Statistical Methods and Applications (2006) 15: 3–26
DOI 10.1007/s10260-006-0005-9

ORIGINAL ARTICLE

**Isabella Morlini**

# On multicollinearity and concurvity in some nonlinear multivariate models

Accepted: 7 December 2005 / Published online: 7 February 2006


**Abstract** Recent developments of multivariate smoothing methods provide a rich collection of feasible models for nonparametric multivariate data analysis. Among the most interpretable are those with smoothed additive terms. Construction of various methods and algorithms for computing the models have been the main concern in literature in this area. Less results are available on the validation of computed fit, instead, and many applications of nonparametric methods end up in computing and comparing the generalized validation error or related indexes. This article reviews the behaviour of some of the best known multivariate nonparametric methods, based on subset selection and on projection, when (exact) collinearity or multicollinearity (near collinearity) is present in the input matrix. It shows the possible aliasing effects in computed fits of some selection methods and explores the properties of the projection spaces reached by projection methods in order to help data analysts to select the best model in case of ill conditioned input matrices. Two simulation studies and a real data set application are presented to illustrate further the effects of collinearity or multicollinearity in the fit.

**Keywords** Additive models · CART · Collinearity · MARS · Multi-layer perceptron · Projection pursuit regression

## 1 Introduction

Although multicollinearity (that is, near collinearity) or collinearity up to numerical precision is rare in most statistical applications, it frequently occurs in exploratory data analysis or in data mining, where many and redundant variables are often included in the modelling phase. On the other hand, it is seldom convenient

I. Morlini
Dipartimento di Scienze Sociali, Cognitive e Quantitative, Universitá di Modena e Reggio Emilia, Viale Allegri 9, 42.100 Reggio Emilia, Italy
E-mail: morlini.isabella@unimore.it

to apply pre-processing strategies like principal component analysis or variable selection, in order to achieve a full rank input matrix. In the first case, it is well known that the axes determined by the first principal components are not necessary the best axes for the successive modelling phase, e.g. classification or regression, since the principal component analysis maximizes an objective function which only depends on the input data and is different from the criteria minimized in classification and regression, which also depend on the target data. For what concerns variable selection, in many applications it is better to use a combination of all the input variables rather then choosing a subset of variables. For example, in quality control, a weighted average of the control variables is preferable to the ones with the highest correlations with the response.

It is the goal of this paper to show that projection methods should be preferred in presence of multicollinearity or collinearity among nonlinear multivariate models which are often applied in data mining and in regression problems when the form of the relationship between a dependent variable and multiple predictors is not known a priori. As a matter of fact, models with smooth *univariate* additive terms (or basis functions) may present instability in the fitting process when the input matrix is bad conditioned. Instability is referred to contributions of variables to the additive model. These contributions become very sensitive to the order of variables and may fluctuate wildly as predictors are added to or removed from the model. Unstable contributions of variables cause difficulty in identifying individual additive terms and impact the interpretation of the additive feature. Projection methods, which are characterized by *multivariate* basis functions, do not share this instability. Coefficients estimated for each additive term, as long as estimation of functions themselves, when not fixed, do not depend on the order of variables. Moreover, both estimates are insensitive to small random error in the dependent variable, avoiding difficulty in identifying basis functions.

As described in Buja et al. (1989) the generalized additive models (GAM) present great instability and arbitrariness in the fitting process when the predictors are collinear since results reached by the backfitting algorithm depend on the order in which variables are presented. In this paper an example is reported in which results differ not only for the coefficients given to each basis functions, but also for variables included in the model, since some variables are given zero degrees of freedom. By means of further data sets it is also illustrated the behaviour of the backfitting algorithm in GAM when multicollinearity is present in the input matrix and outlined the differences between the aliasing effects produced by backfitting in multivariate adaptive regression splines (MARS).

Projection methods like projection pursuit regression (PPR) and the multi-layer perceptron (MLP) are not affected by collinearity since these models first perform a nonlinear transformation from the space of the inputs to a new space, which is the projection step, and then a linear transformation from this new space, which is the modelling step. In Ingrassia (1999) it is demonstrated that in the new coordinate systems, if the nonlinear transformation is a sigmoidal one, the points are uncorrelated even if the original inputs are correlated. In Ingrassia and Morlini (2005) it is proved that the optimal dimension of the projection space (in terms of trade off between bias and variance) may be larger than the dimension of the input space, when the input matrix is collinear or multicollinear. In this paper it is shown that this is not always the case for PPR, where the nonlinear transformation is a

smoother and not a sigmoidal one. The dimension of the projection space depends on the degree of smoothness given to the function and in same instances a projection space larger than a certain dimension cannot be reached by the algorithm.

It is worth noting that GAM have a stronger motivation as data analytic tools than neural networks and PPR, since each variable is represented separately in the mapping function. These models retain an important interpretation feature of linear models: once they are fitted to data, the coordinate functions can be plotted separately to examine the roles of the variables in predicting the response. However, in presence of collinearity or multicollinearity, this interpretation feature may be lost since the nature of the effect of one variable on the response may change depending on the order of variables in the model. In this case, the researcher interested in model and variable selection besides prediction should prefer a fitting algorithm working in a new coordinates system rather than in the original one.

The paper is organized as follows. Section 2 briefly reviews GAM and the backfitting algorithm. Section 3 introduces the concepts of *concurvity* and *approximate concurvity* and outlines theirs link with collinearity and multicollinearity. In section 4 a brief description of MARS is provided and, drawing from the results of De Veaux and Hungar (1994) and Buja et al. (1989) , the different behaviour of the backfitting algorithm in GAM and MARS is shown. Section 5 introduces projection tools like PPR and the MLP and analyses and compares these models in presence of collinearity. The properties of the projections realized by these two methods and the way for determining the *optimal* dimension of the projection space are also investigated. For PPR, the stability of the backfitting algorithm even in presence of collinearity is motivated. Section 6 focuses on two numerical examples while in section 7 the methods are applied on a real data set from satellite images. Section 8 provides concluding remarks.

## 2 Generalized additive models and the backfitting algorithm: a brief review

Drawing from scatterplot smoothers for a response and a single predictor, there are a number of possibilities for estimating the regression surface in the $p$-variate case. Even if the most straightforward is through the use of a $p$-dimensional scatterplot smoother, due to problems like the well-known curse of dimensionality (Friedman and Stuetzle 1981), the metric assumptions to find neighbourhoods in two or more dimension and the very expensive computational requirements, surface smoothing techniques are in practice not very useful in the multivariate setting. Among all nonparametric multivariate approaches aimed at estimating a regression surface, the additive modelling and the projection tools seem to be the most frequently exploited in practice, since they can be easily implemented by using existing software. GAM and the PPR are implemented in the software package S-Plus while neural networks are implemented in a number of dedicated software or toolboxes related to packages like Matlab or SPSS.

In the regression setting, a generalized additive model (Hastie and Tibshirani 1986, 1990) has the form:

$$E(Y|X_1, X_2, \ldots, X_p) = G\left(a + \sum_{j=1}^{p} f_j(X_j)\right),$$

where $Y$ is a univariate response variable, $X_j$ ($j = 1,\ldots, p$) are $p$ explanatory variables, the $f_j$ are unknown univariate smooth functions and $G(\cdot)$ is the inverse of the *link function*. In the following $G(\cdot)$ will be assumed to be the identity function. To avoid nonidentifiable constants in the model it is required that

$$E\big(f_j(X_j)\big) = 0 \quad j = 1, \ldots, p.$$

This implies that $E(Y) = a$ (assuming the identity link function). Many smoothers require a choice of a smoothing parameter: if the parameter is selected by using the $y$-values as, for example, in crossvalidation, then the resultant smoothers are nonlinear. If this parameter is chosen a priori then the resultant smoothers may become linear. The present paper in concerned with the linear smoothers and their use in the backfitting algorithm.

Given observations $\mathbf{x}_i$, $y_i$, ($i = 1, \ldots, n$), a linear smoother can be written as a linear map $\mathbf{S}_j$: $R^n \rightarrow R^n$ defined by $\hat{\mathbf{y}} = \mathbf{S}_j(\mathbf{y})$. Every smoother matrix $\mathbf{S}_j$ ($j$= 1,. . ., $p$) in the additive model depends on the points $x_{ij}$($i$= 1,…, $n$), as well as the particular smoother, but not on $\mathbf{y}$.

If the following criterion $C$, that is a penalized residual sum of squares, and $\lambda_j$ ($j$= 1, . . ., $p$) smoothing parameters are specified for the problem:

$$C(a, f_1, \ldots, f_p) = \sum_{i=1}^{n} \left\{ y_i - a - \sum_{j=1}^{p} f_j(x_{ij}) \right\}^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 \mathrm{d}t_j, \quad (1)$$

each of the function $f_j$ in the additive model is a cubic spline in the component $X_j$ with knots at each of the unique values of $x_{ij}$, $i$= 1,…, $n$. The degrees of freedom of the model depend on the values of the $\lambda_j$.

