (0.0054)	0.0070 (0.0049)	0.022 (0.0078)	0.025 (0.010)
Ad. MS-lasso	0.0082 (0.0069)	0.0063 (0.0049)	0.015 (0.0064)	0.025 (0.012)
Ad. liso	0.0090 (0.0033)	0.011 (0.0033)	0.011 (0.0036)	0.0094 (0.0033)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.0033 (0.0026)	0.0027 (0.0022)	0.0047 (0.0027)	0.0038 (0.0025)
Scar	0.010 (0.0041)	0.012 (0.0041)	0.013 (0.0042)	0.012 (0.0044)
CPS	0.033 (0.037)	0.059 (0.058)	0.066 (0.079)	0.084 (0.096)
MonBoost	0.0069 (0.0031)	0.0090 (0.0037)	0.0089 (0.0037)	0.0080 (0.0035)
Mboost	0.0073 (0.0048)	0.0085 (0.0051)	0.016 (0.0060)	0.012 (0.0058)

TABLE A4

Case 1c. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.2, with $n = 50$, $p = 7$, $SNR \approx 4$ and $t = 0$. Standard deviations are given in parenthesis. Monotone splines lasso selected 1 interior knot, scam selected 8 and CPS selected 1. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection			Mean squared prediction error	
	TP	FP		
Lin. mod	–	–	0.39 (0.058)	
MS-lasso	3.98 (0.13)	0.14 (0.37)	0.39 (0.086)	
Ad. MS-lasso	3.76 (0.48)	0 (0)	0.41 (0.15)	
Ad. liso	4.0 (0)	0.14 (0.38)	0.27 (0.070)	
Results with correct information on monotonicity direction provided for each covariate:				
Scam	–	–	0.22 (0.079)	
Scar	–	–	0.50 (0.65)	
CPS	–	–	0.59 (0.20)	
MonBoost	4.0 (0)	2.96 (0.29)	0.28 (0.073)	
Mboost	4.0 (0)	1.70 (0.83)	0.23 (0.058)	
Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.040 (0.016)	0.038 (0.017)	0.12 (0.024)	0.042 (0.014)
MS-lasso	0.046 (0.033)	0.044 (0.030)	0.073 (0.032)	0.075 (0.029)
Ad. MS-lasso	0.076 (0.065)	0.052 (0.055)	0.057 (0.042)	0.070 (0.042)
Ad. liso	0.024 (0.014)	0.028 (0.012)	0.030 (0.013)	0.025 (0.011)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.012 (0.011)	0.0098 (0.0093)	0.018 (0.017)	0.013 (0.011)
Scar	0.031 (0.021)	0.035 (0.020)	0.039 (0.024)	0.037 (0.021)
CPS	0.060 (0.054)	0.099 (0.096)	0.062 (0.049)	0.20 (0.18)
MonBoost	0.021 (0.011)	0.026 (0.013)	0.027 (0.013)	0.025 (0.013)
Mboost	0.016 (0.013)	0.019 (0.014)	0.027 (0.013)	0.023 (0.015)

TABLE A5

Case 2a. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.3, where $n = 80$, $p = 7$, $SNR \approx 4$ and $t = 1$. Standard deviations are given in parenthesis. Monotone splines lasso selected 2 interior knots, scam selected 11 and CPS selected 3. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection			Mean squared prediction error
	TP	FP	
Lin. mod	–	–	0.18 (0.021)
MS-lasso	3.99 (0.11)	0.17 (0.42)	0.14 (0.027)
Ad. MS-lasso	3.80 (0.47)	0 (0)	0.14 (0.046)
Ad. liso	4.0 (0)	0.042 (0.21)	0.087 (0.021)
Results with correct information on monotonicity direction provided for each covariate:			
Scam	–	–	0.058 (0.020)
Scar	–	–	0.18 (0.38)
CPS	–	–	0.25 (0.12)
MonBoost	4.0 (0)	2.98 (0.20)	0.078 (0.017)
Mboost	4.0 (0)	0.42 (0.58)	0.087 (0.017)

Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.017 (0.0074)	0.015 (0.0085)	0.060 (0.019)	0.025 (0.0086)
MS-lasso	0.021 (0.012)	0.025 (0.016)	0.029 (0.013)	0.040 (0.019)
Ad. MS-lasso	0.031 (0.025)	0.026 (0.025)	0.023 (0.018)	0.043 (0.039)
Ad. liso	0.0092 (0.0053)	0.011 (0.0052)	0.011 (0.0054)	0.010 (0.0044)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.0031 (0.0029)	0.0027 (0.0022)	0.0035 (0.0035)	0.0031 (0.0026)
Scar	0.0094 (0.0055)	0.010 (0.0055)	0.010 (0.0059)	0.011 (0.0049)
CPS	0.026 (0.031)	0.050 (0.052)	0.013 (0.020)	0.13 (0.15)
MonBoost	0.0070 (0.0042)	0.0095 (0.0052)	0.0085 (0.0042)	0.0097 (0.0053)
Mboost	0.014 (0.0080)	0.018 (0.012)	0.019 (0.0090)	0.022 (0.013)

TABLE A6

Case 2b. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.3, where SNR ≈ 4, $t = 1$, $p = 7$ *and* $n = 150$. *Standard deviations are given in parenthesis. Monotone splines lasso selected* 2 *interior knots, scam selected* 16 *and CPS selected* 4. *"Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.*

Selection			Mean squared prediction error	
	TP	FP		
Lin. mod	–	–	0.17 (0.018)	
MS-lasso	4.0 (0)	0.056 (0.24)	0.093 (0.014)	
Ad. MS-lasso	3.99 (0.089)	0 (0)	0.088 (0.015)	
Ad. liso	4.0 (0)	0.008 (0.089)	0.061 (0.0090)	
Results with correct information on monotonicity direction provided for each covariate:				
Scam	–	–	0.040 (0.017)	
Scar	–	–	0.14 (0.32)	
CPS	–	–	0.19 (0.097)	
MonBoost	4.0 (0)	2.95 (0.31)	0.055 (0.0085)	
Mboost	4.0 (0)	0.050 (0.22)	0.076 (0.010)	
Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.015 (0.0052)	0.013 (0.0059)	0.061 (0.014)	0.024 (0.0057)
MS-lasso	0.012 (0.0057)	0.014 (0.0065)	0.018 (0.0061)	0.024 (0.0082)
Ad. MS-lasso	0.015 (0.0093)	0.012 (0.0078)	0.010 (0.0054)	0.026 (0.0098)
Ad. liso	0.0051 (0.0023)	0.0061 (0.0024)	0.0060 (0.0023)	0.0055 (0.0018)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.0013 (0.0013)	0.0013 (0.0011)	0.0016 (0.0012)	0.0014 (0.0011)
Scar	0.0048 (0.0021)	0.0052 (0.0018)	0.0055 (0.0020)	0.0060 (0.0024)
CPS	0.021 (0.028)	0.044 (0.042)	0.013 (0.021)	0.10 (0.12)
MonBoost	0.0039 (0.0019)	0.0052 (0.0023)	0.0047 (0.0022)	0.0050 (0.0021)
Mboost	0.013 (0.0057)	0.016 (0.0073)	0.017 (0.0065)	0.019 (0.0074)

TABLE A7

Case 3. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.3, where $n = 80$, $p = 7$, $SNR \approx 2$ *and* $t = 0$. *Standard deviations are given in parenthesis. Monotone splines lasso selected* 1 *interior knot, scam selected* 12 *and CPS selected* 2. *"Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.*

