Regularization Methods for Additive Models

Marta Avalos, Yves Grandvalet, and Christophe Ambroise

HEUDIASYC Laboratory UMR CNRS 6599
Compiègne University of Technology
BP 20529 / 60205 Compiègne, France
{avalos, grandvalet, ambroise}@hds.utc.fr
http://www.hds.utc.fr

Abstract. This paper tackles the problem of model complexity in the context of additive models. Several methods have been proposed to estimate smoothing parameters, as well as to perform variable selection. Nevertheless, these procedures are inefficient or computationally expensive in high dimension. Also, the *lasso* technique has been adapted to additive models, however its experimental performance has not been analyzed.
We propose a modified *lasso* for additive models, improving variable selection. A benchmark is also developed, to examine its practical behavior, comparing it with forward selection. Our simulation studies suggest ability to carry out model selection of the proposed method. The *lasso* technique shows up better than forward in the most complex situations. The computing time of modified *lasso* is considerably smaller since it does not depend on the number of relevant variables.

1 Introduction

Additive nonparametric regression model has become a useful statistical tool in analysis of high–dimensional data sets. An additive model [10] is defined by

$$Y = f_0 + \sum_{j=1}^{p} f_j(X_j) + \varepsilon, \tag{1}$$

where the errors ε are independent of the predictor variables X_j, $\mathbb{E}(\varepsilon) = 0$ and $var(\varepsilon) = \sigma^2$. The f_j, are univariate smooth functions and f_0 is a constant. Y is the response variable.

This model's popularity is due to its flexibility, as a nonparametric method, but also, to its interpretability. Furthermore, additive regression gets round the curse of dimensionality.

Some issues related to model complexity have been studied in the context of additive models. Several methods have been proposed to estimate smoothing parameters [10], [8], [11], [15], [4]. These methods are based on generalizing univariate techniques. Nevertheless, the application of these procedures in high

dimension is often inefficient or highly time consuming. The choice of the degree of smoothing is a complicated problem: although univariate in nature, that remains a multivariate problem.

Also, variable selection methods have been formulated for additive models [10], [5], [13], [3], [15]. These proposals exploit the fact that additive regression generalizes linear regression. Since nonparametric methods are used to fit the terms, model selection develops some new flavours. We not only need to select which terms to include in the model, but also how smooth they should be, then, even for few variables, these methods are computationally expensive.

Finally, the *lasso* technique [14] has been adapted to additive models fitted by splines. The lasso (least absolute shrinkage and selection operator) is a regularization procedure intended to tackle the problem of selection of accurate and interpretable linear models. The lasso estimates a vector of regression coefficients by minimizing the residual sum of squares subject to a constraint on the l_1–norm of coefficient vector.

For additive models, this technique transforms a high–dimensional into a low–dimensional hyper–parameter selection problem, which implies many advantages. Some algorithms have been proposed [6], [7], [1]. Their experimental performance has not been, however, analyzed, and above all, these discriminate between linear and nonlinear variables, but does not perform variable selection.

There are many other approaches to model selection methods for supervised regression tasks (see for example [9]). Computational costs are a primary issue in their application to additive models.

We propose a modified lasso for additive spline regression, in order to discriminate between linear and nonlinear variables, but also, between relevant and irrelevant variables. We also develop a benchmark based on Breiman's work [2], to examine the practical behavior of the modified lasso, comparing it to forward variable selection. We focus on those situations where control of complexity is a major problem. Results allow us to deduce conditions of application of each regularization method. The computing time of modified lasso is considerably smaller than the computing time of forward variable selection since it does not depend on the number of relevant variables (when the total number of input variables is fixed).

Section 2 presents lasso applied to additive models. Modifications are introduced in section 3, as well as algorithmic issues. The schema of the testing efficiency procedure and benchmark for additive models are presented in section 4. Section 5 gives simulation results and conclusions.

2 Lasso Adapted to Additive Models

Grandvalet et al. [6], [7] showed the equivalence between adaptive ridge regression and lasso. Thanks to this link, authors derived an EM algorithm to compute the lasso estimates. Subsequently, results obtained for the linear case were generalized to additive models fitted by cubic smoothing splines.