If other univariate regression smoothing techniques such as, for example, polynomial, natural cubic splines, B-splines, or regression splines are used, the functions $f_j$ become an expansion in basis functions and the criterion minimized is the usual sum of squares error. The additive model is then more interpretable since results in a parametric fit. Once the additive model is fitted to data the $p$ coordinate functions can be plotted separately to examine the roles of the variables in predicting the response. The degrees of freedom of the model are equal to the number of basis functions. Assuming we use $k$ basis functions for each $f_j$ (for example, using a natural cubic spline, $k$ is equal to the number of selected knots plus one, while using a B-spline or a regression spline, $k$ is equal to the number of selected knots plus the degree of the spline), the additive model can be fit directly by solving a system of $kp$ linear equations, without the use of an iterative scheme. For large $k$ and $p$, however, backfitting is considered as a numerically stable alternative to solving a large system of equations and it is the default fitting algorithm for additive models in S-Plus. The backfitting algorithm (Hastie and Tibshirani 1990) is a Gauss–Seidel iterative method which consists of the following step:

1. Set $\hat{a} = \frac{1}{n}\sum_{i=1}^{n} y_i$.
2. Initialize: $\hat{\mathbf{f}}_j = \hat{\mathbf{f}}_j^0 = \mathbf{0}$, $j$= 1, . . ., $p$.
3. Cycle: $j = 1, \ldots, p, 1, \ldots, p, \ldots,$

$$\hat{\mathbf{f}}_j \leftarrow \mathbf{S}_j \left( \mathbf{y} - \mathbf{1}\hat{a} - \sum_{k \neq j} \hat{\mathbf{f}}_k \right)$$

$$\hat{\mathbf{f}}_j \leftarrow \hat{\mathbf{f}}_j - \frac{1}{n} \mathbf{1}' \hat{\mathbf{f}}_j$$

until the individual functions do not change or change less then a pre-specified threshold. It is worth distinguishing between *successive* and *simultaneous* iteration schemes, usually also referred to as Gauss–Seidel and Jacobi iterations, respectively. The first scheme updates one component at a time, based on the most recent components available. In contrast, the Jacobi scheme forms a complete new set of updates from a complete old one. The difference between the two approaches can be formalized as follows:

$$\text{Gauss–Seidel: } \hat{\mathbf{f}}_j^{\text{new}} \leftarrow \mathbf{S}_j \left( \mathbf{y} - \hat{a}\mathbf{1} - \sum_{k<j} \hat{f}_k^{\text{new}} - \sum_{k>j} \hat{f}_k^{\text{old}} \right),$$

$$\text{Jacobi: } \hat{\mathbf{f}}_j^{\text{new}} \leftarrow \mathbf{S}_j \left( \mathbf{y} - \hat{a}\mathbf{1} - \sum_{k \neq j} \hat{\mathbf{f}}_k^{\text{old}} \right).$$

It is the Gauss–Seidel scheme that makes GAM vulnerable to ill conditioning in the input matrix, as it will be shown in the next section. Given the $\mathbf{S}_j$ $n \times n$ smoothing matrices and the $n$-dimensional vectors $\mathbf{f}_j$ ($j = 1, \ldots, p$) with components $f_j(x_{1j}), f_j(x_{2j}), \ldots, f_j(x_{nj})$, the system of normal equations solved by backfitting is:

$$\begin{pmatrix} \mathbf{I} & \mathbf{S}_1 & \mathbf{S}_1 & \cdots & \mathbf{S}_1 \\ \mathbf{S}_2 & \mathbf{I} & \mathbf{S}_2 & \cdots & \mathbf{S}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_p & \mathbf{S}_p & \mathbf{S}_p & \cdots & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_p \end{pmatrix} = \begin{pmatrix} \mathbf{S}_1\mathbf{y} \\ \mathbf{S}_2\mathbf{y} \\ \vdots \\ \mathbf{S}_p\mathbf{y} \end{pmatrix}, \tag{2}$$

where $\mathbf{I}$ is a $n \times n$ unit matrix. If the criterion minimized is the cost function (1), then each $\mathbf{S}_j$ is the appropriate cubic spline smoother matrix, depending on $\lambda_j$. For fixed knots regression splines, B-splines or natural cubic splines, each of the $\mathbf{S}_j$ are orthogonal projectors onto a small subspace of $R^n$ and have the form

$$\mathbf{S}_j = \mathbf{B}_j (\mathbf{B_j}' \mathbf{B}_j)^{-1} \mathbf{B_j}',$$

where the sub design matrices $\mathbf{B}_j$ are generated by the appropriate basis spline functions. Note that in GAM, since the constant term is given by $\hat{a} = (1/n) \sum_{i=1}^{n} y_i$, in each $\mathbf{B}_j'$ is not present the first basis function corresponding to the constant component. Properties of a smoothing matrix are given in Buja et al. (1989) and in Hastie et al. (2001, chapter 5). For polynomials and splines the $\mathbf{S}_j$ are symmetric, positive semidefinite and hence have a real eigen-decomposition. For polynomials, B-splines, regression splines and natural cubic splines, the eigenvalues are 0 or 1 only, with corresponding eigenspaces consisting of the space of residuals and

fits, respectively. For smoothing splines the eigenvalues depend on $\lambda_j$ while the eigenvectors look approximately like polynomials of increasing degree and are not affected by changes in $\lambda_j$. Since in GAM the constant term is separated and each of the smooth terms is adjusted to have zero mean, the first eigenvalue is always one and corresponds to the function linear in $X_j$ which is never shrunk. With respect to the smoother matrix $\mathbf{S}$ in a univariate analysis (here the subscript $j$ is not necessary), in an additive model the smoother is implicitly redefined to $\mathbf{S}_j = \mathbf{S} - \mathbf{1} \times \mathbf{1}'/n$ and $\mathbf{S}_j$ has eigenvalue zero for the vector of constants (while in a one dimensional analysis $\mathbf{S}$ has eigenvalue 1 for the vector of constants). The other eigenvalues are real positive values decreasing from 1 to 0.

## 3 Exact and approximate concurvity in GAM

While the term *collinearity* refers to linear dependencies among predictors which lead to degeneracy in the system of equations in a multiple linear model, the term *exact concurvity* (Buja et al. 1989) has been used to describe – exact – nonlinear dependencies which lead to degeneracy in GAM, that is to the existence of infinite solutions. Formally, for the general smoother-based normal equations, concurvity is defined as the existence of a nonzero solution $\mathbf{g} = (\mathbf{g}_1' \ldots \mathbf{g}_p')'$ of the corresponding homogeneous equations

$$\begin{pmatrix} \mathbf{I} & \mathbf{S}_1 & \mathbf{S}_1 & \cdots & \mathbf{S}_1 \\ \mathbf{S}_2 & \mathbf{I} & \mathbf{S}_2 & \cdots & \mathbf{S}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_p & \mathbf{S}_p & \mathbf{S}_p & \cdots & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_p \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}. \tag{3}$$

If such a $\mathbf{g}$ exists, and if $\mathbf{f} = (\mathbf{f}_1' \ldots \mathbf{f}_p')'$ is a solution of (2), then so is $\mathbf{f}+\beta\mathbf{g}$ for any $\beta$ and thus infinitely many solutions exist. In a more intuitive way, concurvity may be thought as the existence of collinearity between (nonlinear) transforms of predictors. Collinearity between predictors is defined as the existence of a nonzero vector $\mathbf{b} = (b_1 \ldots \mathrm{b}_p)'$ such that $\sum_{j=1}^{p} b_j \mathbf{x}_j = \mathbf{0}$, with $\mathbf{x}_j = (x_{ij} \ldots x_{nj})'$. Similarly, for GAM where each smoother is an orthogonal projection, we can associate with predictor $\mathbf{x}_j$ a linear space $V_j$ of transformations and define concurvity to hold if there exist non trivial $\mathbf{g}_j \in V_j$ such that $\sum_{j=1}^{p} \mathbf{g}_j = \mathbf{0}$.

In the context of exact concurvity, Theorems 2 and 5 in Buja et al. (1989) are of fundamental importance. The first Theorem states that the normal equations (2) are always consistent for polynomials and spline smoothers. The second one states that, for the same class of smoothers, exact concurvity can only occur if there is a linear dependence among the eigenspaces of the $\mathbf{S}_j$ corresponding to eigenvalue +1. So, for these smoothers, collinearity implies exact concurvity:

$$\sum_{j=1}^{p} b_j \mathbf{x}_j = \mathbf{0} \rightarrow \sum_{j=1}^{p} \mathbf{g}_j = \mathbf{0}$$

since polynomials and splines preserve constants and linear fits.

In contrast to the treatment of linear systems in literature, where nondegeneracy is usually assumed and a solution to the equations system cannot be achieve –

unless the input matrix is transformed in a full rank matrix or a generalized inverse is defined – the backfitting algorithm in GAM always converges to a solution. The solution to which the algorithm converges depends, besides the initializations $f_j^0$ ($j$= 1,. . ., $p$), on the order of the input variables. This is worked out explicitly for the two smoothers case, that is $p$=2, in Buja et al. (1989). The dependency of the results on the order of variables is, of course, an undesirable property which leads to an unjustified arbitrariness of the backfitting algorithm in the choice of the solution. What it will be shown by numerical examples in the next section is that the solutions may differ not only in the coefficients of each basis functions but also in the degrees of freedom given to each basis function. If basis function related to a variable $j$ are given zero degrees of freedoms, not all variables are included in the final model and the backfitting algorithm makes a sort of variable selection, the variables being arbitrarily included in the model depending on the order in which they are presented. In order to overcome the problem, the modified algorithm proposed in Buja et al. (1989) should be used. This algorithm extracts the projection parts from the smoothers and, drawing an analogy with the linear regression case, essentially solve the problem of concurvity by reparameterizing the normal equations to obtain a full rank model (Eubank and Speckman 1989).