Selection			Mean squared prediction error
	TP	FP	
Lin. mod	–	–	0.71 (0.092)
MS-lasso	4.0 (0)	0.51 (0.71)	0.63 (0.093)
Ad. MS-lasso	3.93 (0.26)	0.04 (0.24)	0.62 (0.11)
Ad. liso	4.0 (0)	0.68 (0.89)	0.64 (0.12)
Results with correct information on monotonicity direction provided for each covariate:			
Scam	–	–	0.60 (0.12)
Scar	–	—	0.90 (0.57)
CPS	–	—	0.87 (0.20)
MonBoost	4.0 (0)	2.98 (0.20)	0.65 (0.12)
Mboost	4.0 (0)	2.16 (0.74)	0.56 (0.089)

Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.042 (0.017)	0.040 (0.019)	0.13 (0.022)	0.043 (0.016)
MS-lasso	0.033 (0.026)	0.030 (0.024)	0.054 (0.025)	0.060 (0.027)
Ad. MS-lasso	0.046 (0.046)	0.030 (0.030)	0.040 (0.024)	0.055 (0.031)
Ad. liso	0.035 (0.019)	0.042 (0.020)	0.045 (0.021)	0.041 (0.021)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.023 (0.019)	0.017 (0.015)	0.028 (0.021)	0.024 (0.018)
Scar	0.048 (0.030)	0.052 (0.026)	0.057 (0.026)	0.055 (0.028)
CPS	0.063 (0.058)	0.10 (0.093)	0.079 (0.078)	0.16 (0.15)
MonBoost	0.029 (0.017)	0.033 (0.018)	0.036 (0.019)	0.032 (0.019)
Mboost	0.021 (0.016)	0.022 (0.018)	0.032 (0.018)	0.028 (0.020)

TABLE A8

Case 4. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 4.1.3, where SNR ≈ 4, $t = 0$, $p = 20$ *and* $n = 200$. *Standard deviations are given in parenthesis. Monotone splines lasso selected* 1 *interior knot, scam selected* 11 *knots and CPS selected* 2 *interior knots. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.*

Selection			Mean squared prediction error	
	TP	FP		
Lin. mod	–	–	0.35 (0.030)	
MS-lasso	4.0 (0)	3.80 (2.78)	0.15 (0.019)	
Ad. MS-lasso	4.0 (0)	0.25 (1.14)	0.14 (0.019)	
Ad. liso	4.0 (0)	0.014 (0.12)	0.15 (0.022)	
Results with correct information on monotonicity direction provided for each covariate:				
Scam	–	–	0.16 (0.032)	
Scar	–	–	0.20 (0.16)	
CPS	–	–	0.19 (0.050)	
MonBoost	4.0 (0)	12.78 (5.18)	0.16 (0.025)	
Mboost	4.0 (0)	1.18 (1.05)	0.15 (0.021)	
Mean squared estimation error				
	g_1	g_2	g_3	g_4
Lin. mod	0.034 (0.0058)	0.033 (0.0064)	0.12 (0.012)	0.037 (0.0059)
MS-lasso	0.0035 (0.0026)	0.0040 (0.0024)	0.016 (0.0049)	0.016 (0.0071)
Ad. MS-lasso	0.0031 (0.0028)	0.0032 (0.0020)	0.011 (0.0033)	0.013 (0.0087)
Ad. liso	0.0072 (0.0024)	0.0085 (0.0023)	0.0090 (0.0026)	0.0073 (0.0024)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.0039 (0.0038)	0.0028 (0.0032)	0.0098 (0.0082)	0.0066 (0.0060)
Scar	0.0087 (0.0032)	0.010 (0.0032)	0.011 (0.0033)	0.0098 (0.0033)
CPS	0.0073 (0.0073)	0.015 (0.019)	0.018 (0.0073)	0.029 (0.36)
MonBoost	0.0060 (0.0025)	0.0077 (0.0031)	0.0078 (0.0031)	0.0073 (0.0032)
Mboost	0.0062 (0.0035)	0.0074 (0.0038)	0.015 (0.0050)	0.010 (0.0045)

TABLE A9

Case 6. Average number of total true and false positives, mean squared prediction errors and mean squared estimation errors for the estimated functions, in the simulation considered in Section 6 with a non-monotone covariate effect g_{4b}, where SNR ≈ 4, $t = 0$, $p = 7$ and $n = 80$. Standard deviations are given in parenthesis. Monotone splines lasso selected 1 interior knot, scam selected 12 interior knots and CPS selected 3 interior knots. "Lin. mod" is the ordinary least squares fit, "MS-lasso" is the monotone splines lasso, "Ad. MS-lasso" is the adaptive monotone splines lasso and "Ad. liso" is the adaptive liso.