Suppose that one has data $\mathcal{L} = \{(\mathbf{x}, \mathbf{y})\}$, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$, $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^t$, $\mathbf{y} = (y_1, \ldots, y_n)^t$. To simplify, assume that the responses are centered. We remind that the lasso estimate is given by the solution of the following constrained optimization problem

$$\min_{\alpha_1,\ldots,\alpha_p} \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \alpha_j\right)^t \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \alpha_j\right) \quad \text{subject to} \quad \sum_{j=1}^{p} |\alpha_j| \leq \tau, \tag{2}$$

and the adaptive ridge estimate is the minimizer of the problem

$$\min_{\alpha_1,\ldots,\alpha_p} \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \alpha_j\right)^t \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \alpha_j\right) + \sum_{j=1}^{p} \lambda_j \alpha_j^2, \tag{3}$$

subject to

$$\sum_{j=1}^{p} \frac{1}{\lambda_j} = \frac{p}{\lambda} \qquad \lambda_j > 0, \tag{4}$$

where τ and λ are predefined values.

We also remind that the cubic smoothing spline is defined as the minimizer of the penalized least squares criterion over all twice–continuously–differentiable functions. This idea is extended to additive models in a straightforward manner:

$$\min_{f_1,\ldots,f_p \in \mathcal{C}^2} \left(\mathbf{y} - \sum_{j=1}^{p} f_j(\mathbf{x}_j)\right)^t \left(\mathbf{y} - \sum_{j=1}^{p} f_j(\mathbf{x}_j)\right) + \sum_{j=1}^{p} \lambda_j \int_{a_j}^{b_j} f_j''(x)^2 dx, \tag{5}$$

where $[a_j, b_j]$ is the interval for which an estimate of f_j is sought. This interval is arbitrary, as long as it contains the data; $\widehat{f}_j$ is linear beyond the extreme data points no matter what the values of a_j and b_j are. Each function in (5) is penalized by a separate fixed smoothing parameter λ_j.

Let $\mathbf{N}_j$ denote the $n \times (n+2)$ matrix of the unconstrained natural B–spline basis, evaluated at x_{ij}. Let $\mathbf{\Omega}_j$ be the $(n+2) \times (n+2)$ matrix corresponding to the penalization of the second derivative of $\widehat{\mathbf{f}}_j$. The coefficients of $\widehat{\mathbf{f}}_j$ in the unconstrained B–spline basis are noted β_j. Then, the extension of lasso to additive models fitted by cubic splines, using the equivalence between (2) and (3)–(4) is given by

$$\min_{\beta_1,\ldots,\beta_p} \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{N}_j \beta_j\right)^t \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{N}_j \beta_j\right) + \sum_{j=1}^{p} \lambda_j \beta_j^t \mathbf{\Omega}_j \beta_j, \tag{6}$$

subject to

$$\sum_{j=1}^{p} \frac{1}{\lambda_j} = \frac{p}{\lambda} \qquad \lambda_j > 0, \tag{7}$$

where λ is a predefined value. The expression (6) shows that this problem is equivalent to a standard additive spline model, where the penalization terms λ_j applied to each additive component are optimized subject to constraints (7). This problem has the same solution as

$$\min_{\beta_1,\ldots,\beta_p} \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{N}_j\beta_j\right)^t \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{N}_j\beta_j\right) + \frac{\lambda}{p}\left(\sum_{j=1}^{p}\sqrt{\beta_j^t\mathbf{\Omega}_j\beta_j}\right)^2. \quad (8)$$

The penalizer in (7) generalizes the lasso penalizer $\sum_{j=1}^{p}|\alpha_j| = \sum_{j=1}^{p}\sqrt{\alpha_j^2}$. Note that the writing (6)–(7) can also be motivated from a hierarchical Bayesian viewpoint.

Grandvalet et al. proposed a modified EM algorithm, including backfitting (see section 3.2), to estimate coefficients β_j. This method does not perform variable selection. When after convergence $\widehat{\beta}_j^t\mathbf{\Omega}_j\widehat{\beta}_j = 0$, the jth predictor is not eliminated but linearized.