While exact concurvity is unlikely, but it is predictable for symmetric smoothers with eigenvalues in [0, 1] since it can only be an exact collinearity among the untransformed predictors, approximate concurvity may cause harm too, in that some or all of the estimated basis functions are likely to be unstable. Approximate concurvity is defined as the existence of an approximate minimizer of the penalized least squares criterion which leads to approximate nonlinear additive relations among the predictors (Buja et al. 1989). As multicollinearity in the linear regression setting, approximate concurvity causes difficulties in separating effects in the model with the consequence that the parameter estimates may be poor. However, it does not describe degeneracy in a technical sense (that is, the solutions to system (2) are not infinite and the form of the fit is not fully predictable from the model and the design) unless the following two conditions are satisfied:

1. the $p$ predictors lye exactly on a lower dimensional manifold, for example, a curve for two variables,
2. the additive functions defining the manifold are preserved by the respective smoothers.

For linear smoothers approximate concurvity maybe caused by multicollinearity (or ill conditioning) in the input matrix. If all the $\mathbf{S}_j$ are projectors approximate concurvity causes numerical problems in the backfitting algorithm but do not lead to degeneracy in a technical sense. The result is different when it comes to smoothing splines smoothers, for which the matrices $\mathbf{S}_j$may have a great number of eigenvalues close to 1. For these smoothers approximate concurvity may lead to degeneracy if the above two conditions are satisfied. For all smoothers the risk of numerical problems in the Guass Seidel algorithm and unstable estimates of the basis functions gets higher as long as the mapping function gets more flexible. Approximate concurvity for linear smoothers is the same in spirit as the *prospective cocurvity* defined in Gu (1992): by construction, the decomposition $\sum_j f_j(X_j)$ is well defined on its domain $\mathcal{I}$ ; however, when the additive functions are estimated from the data, information comes from the design points $(\mathcal{I})_0$= [$\mathbf{x}_i$], $i$=1,. . .,$n$, and

observed concurvity occurs when the restriction of the estimated $f_j'$s to $(\mathcal{I})_0$ are nearly collinear. Decomposition is thus not well defined on the restricted domain $(\mathcal{I})_0$. The problem with approximate or observed concurvity is that it is not foreseeable. In order to detect it the model must be checked by carrying out retrospective diagnostics [see Gu (1992)] or by analysing the degrees of freedom of the linear part of the smoothers in the modified algorithm proposed in Buja et al. (1989) Another diagnostic tool proposed by Donnel et al. (1994) is the additive principal components (APCs) analysis of the predictor variables. Smallest ACPs amount to data description in terms of approximate implicit equations: $\sum_j f_j(X_j) \approx 0$ with the smallest variances of $\sum_j f_j$, subject to some normalizing constraint. The minima variances characterization lead to solutions to an eingenproblem that generalizes linear principal components. Eigenvalues $\boldsymbol{l}$= **var**$\sum_j f_j$ measure the strength of additive degeneracy. They are nonnegative and, by definition (see Donnel et al. 1994), below 1. An ACP with zero eigenvalue reveals the presence of exact concurvity in the predictor variables. As long as the approach proposed by Gu (1992) this is a retrospective analysis since the additive functions must be estimated before diagnosing approximate concurvity.

An example of observed concurvity caused by ill conditioning of the input matrix is reported in the second simulation study of section 6. A second example is the real data set application of section 7.

## 4 MARS and collinearity: a comparison with GAM

Regarding the backfitting algorithm used to fit MARS (Friedman 1991) a study by De Veaux et al. (1993) and De Veaux and Hungar (1994) show that the algorithm exhibits problems in choosing among predictors when multicollinearity is present. These problems are somehow different from the dependence of the solutions on the order of variables. Let briefly introduce MARS and the procedure used to fit the model. The model is an expansion in piecewise linear basis functions, of the form $(x_{ij} - t)_+$ and $(t - \text{it } x_{ij})_+$ where

$$(x_{ij}-t)_+ = \begin{cases} x_{ij} - t & \text{if } x_{ij} > t \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad (t-x_{ij})_+ = \begin{cases} t-x_{ij} & \text{if } x_{ij} < t \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$t \in \{x_{1j},\ x_{2j},\ \ldots,\ x_{nj}\}$ is the knot and $j$= 1, . . ., $p$. The model building strategy is like a forward stepwise linear regression using functions (4) and their products. The model has the form:

$$\hat{y} = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(\mathbf{X}) \tag{5}$$

where each $h_m$ is a function in the set of the functions (4) or a product of two or more such functions. So, interaction between variables is explicitly allowed. Given a choice for the $h_m$, the coefficients $\beta_m$ are estimated by minimizing the residual sum of squares, that is, by standard linear regression. The more difficult part, however, is the construction of the functions $h_m$. Starting with the constant $h_0$=1, all the 2 *np* functions (4) and their products are possible candidates at each

stage of the backfitting algorithm. It is added to the model the basis function that produces the largest decrease in a cross-validated criterion or in the penalized residual sum of squares. The process continues until the model contains some preset maximum number of terms. Usually the final model overfits the data and so a backward deletion procedure is applied. As shown in De Veaux and Hungar (1994), if two predictors are both correlated with themselves and with the response, at some stage of the forward selection procedure, MARS may be forced to choose between placing a knot on one of these predictors. The choice may be somewhat arbitrary if both results in roughly the same error or used criterion value. The choice has potentially strong impact on the choice of all further variables and knot selections and thus on the final model as well. This is a result of the tree structure in MARS. In an extreme case it may happen that a much better model would result if the other predictor had been chosen. The backward step, which follows the forward phase and aims to produce a model with comparable performance but fewer terms, is also vulnerable to multicollinearity, especially in the additive case (when no interaction is allowed) since over-fitting is avoided by reducing the number of knots rather than via a smoothness penalty.

Some of the effects of collinearity in the construction of final model are the same in MARS and GAM. For example, only some of the input variables may be represented in the final model and the interpretability of the final model when one correlated predictor is chosen over another, becomes a difficult task. Indeed, instability of the backfitting algorithm is due to its Gauss–Seidel scheme in GAM, and results depend on the order of variables. In MARS, at every stage, all variables are considered, and thus the order of variables has no impact on the final model, but the tree structure makes MARS vulnerable to the subset of variables considered. If, for example, variable $X_1$ is selected in an early stage and a new variable correlated with $X_1$ is introduced in a second analysis, it may happen that this variable is selected instead of $X_1$, since both result in the same value of the criterion minimized. A second difference regards the distinction between collinearity and multicollinearity. This distinction has no impact in MARS since the arbitrariness in the selection process depends on the bivariate correlations rather than on the conditioning of the input matrix (arbitrariness may also occur if there is a single high correlation coefficient in the input correlation matrix). On the other hand, as it will be also shown by the numerical example in section 6 and the real data set application in section 7, this distinction is of great importance in GAM. Collinearity causes concurvity and thus the linear part of the solution to which the backfitting converges to inevitably depend on the order of variables. Multicollinearity may cause concurvity or approximate concurvity. Approximate concurvity always leads to poor parameter estimates but may cause dependency of the results on the order of variables only if smoothing splines smoothers are used

## 5 Projection methods and collinearity

Projection pursuit regression and the MLP belong to the class of methods having the following form

$$\hat{y}_i = \sum_{m=1}^{M} w_m \phi_m(\mathbf{a}'_m \mathbf{x}_i) + w_0$$

where the $\mathbf{a}_m$ ($m = 1, \ldots, M$) are $p$-vectors of unknown parameters, normalized to have unit length, the $w_i$ are unknown coefficients and $\mathbf{x}_i = (x_{i1} \ldots x_{ip})'$. These methods simultaneously project the data in an $M$-dimensional space (when $p$ is large, usually $M$ is far less than $p$) and model the features (which are linear combinations of the inputs) in this new input space. The key difference between them is that the PPR uses nonparametric functions to model the derived features while MLP uses a far simpler sigmoidal function. Like GAM, these are linear expansion of basis functions and thus are additive but in derived features of the input variables and not on the single variables $X_1, \ldots, X_p$. Without loss of generality, in the following we can set $w_0 = 0$.

These models perform the following transformations:

1. a nonlinear transformation from $R^p$ to $R^M$ given by the functions $\phi_m$, that is $\mathbf{x} \rightarrow \phi_1(\mathbf{a}_1'\mathbf{x}_i), \ldots, \phi_M(\mathbf{a}_M'\mathbf{x}_i)$, which is the projection step;
2. a linear transformation from $\mathrm{R}^M$ to $\mathrm{R}^1$ according to $w_1, \ldots, w_M$, which is the modelling step.