Selection			Mean squared prediction error
	TP	FP	
Lin. mod	–	–	0.89 (0.068)
MS-lasso	4.0 (0)	0.25 (0.50)	0.34 (0.066)
Ad. MS-lasso	3.95 (0.21)	0 (0)	0.34 (0.087)
Ad. liso	4.0 (0)	0.18 (0.40)	0.23 (0.054)
Results with correct information on monotonicity direction provided for each covariate:			
Scam	–	–	0.74 (0.13)
Scar	–	–	0.87 (0.53)
CPS	–	–	0.71 (0.080)
MonBoost	4.0 (0)	2.95 (0.36)	0.65 (0.087)
Mboost	4.0 (0)	2.28 (0.71)	0.58 (0.059)

Mean squared estimation error				
	g_1	g_2	g_3	g_{4b}
Lin. mod	0.047 (0.037)	0.041 (0.019)	0.13 (0.026)	0.53 (0.066)
MS-lasso	0.022 (0.015)	0.019 (0.014)	0.043 (0.016)	0.097 (0.037)
Ad. MS-lasso	0.054 (0.039)	0.030 (0.022)	0.040 (0.020)	0.051 (0.036)
Ad. liso	0.017 (0.0077)	0.020 (0.0075)	0.020 (0.0072)	0.033 (0.011)
Results with correct information on monotonicity direction provided for each covariate:				
Scam	0.025 (0.018)	0.026 (0.020)	0.034 (0.025)	0.46 (0.13)
Scar	0.052 (0.030)	0.058 (0.030)	0.062 (0.031)	0.40 (0.051)
CPS	0.017 (0.023)	0.033 (0.045)	0.030 (0.035)	0.51 (0.066)
MonBoost	0.030 (0.018)	0.035 (0.018)	0.037 (0.019)	0.35 (0.069)
Mboost	0.022 (0.021)	0.022 (0.018)	0.033 (0.020)	0.36 (0.071)