Another algorithm based on sequential quadratic programming was suggested by Bakin [1]. This methodology seems to be, however, more complex than the precedent one and does not perform variable selection either.

3 Modified Lasso

3.1 The Smoother Matrix

Splines are linear smoothers, that is, the univariate fits can be written as $\widehat{\mathbf{f}}_j = \mathbf{S}_j\mathbf{y}$, where $\mathbf{S}_j$ is an $n \times n$ matrix called *smoother matrix*. The latter depends on the smoothing parameter and the observed points $\mathbf{x}_j$, but not on $\mathbf{y}$.

The smoother matrix of a cubic smoothing spline has two unitary eigenvalues corresponding to the constant and linear functions (its projection part), and $n-2$ non–negative eigenvalues strictly less than 1 corresponding to different compounds of the non-linear part (its *shrinking* part). Also, $\mathbf{S}_j$ is symmetric, then $\mathbf{S}_j = \mathbf{G}_j + \widetilde{\mathbf{S}}_j$, where $\mathbf{G}_j$ is the matrix that projects onto the space of eigenvalue 1 for the jth smoother, and $\widetilde{\mathbf{S}}_j$ is the *shrinking* matrix [10].

For cubic smoothing splines, $\mathbf{G}_j$ is the *hat* matrix corresponding to the least–squares regression on $(\mathbf{1}, \mathbf{x}_j)$, the smoother matrix is calculated as $\mathbf{S}_j = \mathbf{N}_j(\mathbf{N}_j^t\mathbf{N}_j + \lambda_j\mathbf{\Omega}_j)^{-1}\mathbf{N}_j^t$, and $\widetilde{\mathbf{S}}_j$ is found by $\mathbf{S}_j - \mathbf{G}_j$:

$$\widetilde{\mathbf{S}}_j = \mathbf{S}_j - \left(\frac{1}{n}\mathbf{1}\mathbf{1}^t + \mathbf{x}_j\left(\mathbf{x}_j^t\mathbf{x}_j\right)^{-1}\mathbf{x}_j^t\right). \quad (9)$$

3.2 The Backfitting Algorithm

Additive models can be estimated by the backfitting algorithm which consists in fitting iteratively $\widehat{\mathbf{f}}_j = \mathbf{S}_j(\mathbf{y} - \sum_{k\neq j}\widehat{\mathbf{f}}_k)$, $j = 1, \ldots, p$.

Taking into account the decomposition of $\mathbf{S}_j$, the backfitting algorithm can be divided into two steps: 1. estimation of the projection part, $\mathbf{g} = \mathbf{G}(\mathbf{y} - \sum \widetilde{\mathbf{f}}_j)$, where $\mathbf{G}$ is the *hat* matrix corresponding to the least–squares regression on $(\mathbf{1}, \mathbf{x}_1, \ldots, \mathbf{x}_p)$, and 2. estimation of the shrinking parts, $\widetilde{\mathbf{f}}_j = \widetilde{\mathbf{S}}_j(\mathbf{y} - \mathbf{g} - \sum_{k \neq j} \widetilde{\mathbf{f}}_k)$. The final estimate for the overall fit is $\widehat{\mathbf{f}} = \mathbf{g} + \sum \widetilde{\mathbf{f}}_j$.

In addition, linear and nonlinear parts work on orthogonal spaces: $\mathbf{G}\widetilde{\mathbf{f}}_j = \mathbf{0}$ and $\widetilde{\mathbf{S}}_j \mathbf{g} = \mathbf{0}$, so estimation of additive models fitted by cubic splines, using the backfitting algorithm can be separated in its linear and its nonlinear part.

3.3 Isolating Linear from Nonlinear Penalization Terms

Penalization term in (7) only acts on the nonlinear components of $\widehat{\mathbf{f}}$. Consequently, severely penalized covariates are not eliminated but linearized. Another term acting on the linear component should be applied to perform subset selection.