In other words, these models operate a linear regression or discrimination on a suitable nonlinear transformation of the input data. The suitability of the non linear transformation is guarantee since the projection step and the modelling phase are faced simultaneously and the dimension $M$ of the projection space and the vectors $\mathbf{a}_m$ are optimized in a supervised manner, according the target values $t$. The mapping function realized by a PPR can be also written in the form:

$$\hat{y}_i = \sum_{m=1}^{M} g_m(\mathbf{a}_m'\mathbf{x}_i) \tag{6}$$

where the functions $g_m$ are estimated along with the directions $\mathbf{a}_m$ using some flexible smoothing method. In S-Plus these functions are running mean smoothers with fixed or cross-validated span values, as suggested in the original paper by Friedman and Stuetzle (1981). Given the training data, the model is fit seeking the approximate minimizer of the error function

$$\sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M} g_m(\mathbf{a}_m'\mathbf{x}_i)\right)^2 \tag{7}$$

over functions $g_m$ and direction vectors $\mathbf{a}_m$, $m = 1,\ldots, M$. Starting from just one term ($M = 1$), given the direction vector $\mathbf{a}_1$, the running mean smoother is applied to the derived variable $z_{im} = \mathbf{a}_m'\mathbf{x}_i$ to obtain an estimate of $g_1$. On the other hand, given $g_1$, we want to minimize the error function (7) over $\mathbf{a}_1$. A Gauss–Newton search is applied for this task, which is a quasi-Newton method where the part of the Hessian involving the second derivatives is discarded (Friedman 1984). These two steps, estimation of $g_1$ and $\mathbf{a}_1$, are iterated until convergence. For the other terms, the model is built in a forward stage-wise manner, adding a pair $(g_m, \mathbf{a}_m')$ at each stage. After each step the $g_m$ from previous step are readjusted using the backfitting procedure. Note that the running mean smoothers are not symmetric and thus collinearity in the matrix $Z = (z_{im})$ of the derived variables does not imply concurvity. The number of terms $M$, which determines the dimension of the projection space, is estimated as part of the forward stage-wise strategy. Having fit a

model with a large number of terms, the model with $M$ components is retained if the next model with ($M$+1) terms does not appreciably have an improved performance. Since the running mean smoother can be expressed as follows

$$g_{im} = w_m \frac{\sum_k z_{km}}{ns},$$
$$k = \max\left(i - \frac{ns-1}{2},\ 1\right), \ldots, i-1,\ i,\ i+1, \ldots,\ \min\left(i + \frac{ns-1}{2},\ n\right)$$

with the value of the span $s$ between 0 and 1, the coefficients $w_m$ determine the importance of each term in predicting the output and can be used in estimating the dimension $M$ as well as the value of the error function (7) at each stage.

The function realized by an MLP with a sigmoidal transfer function in the input layer and a linear transfer function in the hidden layer, is of the form:

$$\hat{y}_i = \sum_{m=1}^{M} w_m \sigma(\mathbf{a}_m' \mathbf{x}_i) \tag{8}$$

where $\sigma(\cdot)$ is a sigmoidal function and the $w_m$ are parameters also called weights. Here the number $M$ of the projection space, that is the number of hidden units, is determined by crossvalidation. The sum of squares error is used as measure of fit and the generic approach to minimize this error is by gradient descent, called backpropagation in the neural network setting. Since backpropagation can be very slow, and for this reason is usually not the method of choice, second order techniques such as Newton's methods are frequently used. Better approaches to fitting also include conjugate methods or variable metric methods which avoid explicit computation of the second derivative matrix while still proving faster convergence. For a description of the training algorithms and a review of the most important issues in training an MLP, like the overfitting problem and the presence of multiple minima, the reader is referred to Bishop (1995) and Ripley (1996).

As stated before, the difference between the two methods is that the PPR model uses nonparametric functions $g_m$ while the MLP uses a far simpler sigmoidal function. In presence of collinearity, the sigmoidal functions have an interesting property. If we consider $n$ linearly dependent points in $R^p$ and a ($M\times$ $p$) matrix $\mathbf{A}$ with values on the hypercube $[-u, u]^{pM}$ for $u$ =1/$M$, then the points $\mathbf{Ax}_1, \ldots, \mathbf{Ax}_n$ are still linearly dependent because they are obtained by a linear transformation on $\mathbf{x}_1, \ldots, \mathbf{x}_n$. If $\sigma$ is a sigmoidal analytic function on $(-r, r)$, with $r$>0, then

$$rank\left(\begin{bmatrix} \sigma(\mathbf{Ax}_1)' \\ \vdots \\ \sigma(\mathbf{Ax}_n)' \end{bmatrix}\right) = M$$

for almost all matrix $\mathbf{A}$ and for $M \leq n$ (Ingrassia (1999); Ingrassia and Morlini 2005). These results show that the projection space in an MLP may be also an over-space of dimension $M$ with $n \geq M > p$ because the points in this overspace are linearly independent and the system:

$$\begin{array}{cccc}
w_1\sigma(\mathbf{a}_1'\mathbf{x}_1) + \cdots + & \cdots & w_n\sigma(\mathbf{a}_n'\mathbf{x}_1) & = y_1 \\
\vdots \quad + \cdots + & \cdots & \vdots & = \vdots \\
w_1\sigma(\mathbf{a}_1'\mathbf{x}_i) + \cdots + & \cdots & w_n\sigma(\mathbf{a}_n'\mathbf{x}_i) & = y_i \\
\vdots \quad + \cdots + & \cdots & \vdots & = \vdots \\
w_1\sigma(\mathbf{a}_1'\mathbf{x}_n) + \cdots + & \cdots & w_n\sigma(\mathbf{a}_n'\mathbf{x}_n) & = y_n
\end{array}$$

has a unique solution. The PPR model implemented in S-Plus may hold the same property, since running means are asymmetric smoothers which may not preserve constants and linear fits. The dependency or independency of points $g_1(\mathbf{Ax}_1),\ldots,$ $g_M$ $(\mathbf{Ax}_n)$ in $R^M$ for $M \leq n$, depends on the value given to the span. For a given span, if $M^*$ is the maximal dimension of the projection space in which the points $g_1(\mathbf{Ax}_1),\ldots,$ $g_{M*}(\mathbf{Ax}_n)$ are linearly independent, and we chose a dimension $M > M^*$, the fitting algorithm set $\mathbf{a}_m' = \mathbf{0}$ for $m = M^*+1,\ldots, M$. So, the backfitting in PPR is applied to the points $g_1(\mathbf{Ax}_1),\ldots,$ $g_{M*}(\mathbf{Ax}_n)$ which are linearly independent and collinearity or multicollinearity in the input matrix affect the dimension of the projection space but not the stability of the estimates reached by the backfitting.

## 6 Numerical examples

### 6.1 Collinear matrix

To understand further how the backfitting algorithm behaves in GAM, when the input matrix is singular, it is useful to look at a synthetic example. Let $\mathbf{p}$ be a $(6\times1)$ dimensional vector and $\mathbf{U}$a $(6 \times 5)$ matrix, such that $\mathbf{p}'\mathbf{p} = 1$, $\mathbf{U}'\mathbf{U} = \mathbf{I}$, $\mathbf{p}'\mathbf{U} = \mathbf{0}$, $\mathbf{UU}' = \mathbf{I} - \mathbf{pp}'$. Let $\mathbf{R}$ be a $(6\times6)$ matrix of rank 2, independent of $\mathbf{U}$, with the first two columns randomly generated from a uniform distribution in (0,1) and the other columns taken as linear combination of these two. We define the $(100\times6)$ matrix $\mathbf{X}$ of the predictor variables, as

$$\mathbf{X} = (\mathbf{zp}' + \mathbf{VU}') \cdot \mathbf{R}$$

with $\mathbf{z}$ of dimension $(100\times1)$ and elements generated from an $N(0,\ 256)$, $\mathbf{V}$ of dimension $(100\times5)$ with the first column generated from an $N(0,\ 49)$, the second one from an $N(0,\ 0.0121)$ and the last three columns from an $N(0,\ 0.005)$. Note that $\mathbf{X}$ is of rank 2. We then define the vector $\mathbf{y}$ of the dependent variable as $\mathbf{y} = \mathbf{z} + \mathbf{e}$, with the elements of $\mathbf{e}$ randomly chosen from an $N(0,\ 0.1225)$. The elements of $\mathbf{z}$, $\mathbf{V}$ and $\mathbf{e}$ are generated independently of each other. The resulting matrix $\mathbf{C}$ of the sizes of correlations between the predictive and the dependent variables is:

$$\mathbf{C} = \begin{bmatrix}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & y \\
x_1 & 1 & & & & & & \\
x_2 & 0.96 & 1 & & & & & \\
x_3 & 0.85 & 0.96 & 1 & & & & \\
x_4 & 0.10 & 0.26 & 0.5 & 1 & & & \\
x_5 & 0.19 & 0.45 & 0.67 & 0.98 & 1 & & \\
x_6 & 0.32 & 0.57 & 0.76 & 0.94 & 0.99 & 1 & \\
y & 0.41 & 0.65 & 0.82 & 0.90 & 0.97 & 0.99 & 1
\end{bmatrix}$$

The peculiarity of this data set is that there are two underlying linear factors that give rise to the input variables. This is clear from the pattern of the high and low correlations in **C** such that the variables in a particular subset have high correlations among themselves but low correlations with all the other variables. Nevertheless only the one dimensional components **z** is important for the prediction of **y** and this component is not a linear function of **X**, in general, because of **R**. Note that the values of the variances in the normal distributions and the parameter of the uniform distribution used to generate the data, as long as the number of patterns (equal to 100) have not a special meaning in this example. The aim is to achieve an $(n \times p)$ input matrix with a rank less then $p$ and a data set such that the number of components relevant for the prediction of the target variable is known. This allows evaluating the ability of projection methods to find the actual dimension of the projection space. It's also important to note that **R** is created independently of **U**. If, for example, we should have set $\mathbf{R}= \mathbf{u}_1\mathbf{u}_1' + \mathbf{u}_2\mathbf{u}_2'$ (where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the first and the second column of **U**,respectively) then $\mathbf{X}=\mathbf{v}_1\mathbf{u}_1' + \mathbf{v}_2\mathbf{u}_2'$ (where $\mathbf{v}_1$ and $\mathbf{v}_2$ are the first and the second column of **V**,respectively). The columns of **X** should have been independent of **z** and **y** impossible to predict by linear or nonlinear functions of the columns of **R**.