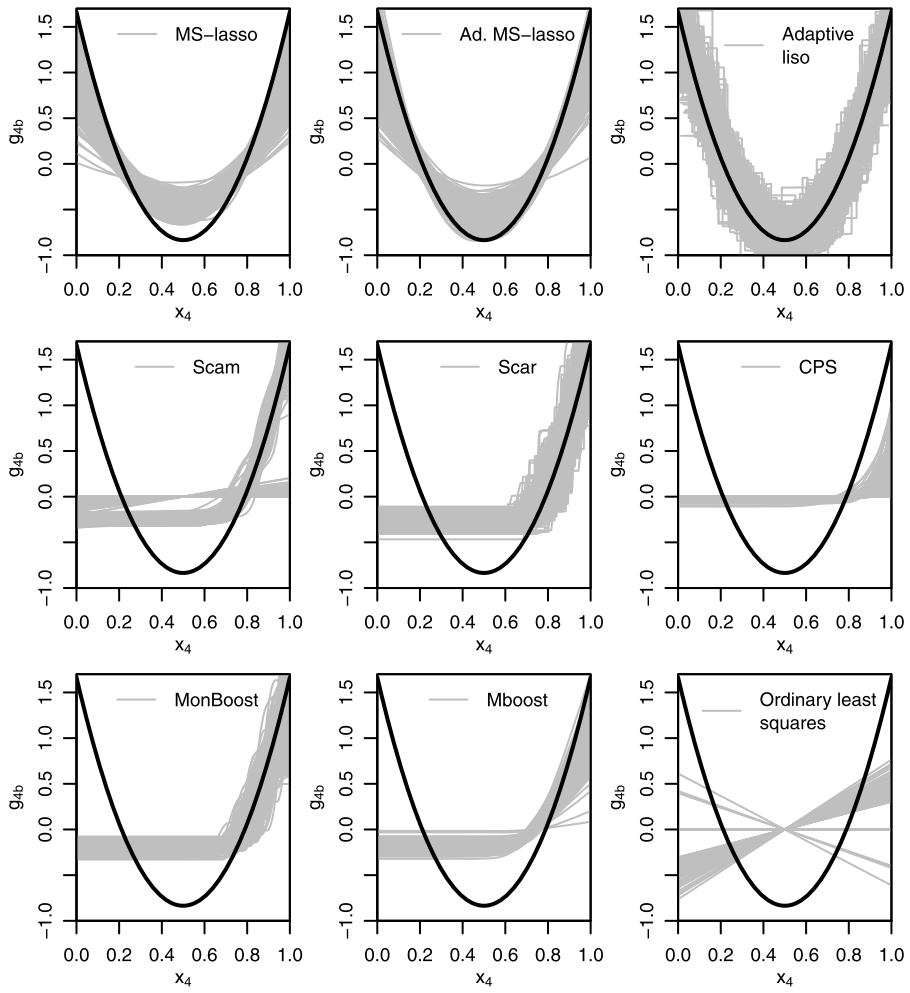

FIG A1. *Case 6. Estimated functions for g_{4b} in the simulation considered in Section 6 with $n = 80$, $p = 7$, SNR ≈ 4 and $t = 0$, for all the different methods, with a non-monotone function. The true function is given in black.*

Appendix B: Algorithm for MonBoost

The algorithm in the one dimensional case with Gaussian response is provided in [44], and is restated in Algorithm 1, where M denotes the number of iterations, $\mathbf{y}$ is the vector with observed responses and $\mathbf{x}$ is the vector with observations of the predictor, and the covariate index j is dropped for notational convenience. The extension to the multivariate case is straight forward, with $m \times p$ basis functions instead of only m.

Algorithm 1 MonBoost

Initialise:

Standardise $\mathbf{y}$ to have mean zero, so $\hat{\boldsymbol{\mu}}^{(0)} = (\bar{y}, \ldots, \bar{y})$. Let $\hat{\beta}_0 = (0, \ldots, 0)$.

Iteration:

for r=1 to M **do**

$\mathbf{u}^{(r)} = \mathbf{y} - \hat{\boldsymbol{\mu}}^{(r-1)}$ ▷ Compute the current residuals

for k=1 to m **do**

Compute the ridge estimators $\hat{\beta}_k$ with tuning parameter λ for the model

$$\mathbf{u}^{(r)} = \beta_k I_k^{(l)}(\mathbf{x}) + \boldsymbol{\epsilon}.$$

end for

From the subset of components that fulfill the constraint $\hat{\beta}_k^{(r)} = \hat{\beta}_k^{(r-1)} + \hat{\beta}_k \geq 0$, choose the component $\hat{\gamma}^{(r)}$ which minimises $||\mathbf{u}^{(r)} - \hat{\beta}_k I_k^{(l)}(x)||^2$.

if $\hat{\beta}_k^{(r)} \leq 0$ for all k **then**

stop iteration.

else

$\hat{\gamma}^{(r)} = k$.

end if

Set

$$\hat{\beta}_k^{(r)} = \begin{cases} \hat{\beta}_k^{(r-1)} + \hat{\beta}_k, & \text{if } k = \hat{\gamma}^{(r)}, \\ \hat{\beta}_k^{(r-1)}, & \text{otherwise,} \end{cases}$$

and

$$\hat{\boldsymbol{\mu}}^{(r)} = \hat{\boldsymbol{\mu}}^{(r-1)} + \hat{\beta}_{\hat{\gamma}^{(r)}} I_{\hat{\gamma}^{(r)}}^{(l)}(\mathbf{x}).$$

end for