The previous decomposition of cubic splines allow us to write linear and nonlinear part separately:

$$\widehat{\mathbf{f}} = \mathbf{g} + \widetilde{\mathbf{f}} = \sum_{j=1}^{p} \mathbf{x}_j \widehat{\alpha}_j + \sum_{j=1}^{p} \widetilde{\mathbf{f}}_j(\mathbf{x}_j) = \mathbf{x}\widehat{\alpha} + \sum_{j=1}^{p} \widetilde{\mathbf{N}}_j \widetilde{\beta}_j, \tag{10}$$

where $\widetilde{\mathbf{N}}_j$ denotes the matrix of the nonlinear part of the unconstrained spline basis, evaluated at x_{ij}, $\widetilde{\beta}_j$ denotes the coefficients of $\widetilde{\mathbf{f}}_j$ in the nonlinear part of the unconstrained spline basis, and $\alpha = (\alpha_1, \ldots, \alpha_p)^t$ denotes linear least squares coefficients.

Regarding penalization terms, a simple extension of (7) is to minimize (with respect to α and $\widetilde{\beta}_j$, $j = 1, \ldots, p$)

$$\left(\mathbf{y} - \mathbf{x}\alpha - \sum_{j=1}^{p} \widetilde{\mathbf{f}}_j\right)^t \left(\mathbf{y} - \mathbf{x}\alpha - \sum_{j=1}^{p} \widetilde{\mathbf{f}}_j\right) + \sum_{j=1}^{p} \mu_j \alpha_j^2 + \sum_{j=1}^{p} \lambda_j \widetilde{\beta}_j^t \widetilde{\mathbf{\Omega}}_j \widetilde{\beta}_j, \tag{11}$$

subject to

$$\sum_{j=1}^{p} \frac{1}{\mu_j} = \frac{p}{\mu}, \qquad \mu_j > 0 \qquad \text{and} \qquad \sum_{j=1}^{p} \frac{1}{\lambda_j} = \frac{p}{\lambda}, \qquad \lambda_j > 0, \tag{12}$$

where μ and λ are predefined values, and $\widetilde{\mathbf{\Omega}}_j$ is the matrix corresponding to the penalization of the second derivative of $\widetilde{\mathbf{f}}_j$.

When after convergence, $\mu_j \approx \infty$ and $\lambda_j \approx \infty$, the jth covariate is eliminated. If $\mu_j < \infty$ and $\lambda_j \approx \infty$, the jth covariate is linearized. When $\mu_j \approx \infty$ and $\lambda_j < \infty$, the jth covariate is estimated to be strictly nonlinear.

3.4 Algorithm

1. Initialize: $\mu_j = \mu$, $\mathbf{\Lambda} = \mu \mathbf{I}_p$, $\lambda_j = \lambda$.
2. Linear components:
   (a) Compute linear coefficients: $\alpha = \left(\mathbf{x}^t\mathbf{x} + \mathbf{\Lambda}\right)^{-1} \mathbf{x}^t\mathbf{y}$.
   (b) Compute linear penalizers: $\mu_j = \mu \frac{\|\alpha\|_1}{p|\alpha_j|}$, $\qquad \mathbf{\Lambda} = \mathrm{diag}(\mu_j)$.
3. Repeat step 2 until convergence.
4. Nonlinear components:
   (a) Initialize: $\widetilde{\mathbf{f}}_j,\ j = 1, \ldots, p,\ \widetilde{\mathbf{f}} = \sum_j^p \widetilde{\mathbf{f}}_j$.
   (b) Calculate: $\mathbf{g} = \mathbf{G}\left(\mathbf{y} - \widetilde{\mathbf{f}}\right)$.
   (c) One cycle of backfitting: $\widetilde{\mathbf{f}}_j = \widetilde{\mathbf{S}}_j\left(\mathbf{y} - \mathbf{g} - \sum_{k \neq j} \widetilde{\mathbf{f}}_k\right)$, $j = 1, \ldots, p$.
   (d) Repeat step (b) and (c) until convergence.
   (e) Compute $\widetilde{\beta}_j$ from the final estimates
   $\widetilde{\beta}_j = \left(\widetilde{\mathbf{N}}_j^t \widetilde{\mathbf{N}}_j + \lambda_j \widetilde{\mathbf{\Omega}}_j\right)^{-1} \widetilde{\mathbf{N}}_j^t \left(\mathbf{y} - \sum_{k \neq j} \widetilde{\mathbf{f}}_k\right)$.
   (f) Compute nonlinear penalizers: $\lambda_j = \lambda \dfrac{\sum_{j=1}^p \sqrt{\widetilde{\beta}_j^t \widetilde{\mathbf{\Omega}}_j \widetilde{\beta}_j}}{p\sqrt{\widetilde{\beta}_j^t \widetilde{\mathbf{\Omega}}_j \widetilde{\beta}_j^t}}$.
5. Repeat step 4 until convergence.