Linear projection methods like Principal Components Regression and Partial Least Squares extract two components, as was to be expected. Regarding nonlinear models, results are as follows. For completeness we also report results by classification and regression trees (CART) which are a powerful nonparametric regression tool using recursive partitioning of the space of independent variables, as MARS, but not the backfitting algorithm. Detailed discussion on CART can be found in Breiman et al. (1984), Clark and Pregibon (1992) and Venables and Ripley (1994)

Classification regression trees show stable behaviours: if we consider the full data set and other subsets of variables including $x_5$ and $x_6$, in all cases, given the same parameters choice, they select variables $x_5$ and $x_6$. The model complexity, equal to 8 terminal nodes, still remains unchanged.

Multivariate adaptive regressian splines behaves in a different way. Considering the full data set and the subset of variables $x_2$, $x_3$, $x_5$, $x_6$, restricting the interaction order to 2 and setting the maximum number of basis functions equal to 15 (the default value in the MARS package), we find that at the second knot placement the algorithm has to choose between $x_2$ and $x_3$. In terms of the generalized crossvalidation (GCV) error, these two alternatives are the same and the variable selected depends on the set of predictors used to build the model (Table 1).

The choice between $x_2$ and $x_3$ results in the same GCV error, but in different choices in the subsequent stages of the hierarchy and in the backward stepwise elimination step. Models built have approximately the same fit, but different degrees of freedom and different knots placement (Table 2).

Generalized additive models, given the same parameters, result in different models, depending on the order in which the variables are presented. For example, with smoothing splines with degrees ranging from 1 to 6, for variables presented in theirs original order, that is from $x_1$ to $x_6$, predictors with a parametric degree of freedom are $x_1$ and $x_2$. For variables in the reverse order, that is from $x_6$ to $x_1$, predictors with a parametric degree of freedom are $x_6$ and $x_5$ (Table 3).

The two models, for every degree of the smoothing splines, have approximately the same penalized sum of squares (PSS) error and equal number of total degrees

**Table 1** Forward stepwise knot placement in MARS when all input variables are used and when the subset of variables $x_2$, $x_3$, $x_5$, $x_6$ is used

| All variables | | | | Variables $x_2$, $x_3$, $x_5$, $x_6$ | | | |
|---|---|---|---|---|---|---|---|
| Basis function | GCV error | Variable | Knot | Basis function | GCV error | Variable | Knot |
| 0 | 158.29 | | | 0 | 158.29 | | |
| 1 | 1.770 | $x_6$ | −155.68 | 1 | 1.770 | $x_6$ | −155.68 |
| 3 2 | 0.115 | $x_3$ | −37.920 | 3 2 | 0.115 | $x_2$ | −48.850 |
| 5 4 | 0.123 | $x_4$ | −36.050 | 5 4 | 0.123 | $x_5$ | −10.000 |
| 7 6 | 0.134 | $x_1$ | 73.060 | 6 | 0.134 | $x_3$ | −111.64 |
| 9 8 | 0.144 | $x_3$ | 40.150 | 8 7 | 0.144 | $x_2$ | 34.050 |
| 10 | 0.155 | $x_1$ | −106.56 | 10 9 | 0.153 | $x_5$ | 12.090 |
| 12 11 | 0.166 | $x_1$ | −15.520 | 12 11 | 0.169 | $x_3$ | 2.300 |
| 14 13 | 0.185 | $x_1$ | 35.170 | 14 13 | 0.184 | $x_6$ | 107.420 |
| 15 | 0.208 | $x_1$ | −106.56 | 15 | 0.204 | $x_3$ | −111.64 |

**Table 2** Final model, after backward stepwise elimination, in MARS when all input variables are used and when the subset of variables $x_2$, $x_3$, $x_5$, $x_6$ is used

| All variables | | | | Variables $x_2$, $x_3$, $x_5$, $x_6$ | | | |
|---|---|---|---|---|---|---|---|
| Basis Function | Coefficient | Variable | Knot | Basis function | Coefficient | Variable | Knot |
| 0 | −24.39 | | | 0 | −25.771 | | |
| 1 | 0.148 | $x_6$ | −155.68 | 1 | 0.157 | $x_6$ | −155.68 |
| 2 | 0.036 | $x_3$ | −37.92 | 2 | 0.030 | $x_2$ | −48.85 |
| 3 | −0.042 | $x_3$ | −37.92 | 3 | −0.038 | $x_2$ | −48.85 |

**Table 3** Some results for additive models with smoothing splines with degrees ranging from 1 (linear fit) to 6

| Variables | Variables with a parametric degree of freedom |
|---|---|
| Original order | $x_1$ and $x_2$ |
| Reverse order | $x_5$ and $x_6$ |

**Table 4** Number of basis functions selected by the backfitting algorithm in additive models with cubic B-splines

| Variables | intercept | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|---|
| Three degrees of freedom (zero knots) | | | | | | | |
| Original order | 1 | 3 | 3 | 2 | 1 | 0 | 0 |
| Reverse order | 1 | 0 | 0 | 1 | 2 | 3 | 3 |
| four degrees of freedom (1 knot) | | | | | | | |
| Original order | 1 | 4 | 4 | 3 | 2 | 1 | 1 |
| Reverse order | 1 | 1 | 1 | 2 | 3 | 4 | 4 |
| seven degrees of freedom (4 knots) | | | | | | | |
| Original order | 1 | 7 | 7 | 6 | 5 | 4 | 4 |
| Reverse order | 1 | 4 | 4 | 5 | 6 | 7 | 7 |

of freedom. However, basis functions selected are different and quite arbitrary. The best crossvalidated error is given with one degree of the smoothing splines, that is with the linear fit. The problem of basis functions selection and knots placement in presence of collinearity is more understandable when using B-splines or natural cubic splines, since GAM reduce in a parametric fit. Table 4 shows the number

**Table 5** Number of knots selected by the backfitting algorithm in additive models with natural cubic splines

| Variables | intercept | $x_1$ | $x_2$ | $x$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|---|
| Two degrees of freedom | | | | | | | |
| Original order | 1 | 2 | 2 | 1 | 1 | 1 | 1 |
| Reverse order | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| Three degrees of freedom | | | | | | | |
| Original order | 1 | 3 | 3 | 2 | 2 | 2 | 2 |
| Reverse order | 1 | 2 | 2 | 2 | 2 | 3 | 3 |

of basis functions selected by the backfitting algorithm in GAM with cubic B-splines (degrees = 3) of different degrees of freedom. Table 5 shows the number of selected knots in GAM with natural cubic splines with 2 and three degrees of freedom. Results are carried out using the software S-Plus (see Becker et al. 1988).

For what concern projection methods, having fit a PPR model with a $R^6$ projection space with automatic span selection (that is, with local cross-validation), we obtain the following coefficients: $w_1 = 12.045$, $w_2 = 0.218$, $w_3 = 0.161$, $w_4 = 0.194$, $w_5 = 0.134$, $w_6 = 0.103$, where the last five coefficients are substantially smaller than the first one. Considering that the decrease in then error function (7) from $M = 1$ to $M = 2$ is less than 0.0005, we see that PPR correctly finds an $R^1$ projection space. For different span values set a priori and ranging from 0 to 1 the first coefficient always results substantially greater than the others and thus all models agree on finding an $R^1$ projection space. The maximal dimension of the projection space changes with the span values. For a span = 1 this dimension is 2 (the rank of the input matrix) since the smoother preserves linear fit. For a span equal to zero, it is 100, as for the MLP. Results are different for intermediate span values. For example, with a span equal to 0.8, the maximal dimension is 4 while for a span equal to 0.5 it is 3.

A sigmoidal MLP trained with the conjugate gradient algorithm and with a number of hidden units ranging from 1 to 20 gives the best cross-validated results with 1 hidden unit (that is, an $R^1$ nonlinear projection space).

As for GAM and CART, results for PPR are carried out using the software S-Plus 2000 while for the MLP are carried out with the Matlab 6.1 packages.

It is worth noting that in this section we have not reported the performance of each model in terms of GCV error or related indexes, since the emphasis is on the stability of the algorithm for nonparametric methods and on the ability of finding the actual dimension of the projection space for projection methods, rather than on numerical results.

## 6.2 Multicollinear matrix

This experiment is an example of multicollinearity leading to observed concurvity in the basis functions. Drawing from one of the simulations proposed in Gu (1992), we define a (100×6) matrix $\mathbf{X} = (x_{ij})$ of predictor variables as follows: the first three variables $x_{i1}$, $x_{i2}$, $x_{i3}$, $i$=1,. . ., 100, are generated from a uniform distribution in (0,1); $x_{i4} = x_{i1}^{(1/2)}$, $x_{i5} = x_{i2}^2 + x_{i3}^2$, $x_{i6} = x_{i1}^2 + x_{i2}^2$, $i$=1,. . ., 100. We then define the dependent variable

$$y_i = 10\sin\pi x_{i1} + \exp(3x_{i2}) + 5\cos(2\pi x_{i3}) - x\cos(2\pi x_{i4}) + \exp(x_{i5}) - 5\sin(\pi x_{i6})$$

Once again the coefficients here have no special meaning. The aim is to define a multicollinearity matrix leading to concurvity. The resulting matrix of the size of the correlations between the dependent variable and the predictors is:

$$\mathbf{C} = \begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & y \\ x_1 & 1 & & & & & & \\ x_2 & 0.02 & 1 & & & & & \\ x_3 & 0.01 & 0.06 & 1 & & & & \\ x_4 & 0.97 & 0.01 & 0.05 & 1 & & & \\ x_5 & 0.03 & 0.60 & 0.73 & 0.02 & 1 & & \\ x_6 & 0.71 & 0.64 & 0.08 & 0.68 & 0.35 & 1 & \\ y & 0.01 & 0.71 & 0.11 & 0.08 & 0.67 & 0.45 & 1 \end{bmatrix}$$

Even if the input matrix is of full rank and the bivariate correlations are not all significant ($\alpha = 0.01$), the input matrix is badly conditioned, as measured by the condition number of Belsley (1984, 1991), equal to 49.4. This number is computed as the ratio of the largest singular value to the smallest singular value of the centred and standardized matrix of the predictor variables, with inclusion of the column of ones for the intercept. While the choice of standardizing the input matrix is necessary to prevent the eigenanalysis from being dependent on the units of measure of variables and thus dominated by one or two of the independent variables, the choice of centring is somehow arbitrary. Some authors argue that variables should not be centred since centring makes all independent variables orthogonal to the intercept column and, hence removes any collinearity that involves the intercept. Belsley et al. (1980) and Belsley (1984) argue that this correction for the mean is part of the multiple regression arithmetic and should be taken into account when assessing the collinearity problem. Regarding the interpretation of the condition number, Belsley et al. (1980) suggest that a value around 10 indicates weak dependencies that may be starting to affect the regression estimates. Condition numbers of 30 to 100 indicate moderate to strong dependencies and numbers larger than 100 indicate serious collinearity problems.

Classification and regression trees, with a minimum number of observation before split equal to 5, a minimum node size equal to 10 and a minimum node deviance equal to 0.01 (default parameters choice in S-Plus) select variables $x_2$, $x_3$, $x_5$ and $x_6$ and reach a GCV error equal to 3.12. The same results, in terms of performance power and selection of variables, are reached when the subset of variables $x_2$, $x_3$, $x_5$ and $x_6$ are used as inputs.

Multivariate adaptive regression splines, with the default parameter values, behaves in a similar way: the GCV error is 2.6 and all variables are selected.

Generalized additive models with smoothing splines with a degree ranging from 1 to 6 and with variables presented in the original order and in the reverse order differ in the values of the coefficients given to each basis functions but not in the nonparametric degrees of freedom. Differences in the coefficients gets larger as long as the degrees increase. The PSS error ranges from 2.15 for the linear fit 1° to 0.27 for 6°. With natural cubic splines the behaviour is similar. The number of knots selected by the backfitting algorithm remains unchanged if variables are

**Table 6** Number of basis functions selected by the backfitting algorithm in additive models with cubic B-splines

| Variables | intercept | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|---|
| Three degrees of freedom (zero knots) | | | | | | | |
| original order | 1 | 3 | 3 | 3 | 2 | 2 | 2 |
| reverse order | 1 | 2 | 2 | 3 | 3 | 3 | 3 |
| four degrees of freedom (1 knot) | | | | | | | |
| Original order | 1 | 4 | 4 | 4 | 3 | 3 | 3 |
| Reverse order | 1 | 3 | 3 | 4 | 4 | 4 | 4 |
| six degrees of freedom (3 knots) | | | | | | | |
| Original order | 1 | 6 | 6 | 6 | 5 | 5 | 5 |
| Reverse order | 1 | 5 | 5 | 6 | 6 | 6 | 6 |

presented in the reverse order, while the coefficients appear remarkably different. The PSS error ranges from 0.9 for one degree of freedom to 0.87 for six degrees of freedom. The use of B-spline clearly reveals the presence of observed concurvity since the number of basis functions selected for each variable changes when variables are presented in a different order. Table 6 reports some results. The PSS error ranges from 1.02 for the model with zero knots to 0.05 for the model with three knots.

Note that GAM numerical performances cannot be directly compared with those of MARS and CART since these latter models can be given more flexibility with nondefault parameter values and results for them are crossvalidated.

With a PPR model with a $R^4$ projection space and automatic span selection, we obtain the following coefficients: $w_1 = 7.71$, $w_2 = 1.57$, $w_3 = 0.89$, $w_4 = 0.81$. The last two coefficients are substantially smaller than the first two. Considering that the decrease in then error function (7) from $M = 2$ to $M = 2$ is less than 0.02, we choose an $R^2$ projection space. The PSSE error is equal to 0.14. For different span values set a priori and ranging from 0 to 1 the first two or three coefficients always result substantially greater than the others and thus all models agree on finding an $R^2$ or $R^3$ projection space. The PSSE error for models with a span equal to 0.2, 0.5, 0.6, and 0.8 and a two- or three- dimensional projection space, ranges from 0.14 to 0.3.

For what concerns the maximal dimension of the projection space, this is equal to 2 for a span $=1$ and 66 for a span $= 0.5$. Results are different for intermediate span values.

A sigmoidal MLP trained with the conjugate gradient algorithm and with a number $M$ of hidden units ranging from 1 to 20 gives the best cross-validated result with $M > 3$. Cross-validated sum of squares errors, as long as the averages of the size of the weights between the hidden and the output units (that is, the average of $\sum_{m=1}^{M} |w_m|$ for each of the crossvalidated model) are reported in Table 7. Models with similar crossvalidated errors have similar values of the size of the weights between the hidden and the output units. This observation is corroborated by the value of the correlation between the crossvalidated error and the $\sum_{m=1}^{M} |w_m|$, equal to 0.83. These results confirm the analysis Ingrassia and Morlini (2005) which shows that the generalization performance of an MLP is determined by the sum of the sizes of the weights in the hidden layer rather than by the number of these weights and thus by the number of hidden units.

**Table 7** Results for a MLP with a number of hidden units ranging from 1 to 20

| Number of hidden units | Training error | Crossvalidated error | $\sum_{m=1}^{M} \lvert w_m \rvert$ |
|---|---|---|---|
| 1 | 4.94 | 6.74 | 544.75 |
| 2 | 3.51 | 5.07 | 676.99 |
| 3 | 2.92 | 4.45 | 323.24 |
| 4 | 2.15 | 3.81 | 106.62 |
| 5 | 1.56 | 3.08 | 138.31 |
| 6 | 2.09 | 3.79 | 122.79 |
| 7 | 1.56 | 3.13 | 137.97 |
| 8 | 1.32 | 3.27 | 131.31 |
| 9 | 1.23 | 2.96 | 138.02 |
| 10 | 0.91 | 2.82 | 126.35 |
| 11 | 0.92 | 2.72 | 119.90 |
| 12 | 0.59 | 2.32 | 132.43 |
| 13 | 0.55 | 2.51 | 135.74 |
| 14 | 0.53 | 2.43 | 130.79 |
| 15 | 0.58 | 2.51 | 130.39 |
| 16 | 0.57 | 2.43 | 137.42 |
| 17 | 0.44 | 2.34 | 142.27 |
| 18 | 0.51 | 2.51 | 139.77 |
| 19 | 0.53 | 2.63 | 157.44 |
| 20 | 0.60 | 2.59 | 148.84 |

Numerical results of these two projection methods are not directly comparable, since errors for the PPR are not cross-validated. However, both algorithm show stable performances with respect to the choice of the dimension of the projection space. In particular, the MLP shows similar results for a number $M > 3$ while the PPR shows stable performances for a number $M > 1$ and for different span values.

## 7 Real data set example

In this section the behaviours of MARS, GAM and projection tools in presence of a bad conditioned input matrix are studied by means of a satellite images (*satimage*) data set. This dataset is taken from the ftp anonymous "UCI Repository of Machine Learning Databases and Domain Theories" (ics.uci.edu:pub/machine-learniong-databases). Past usages of this data set are, among others, in Michie et al. (1994), Guerin-Dugue A et al. (1995). The data set was generated from Landsat Multi-Spectral Scanner (MSS) images. One frame of Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and regions of the visible spectrum) and two are in the (near) infra-red. Each pixel is an 8-bity binary word with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about $80 \times 80\, m^2$. Each image contains $2340 \times 3380$ pixels. The *satimage* data set is a tiny sub-area of a scene, consisting of $82 \times 100$ pixels with the binary values converted to ASCII form. Each line in the data set correspond to a $3 \times 3$ square neighbourhood of pixels completely contained within the $82 \times 100$ sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixel in the $3 \times 3$ neighbourhood and a number indicating the classification label of the central pixel. The aim is to predict this