In spite of orthogonality, projection step 4.(b) is iterated with backfitting step 4.(c) to improve numerical stability of the algorithm. In order to compute nonlinear penalizers in 4.(f), calculating β_j instead of $\widetilde{\beta}_j$ in 4.(e) is sufficient, since $\beta_j^t \mathbf{\Omega}_j \beta_j^t$ is insensitive to the linear components.

An efficient lasso algorithm was proposed by Osborne et al. [12]. This one can be used for the linear part of the algorithm, and may be adapted to the nonlinear part.

3.5 Complexity Parameter Selection

The initial multidimensional parameter selection problem is transformed into a 2–dimensional problem. A popular criterion for choosing complexity parameters is cross–validation, which is an estimate of the prediction error (see section 4.1). Calculating the CV function, is computationally intensive. A fast approximation of CV is generalized cross–validation [10], [8], [14]:

$$\mathrm{GCV}(\mu, \lambda) = \frac{1}{n} \frac{(\mathbf{y} - \widehat{\mathbf{f}})^t (\mathbf{y} - \widehat{\mathbf{f}})}{\left(1 - df(\mu, \lambda)/n\right)^2}. \tag{13}$$

The GCV function is evaluated over a 2–dimensional grid of values. The point $(\widehat{\mu}, \widehat{\lambda})$ yielding the lowest GCV is selected. The effective number of parameters (or the effective degrees of freedom) *df* is estimated by

$$df(\mu, \lambda) \approx \sum_{j=1}^p df_j(\mu, \lambda) = \mathrm{tr}\left[\mathbf{x}\left(\mathbf{x}^t\mathbf{x} + \mathbf{\Lambda}\right)^{-1} \mathbf{x}^t\right] + \sum_{j=1}^p \mathrm{tr}\left[\widetilde{\mathbf{S}}_j(\lambda_j)\right]. \tag{14}$$

The *df* of the overall function is approximated by the sum of the univariate effective number of parameters, df_j [10]. Usually, variable selection methods do not take into account the effect of model selection [16]. It is assumed that the selected model is given a priori. The present estimate suffers from the same inaccuracy: the cost of the individual penalization terms estimation is not measured.

4 Efficiency Testing Procedure

Our goal is to analyze the modified lasso behavior and compare it to another model selection method: forward variable selection. Comparison criteria and procedures for simulations are detailed next.

4.1 Comparison Criteria

The objective of prediction is to construct a function $f_{\mathcal{L}}$ providing accurate prediction of future examples. That is, we want the prediction error

$$\mathrm{PE}(f) = \mathbb{E}_{Y\mathbf{X}}[(Y - f(\mathbf{X}))^2] \tag{15}$$

to be small.

Let $\widehat{f_{\mathcal{L}}}$ be the underlying function estimator, which depends on the complexity parameters. Let $\widehat{\lambda}$ et λ^* denote complexity parameters estimations, the former is calculated by a given method, and the latter is the value minimizing PE, estimated from an "infinite" test set (Breiman's *crystal ball* [2]):

$$\widehat{\lambda} = \underset{\lambda}{\operatorname{argmin}}\, \widehat{\mathrm{PE}}\left(\widehat{f}(.,\lambda)\right), \qquad \lambda^* = \underset{\lambda}{\operatorname{argmin}}\, \mathrm{PE}\left(\widehat{f}(.,\lambda)\right). \tag{16}$$

4.2 A Benchmark for Additive Models

Our objective is to define the bases of a benchmark for additive models, based on Breiman's work: "Which situations make estimation difficult?" and "What parameters are these situations controlled through?"