**Table 8** Percentage of total variance and cumulative variance explained by each linear principal component (PC)

| PC | percentage of variance explained | | PC | percentage of variance explained | | PC | percentage of variance explained | | PC | percentage of variance explained | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Total | Cumulative | | Total | Cumulative | | Total | Cumulative | | Total | Cumulative |
| 1 | 70.20 | 70.20 | 10 | 0.70 | 94.52 | 19 | 0.21 | 97.51 | 28 | 0.14 | 99.03 |
| 2 | 7.64 | 77.84 | 11 | 0.53 | 95.05 | 20 | 0.21 | 97.72 | 29 | 0.14 | 99.17 |
| 3 | 5.69 | 83.53 | 12 | 0.51 | 95.56 | 21 | 0.20 | 97.92 | 30 | 0.13 | 99.30 |
| 4 | 3.56 | 87.09 | 13 | 0.38 | 95.94 | 22 | 0.18 | 98.01 | 31 | 0.13 | 99.43 |
| 5 | 1.83 | 88.92 | 14 | 0.34 | 96.28 | 23 | 0.17 | 98.27 | 32 | 0.12 | 99.55 |
| 6 | 1.62 | 90.54 | 15 | 0.32 | 96.60 | 24 | 0.16 | 98.43 | 33 | 0.12 | 99.67 |
| 7 | 1.25 | 91.79 | 16 | 0.26 | 96.86 | 25 | 0.16 | 98.59 | 34 | 0.11 | 99.78 |
| 8 | 1.06 | 92.85 | 17 | 0.23 | 97.09 | 26 | 0.15 | 98.74 | 35 | 0.11 | 99.89 |
| 9 | 0.97 | 93.82 | 18 | 0.21 | 97.30 | 27 | 0.15 | 98.89 | 36 | 0.11 | 100.0 |

classification (the target value) given the multi-spectral values (inputs). The dataset contains 6,435 patterns with 36 predictor variables (4 spectral bands $\times$ 9 pixels in the neighbourhood) plus the class label. The predictors are numerical, in the range 0 – 255 (8 bits). The class label is a code for the following classes: 1 = red soil, 2 = cotton crop, 3 = grey soil, 4 = dump grey soil, 5 = soil with vegetation stubble, 6 = mixture class, 7 = very dump grey soil. Here we consider class 3 and class 4 only, with 1,358 and 626 number of patterns, respectively, and we set the target values equal to 1 for the first class and to 0 for the second one. We split the data in a training set of 1,484 number of patterns and a validation set of 500 number of patterns. Results given in the following are referred to the validation set and are therefore comparable.

Even if the input matrix is of full rank and the bivariate correlations are not all significant ($\alpha = 0.01$), the matrix is badly conditioned, as measured by the condition number, equal to 6,520. The degree of multicollinearity may be also perceived from Table 8, in which are reported the percentage of variance explained by each of the linear principal components.

Classification and regression trees, given the same parameter choice (a minimum number of observations before split equal to 25, a minimum node size equal to 50 and a minimum node deviance equal to 0.1), show stable results when considering the full data set and a subset of 22 variables. This subset is chosen considering all variables actually used in the three construction of the full data set analysis and, for the remaining variables, only one of two eventually highly correlated predictors. The resulting tree is the same in each analysis and variables selected are $x_{13}$, $x_{14}$, $x_{19}$, $x_{22}$, $x_{24}$, $x_{25}$. The percentage of observations correctly classified is equal to 83%.

Multivariate adaptive regressian splines behave in a different manner. If we consider the full data set, restrict the interaction order to 2 and set equal to 10 the minimum number of observations between knots and equal to 36 the maximum number of basis functions, the variables appearing in the final model are $x_{12}$, $x_{13}$, $x_{18}$, $x_{19}$, $x_{22}$, $x_{24}$, $x_{25}$, $x_{27}$, $x_{34}$, $x_{35}$, $x_{36}$. With a subset of 22 variables, suitably chosen as in the CART application, variables used in the final model are $x_8$, $x_{12}$, $x_{13}$, $x_{18}$, $x_{19}$, $x_{24}$, $x_{25}$, $x_{26}$, $x_{27}$, $x_{34}$, $x_{35}$, $x_{36}$. The output from the MARS model (Table 9) shows what has happened in the forward stepwise knot placement (before stepwise

**Table 9** Forward stepwise knot placement in MARS when all input variables are used and when a subset of 22 variables is used

| All variables | | | | Subset of 22 variables | | | |
|---|---|---|---|---|---|---|---|
| Basis Function | GCV Error | Variable | Knot | Basis Function | GCV Error | Variable | Knot |
| 0 | 0.216 | | | 0 | 0.216 | | |
| 2 1 | 0.099 | $x_{19}$ | 109 | 2 1 | 0.099 | $x_{19}$ | 109 |
| 4 3 | 0.088 | $x_{23}$ | 88 | 4 3 | 0.088 | $x_{24}$ | 70 |
| 6 5 | 0.084 | $x_{13}$ | 72 | 6 5 | 0.083 | $x_{13}$ | 70 |
| 8 7 | 0.081 | $x_{18}$ | 100 | 8 7 | 0.080 | $x_{22}$ | 102 |
| 10 9 | 0.078 | $x_{18}$ | 91 | 10 9 | 0.079 | $x_{19}$ | 92 |
| 12 11 | 0.076 | $x_{24}$ | 71 | 12 11 | 0.076 | $x_{18}$ | 99 |
| 14 13 | 0.075 | $x_{27}$ | 110 | 14 13 | 0.075 | $x_{18}$ | 91 |
| 16 15 | 0.074 | $x_{22}$ | 100 | 16 15 | 0.074 | $x_{25}$ | 63 |
| 18 17 | 0.074 | $x_{35}$ | 93 | 18 17 | 0.073 | $x_{12}$ | 72 |
| 20 19 | 0.073 | $x_{12}$ | 72 | 20 19 | 0.072 | $x_{35}$ | 93 |
| 22 21 | 0.072 | $x_{18}$ | 98 | 22 21 | 0.072 | $x_{34}$ | 99 |
| 24 23 | 0.072 | $x_{18}$ | 96 | 24 23 | 0.071 | $x_{12}$ | 83 |
| 26 25 | 0.071 | $x_{12}$ | 86 | 26 25 | 0.071 | $x_{18}$ | 108 |
| 28 27 | 0.071 | $x_{34}$ | 99 | 28 27 | 0.071 | $x_{27}$ | 110 |
| 30 29 | 0.071 | $x_{25}$ | 63 | 30 29 | 0.071 | $x_{36}$ | 74 |
| 32 31 | 0.071 | $x_{36}$ | 73 | 32 31 | 0.071 | $x_{12}$ | 73 |
| 34 33 | 0.071 | $x_{19}$ | 94 | 34 33 | 0.071 | $x_{8}$ | 62 |
| 36 35 | 0.070 | $x_{19}$ | 106 | 35 | 0.071 | $x_{26}$ | 34 |
| | | | | 36 | 0.071 | $x_{6}$ | 37 |

deletion). At the second knot placement, MARS has to choose between placing a knot on variable $x_{23}$ or $x_{24}$. These two alternatives result in the same GCV error, but in different future choices in the tree construction. The final models are the same in terms of goodness of fit (the percentage of correct classification is 91.83% with 36 eligible predictors and 91.68% with 22 eligible predictors), but result in different variable selections.

Generalized additive models, with a logit link function and binomial error, show stable results when using natural cubic, regression, and beta splines. With three and four degrees of freedom, the percentage of correct classification ranges from 87 to 92%.

With smoothing splines smoothers, the unstable contributions of variables to the additive model become stronger as the degree given to the smoothers increases. For example, with 3° and 4° , the parametric and the nonparametric degrees of freedom remain unchanged while some of the coefficients given to the basis functions differ in the analysis of the variables given in theirs original order (from $x_1$ to $x_{36}$) and in the analysis with the predictors in the reverse order (from $x_{36}$ to $x_1$). With more than 4° differences are not only in the values of the coefficients, but also in the nonparametric degrees of freedom given to some of the basis functions. For 8°, some of the differences are reported in Table 10. In this application, all variables are given one parametric degree of freedom, and are therefore included in the final model. However, multicollinearity in the input matrix has caused approximate concurvity in the fit.

The presence of approximate concurvity may be also detected by the diagnostics proposed by Gu (1992) and based on the retrospective linear model $\mathbf{z} = \mathbf{f}_1 + \ldots + \mathbf{f}_p + \mathbf{e}$. The diagnostics, that is the collinearity indices $k_j$ of Stewart (1992)

**Table 10** Some differences in GAM with smoothing splines smoothers with 8° degrees, in the satimage data set

| Variables in the original order (from $x_1$ to $x_{36}$) | | | Variables in the reverse order (from $x_{36}$ to $x_1$) | | |
|---|---|---|---|---|---|
| Variable | Nonparametric degrees of freedom | Coefficient | Variable | Nonparametric degrees of freedom | Coefficient |
| Intercept | | 55.11780 | Intercept | | 53.44784 |
| $x_{18}$ | 6.8 | 0.042066 | $x_{18}$ | 6.7 | 0.039324 |
| $x_{23}$ | 7.0 | −0.038104 | $x_{23}$ | 6.9 | −0.037834 |
| $x_{24}$ | 6.9 | −0.198214 | $x_{24}$ | 6.8 | −0.195613 |
| $x_{29}$ | 6.9 | 0.108150 | $x_{29}$ | 6.8 | 0.108568 |
| $x_{32}$ | 6.8 | 0.014955 | $x_{32}$ | 6.7 | 0.015996 |
| $x_{35}$ | 7.0 | 0.044460 | $x_{35}$ | 6.9 | 0.044025 |