The number of observations–effective number of parameters ratio, n/df. When n/df is high, the problem is easy to solve. Every "reasonable" method will find an accurate solution. Conversely, a low ratio makes the problem insolvable. The effective number of parameters depends on the number of covariates, p, and on other parameters described next.

Concurvity. Concurvity describes a situation in which predictors are linearly or nonlinearly dependent. This phenomenon causes non–unicity of estimations. The particular case of collinearity (linear dependence) can be controlled through the correlation matrix of predictors. Predictors are normally distributed:

$$\mathbf{X} \sim \mathcal{N}_p(\mathbf{0}, \mathbf{\Gamma}), \qquad \Gamma_{ij} = \rho^{|i-j|}. \tag{17}$$

The number of relevant variables. The nonzero coefficients are in two clusters of adjacent variables with centers at fixed l's. The coefficient values are given by

$$\delta_j = \mathbb{I}_{\{\omega - |j-l| > 0\}}, \quad j = 1, \ldots, p, \tag{18}$$

where ω is a fixed integer governing the cluster width.

Underlying functions. The more the structures of the underlying functions are complex, the harder they are to estimate. One factor that makes the structure complex is rough changes in curvature. Sine functions allow us to handle diverse situations, from almost linear to highly zig-zagged curves. We are interested in controlling intra– and inter–covariates curvatures changes, thus define:

$$f_j(\mathbf{x}_j) = \delta_j \sin 2\pi k_j \mathbf{x}_j, \tag{19}$$

and $f_0 = 0$. Consider $\kappa = k_{j+1} - k_j$, $j = 1, \ldots, p-1$, the sum $\sum_{j=1}^{p} k_j$ fixed to keep the same overall degree of complexity.

Noise level. Noise is introduced through error variance

$$Y = \textstyle\sum_{j=1}^{p} f_j + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{20}$$

In order to avoid sensitivity to scale, the noise effect is controlled through the determination coefficient, which is a function of σ.

5 Results

5.1 Example

Figure (1) reproduces the simulated example in dimension five used in [7]. The fitted univariate functions are plotted for four values of (μ, λ). The response variable depends linearly on the first covariate. The last covariate is not relevant. The other covariates affect the response, but the smoothness of the dependancies decreases with the coordinate number of the covariates.

For $\mu = \lambda = 1$, the individual penalization terms of the last covariate, μ_5, λ_5, were estimated to be extremely large, as well as the individual nonlinear penalization term corresponding to the first covariate, λ_1. Therefore, the first covariate was estimated to be linear and the last one was estimated to be irrelevant. For high values of μ and λ, the dependences on the least smooth covariates are difficult to capture.

5.2 Comparison

Implementation. The forward selection version of the backward elimination algorithm given in [3] was implemented. The GCV criteria is used to select variables as well as to choose smoothing parameters. The GCV function was evaluated over a grid which dimension depends on the number of variables selected in the

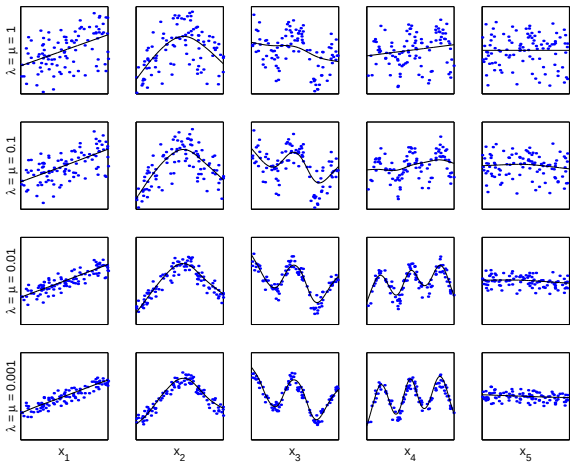

Fig. 1. Modified lasso on simulated data. The underlying model is $y = x_1 + \cos(\pi x_2) + \cos(2\pi x_3) + \cos(3\pi x_4) + \varepsilon$. The covariates are independently drawn from a uniform distribution on $[-1, 1]$ and ε is a Gaussian noise of standard deviation $\sigma = 0.3$. The solid curves are the estimated univariate functions for different values of (μ, λ), and dots are partial residuals.

prior iteration. Thus, models with $p = 1, 2, 3, 4$ and $p = 5$ or more variables were evaluated over a grid of 25^p, 5^p, 4^p, 3^p, and 2^p values, respectively. The PE value was estimated from a test set of size 20000 sampled from the same distribution as the learning set.

The GCV criteria is used to select complexity parameters of modified lasso. The GCV function was evaluated over a grid of 5^2 values. A reasonably large range was adopted to produce the complexity parameters space, specifically, $\lambda, \mu \in [0.02, 4]$. The log–values were equally spaced on the grid. The PE value of lasso was calculated similarly to the PE of forward selection.

Simulations. Parameters of control described in section (4.2) were fixed as follows. The number of observations and the number of covariates were, respectively, $n = 450$ and $p = 12$. We considered two cases for ρ, low ($\rho = 0.1$) and severe ($\rho = 0.9$) concurvity. With respect to relevant variables, the situations taken into account in simulations were: $\omega = \{1, 2, 3\}$, taken clusters centered at $l = 3$ and $l = 9$. This gave 2, 6 and 10 relevant variables, over a total of 12. No different curvature changes between covariates: $\kappa = 0$ and moderate curvature changes within covariates: $\sum k_j = 9$, were taken into account. We studied a lowly noisy situation: $R^2 = 0.9$.

Table (1) shows estimated PE values for the two methods in comparison, modified lasso using the GCV criteria and forward variable selection, and PE values for the optimal modified lasso, Lasso*, (that is, complexity parameters are the minimizers of PE, they are estimated from an "infinite" test set). We observe that only in the most difficult situations: 6 or 10 relevant variables ($\omega = 2$ or

3) and severe correlation ($\rho = 0.9$), the lasso applied to additive models had a lower PE than forward selection. Estimated PE values corresponding to modified lasso using the GCV criteria are quite higher than PE values corresponding to optimal modified lasso. This may be because this complexity parameter selection method is not sufficiently accurate. A more extended experiment, including other described parameters of control, would be necessary to validate these first results.

Table 1. Estimated PE values for the modified lasso and forward selection and PE values for the optimal modified lasso, for each set of parameters.

ρ	ω	Lasso	Forward	Lasso*
0.1	1	0.109	0.081	?
0.1	2	0.346	0.262	?
0.1	3	0.754	0.713	?
0.9	1	0.199	0.172	?
0.9	2	0.907	0.935	?
0.9	3	1.823	2.212	?

Table (2) shows average computing time in seconds. Whereas computing time of lasso does not seem to depend on the number of relevant variables, computing time of forward variable selection increases considerably with the number of variables that actually generate the model.

Table 2. Average computing time in seconds needed by modified lasso and forward selection, when 2, 6 and 10 variables generate the model ($\omega = \{1, 2, 3\}$, respectively).

ω	Lasso	Forward
1	9854	4416
2	8707	26637
3	7771	108440

Forward selection never missed relevant variables. However, this is not the best solution when $\rho = 0.9$, considering that variables are highly correlated. Moreover forward variable selection picked two irrelevant variables in the situations $\omega = 1$, $\rho = 0.9$ and $\omega = 2$, $\rho = 0.9$. Linear components of modified lasso were estimated to be highly penalized in all cases, for all variables (we remind that underlying functions are centered sines). Figure (2) shows inverse of estimated univariate complexity parameters corresponding to the nonlinear components. Modified lasso penalized severely irrelevant variables. Penalization

of relevant variables increased with concurvity and with the number of relevant variables.

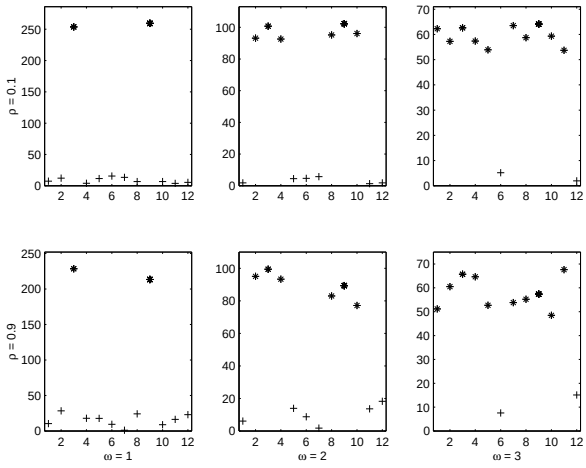

Fig. 2. Inverse of the estimated univariate complexity parameters corresponding to the nonlinear components: $\frac{1}{\lambda_j}$, $j = 1, ..., 12$. All variables are represented ordered from left to right in the horizontal axis. Stars $(*)$ are relevant variables and plus $(+)$ are irrelevant ones. First line of graphics corresponds to low concurvity, second one corresponds to severe concurvity. Columns of graphics correspond to $\omega = 1$, 2 and 3, respectively.

5.3 Conclusion

We propose a modification of lasso for additive models in order to perform variable selection. For each covariate, we differentiate its linear and its nonlinear part, and penalize them independently. Penalization is regulated automatically from two global parameters which are estimated by generalized cross–validation. We have tested this method on a set of problems, in which complexity was very different.

Results of simulations allow us to conclude that lasso perform better than forward selection in the most complex cases. Whereas computing time of lasso does not depend on the number of relevant variables, computing time of forward variable selection increases considerably with the number of variables that actually generate the model. Performance of modified lasso can be got better by improving the complexity parameter selection method.

References

1. S. Bakin. *Adaptive Regression and Model Selection in Data Mining Problems*. PhD thesis, School of Mathematical Sciences, The Australian National University, Canberra, 1999.
2. L. Breiman. Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6):2350–2383, 1996.
3. B. A. Brumback, D. Ruppert, and M. P. Wand. Comment on "Variable selection and function estimation in additive nonparametric regression using a data–based prior" by T. S. Shively, R. Khon and S. Wood. *Journal of the American Statistical Association*, 94(447):794–797, 1999.
4. E. Cantoni and T. J. Hastie. Degrees of freedom tests for smoothing splines. *Biometrika*, 89:251–263, 2002.
5. Z. Chen. Fitting multivariate regression functions by interaction spline models. *J. R. Statist. Soc. B*, 55(2):473–491, 1993.
6. Y. Grandvalet. Least absolute shrinkage is equivalent to quadratic penalization. In L. Niklasson, M. Bodén, and T. Ziemske, editors, *ICANN'98*, volume 1 of *Perspectives in Neural Computing*, pages 201–206. Springer, 1998.
7. Y. Grandvalet and S. Canu. Outcomes of the equivalence of adaptive ridge with least absolute shrinkage. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 445–451. MIT Press, 1998.
8. C. Gu and G. Wahba. Minimizing gcv/gml scores with multiple smoothing parameters via the newton method. *SIAM J. Sci. Statist. Comput.*, 12:383–398, 1991.
9. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research, Special Issue on Variable/Feature Selection*, 3:1157–1182, 2003.
10. T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1990.
11. J. D. Opsomer and D. Ruppert. A fully automated bandwidth selection method for fitting additive models. *J. Multivariate Analysis*, 73:166–179, 1998.
12. M.R. Osborne, B. Presnell, and B.A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.
13. T. S. Shively, R. Khon, and S. Wood. Variable selection and function estimation in additive nonparametric regression using a data–based prior. *Journal of the American Statistical Association*, 94(447):777–806, 1999.
14. R. J. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, B*, 58(1):267–288, 1995.
15. S. N. Wood. Modelling and smoothing parameter estimation with multiple quadratic penalties. *J. R. Statist. Soc. B*, 62(2):413–428, 2000.
16. J. Ye. On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93:120–131, 1998.