**Table 11** Diagnostics for approximate concurvity in GAM with smoothing splines with 8° in the satimage data set

| | $\mathbf{f}_1$ | $\mathbf{f}_2$ | $\mathbf{f}_3$ | $\mathbf{f}_4$ | $\mathbf{f}_5$ | $\mathbf{f}_6$ | $\mathbf{f}_7$ | $\mathbf{f}_8$ | $\mathbf{f}_9$ | $\mathbf{f}_{10}$ | $\mathbf{f}_{11}$ | $\mathbf{f}_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | 88.6 | 83.4 | 90.2 | 93.6 | 86.2 | 81.9 | 86.0 | 96.1 | 90.7 | 84.8 | 93.7 | 102.5 |
| $\cos(\mathbf{z}, \cdot)$ | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| $\cos(\mathbf{e}, \cdot)$ | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.1 | 0.1 | 0.2 | 0.2 |
| | $\mathbf{f}_{13}$ | $\mathbf{f}_{14}$ | $\mathbf{f}_{15}$ | $\mathbf{f}_{16}$ | $\mathbf{f}_{17}$ | $\mathbf{f}_{18}$ | $\mathbf{f}_{19}$ | $\mathbf{f}_{20}$ | $\mathbf{f}_{21}$ | $\mathbf{f}_{22}$ | $\mathbf{f}_{23}$ | $\mathbf{f}_{24}$ |
| $k$ | 96.7 | 88.5 | 88.4 | 91.4 | 97.7 | 95.7 | 95.3 | 94.2 | 103.8 | 89.2 | 91.1 | 95.7 |
| $\cos(\mathbf{z}, \cdot)$ | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| $\cos(\mathbf{e}, \cdot)$ | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| | $\mathbf{f}_{25}$ | $\mathbf{f}_{26}$ | $\mathbf{f}_{27}$ | $\mathbf{f}_{28}$ | $\mathbf{f}_{29}$ | $\mathbf{f}_{30}$ | $\mathbf{f}_{31}$ | $\mathbf{f}_{32}$ | $\mathbf{f}_{33}$ | $\mathbf{f}_{34}$ | $\mathbf{f}_{35}$ | $\mathbf{f}_{36}$ |
| $k$ | 94.9 | 82.7 | 80.3 | 80.3 | 87.7 | 84.6 | 81.4 | 81.0 | 95.7 | 87.5 | 91.6 | 94.1 |
| $\cos(\mathbf{z}, \cdot)$ | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| $\cos(\mathbf{e}, \cdot)$ | 0.1 | 0.2 | 0.2 | 0.1 | 0.1 | 0.2 | 0.2 | 0.1 | 0.1 | 0.2 | 0.1 | 0.1 |

and the cosines between the estimated additive functions after having removed the constant effect, are reported in Table 11. The approximate concurvity is obvious. As a matter of fact, all the collinearity indices are far from 1 and the additive functions are pairwise almost linearly dependent.

For what concern PPR, the maximum dimension of the projection space (for which the direction vectors $\mathbf{a}_m$ are not zero) depends again on the span value. For example, with a crossvalidated span this dimension is equal to 6, while for a span equal to 0.7 it is 4. The "optimal dimension" of the projection space, however, is equal to 1 or 2. The choice between these two values is somehow subjective. For example, with a span equal to 0.5, the coefficients given to the four components are $w_1 = 0.37$, $w_2 = 0.10$, $w_3 = 0.07$, $w_4 = 0.03$, with a decrease in the error function equal to 13% going from one to two components and equal to 3.6% from two to three components. With a twodimensional projection space and different values of the span, ranging from 0.01 to 0.9, the percentage of correct classification ranges from 86% to 93%.

An MLP trained with the conjugate gradient algorithm and with a number of hidden units ranging from 1 to 30 gives the best percentage of classification with 1 hidden unit (that is, 90.61%). Slightly worse results are reached with 10 and 6 hidden units (88.34% and 88.22%, respectively). In the models with 6 and 10

hidden units, the averages of the size of the weights between the hidden and the output units are equal to 20.595 and to 20.696, respectively. These results once again show that generalization performance of an MLP is determined by the sum of the sizes of the weights in the hidden layer rather than on the number of these weights and thus on the number of hidden units.

The prediction power of all the nonparametric methods here analysed result similar (with the only exception of CART, for which the percentage of correct classification is slightly worse). However, the model selection task, as well as the possibility to reach model parsimony and model interpretation, results quite difficult in MARS and GAM. The meaning of model selection is, in this context, the choice of the model to retain for future prediction (for example, the choice between the two models in Tables 1, 2 and 9 for MARS and the choice among models obtained with variables in the original order and in the reverse order in GAM, see Tables 3, 4, 5 and 6). MARS shows unstable results and different variable selections, while GAM reaches additive functions which have pairwise correlations of the same magnitude. The selection of a subset of variables by a retrospective analysis based on these correlations seems to be a hard task, as long as the comprehension of the nature of the effect of each variable on the results. On the contrary, in this example projection methods reach very parsimonious models, with a one- or two-dimensional projection space, and the model selection task, based on the values of the coefficients given to the components for PPR and on the quantity $\sum_{m=1}^{M} |w_m|$ for the MLP, seems to be much easier.

## 8 Conclusion

This work was motivated by the problem of how to do model selection in a nonparametric model. Prediction oriented model selection is based on the GCV error or related indexes and mathematical convenience, regardless of model parsimony and interpretation. When the predictors are collinear or multicollinear, nonparametric models like GAM and MARS, based on the backfitting algorithm, preserve the prediction power but may loose theirs interpretation features. As a matter of fact, they present great instability with respect to the order of variables or to the subset of variables utilized in the analysis and for this reason they may not be the optimal alternative in model building and model selection. This arbitrariness in the selection process is not shared by linear models, in which the original coordinate system is a meaningful one.

In this paper we have analysed the different behaviour of the backfitting algorithm in GAM, MARS and PPR. We have explained why, in presence of a singular input matrix, the solution to which the backfitting algorithm converges depends on the order of variables in GAM while in MARS it depends on the subset of variables used in the model. We have shown that the distinction between collinearity and multicollinearity in MARS has no impact, since the instability of the backfitting algorithm depends on the bivariate linear correlations rather than on the condition of the input matrix, while in GAM this distinction is of great importance. For what concern projection methods, we have shown that collinearity has no impact on the backfitting algorithm used in PPR and we have analysed some properties of the projection spaces realized by PPR and the MLP. Finally, by the first numerical example we have investigated the ability of PPR and of the MLP to find the

correct dimension of the projection space relevant for prediction of the response data, we have investigated the effect of collinearity in GAM and MARS and compared MARS with CART. With the second numerical study and the real data set application we have shown examples of multicollinearity leading to approximate concurvity and outlined the possible effect of approximate concurvity in GAM with smoothing splines smoothers.

**Acknowledgements** I would like to thank the anonymous referee for their suggestions and very thorough and helpful comments which allowed the author to greatly improve the paper.

## References

Becker RA, Chambers JM, Wilks AR (1988) The new S language: a programming environment for data analysis and graphics. Wadsworth & Brooks, Pacific Grove

Belsley DA (1984) Demeaning conditioning diagnostics through centering (with discussion). Am Stat, 38: 73–77

Belsley DA (1991) Conditioning diagnostics, collinearity and weak data in regression. Wiley, New York

Belsley DA, Kuh E, Welsch RE (1980). Regression Diagnostics: Identifying Influential Data and Sources of Collinearity, Wiley, New York.

Bishop C (1995) Neural networks for pattern recognition. Clarendon , Oxford

Breiman L, Friedman JH, Olshen, RA, Stone CJ (1984) Classification and regression trees, Wadsworth, California

Buja A, Donnel D, Stuetzle W (1986) Additive principal components. Technical Report, Department of Statistics, University of Washington

Buja A, Hastie TJ, Tibshirani R, (1989) Linear smoothers and additive models. Ann Stat, 17: 453–555

Clark L–A. Pregibon D (1992) Tree based models. In: Chambers J.M, Hastie T.J (eds) Statistical models in S. Chapman Hall, New york.

De Veaux RD, Psichogios DC, Hungar LH, (1993) A comparison of two non parametric estimation schemes: MARS and neural networks, Comput Chemi Eng, 17 (8): 819–837

De Veaux RD, Hungar LH, (1994) Multicollinearity: a tale of two nonparametric regressions. In Cheeseman P, Oldford RW (eds) Selecting models from data: AI and statistics VI

Donnel DJ, Buja A, Stuetzle W (1994) Analysis of additive dependencies and concurvity using smallest additive principal components, Ann Stati, 22: 1635–1673

Eubank Speckman (1989) Discussion of "linear smoothers and additive models" by Buja A, Hastie TJ & Tibshirani R. Ann Stat, 17: 525–529

Friedman JH (1984) Classification and multiple response regression trough projection pursuit. Department of Statistics, Stanford University, Report LCM006

Friedman JH, (1991) Multivariate adaptive regression splines, The Annals of Statistics 19, 1–141.

Friedman JH, Stuetzle W (1981). Projection pursuit regression, Journal of the American Statistical Association, 76, 817–823.

Gu C (1992) Diagnostics for nonparametric regression models with additive terms. J Ame Stat Assoc, 87: 1051–1058

Guerin-Dugue A et al (1995). Deliverable R3-B4-P task B4: benchmarks, Technical report, Elena-NervesII "Enhanced learning for evolutive neural architecture" ESPRIT-Basic Research Project Number 6891.

Hastie TJ, Tibshirani R (1986) Generalized additive models, Stat Sci, 1: 297–318

Hastie TJ, Tibshirani R (1990). Generalized additive models. Chapman, London

Hastie TJ, Tibshirani R, Friedman JH (2001) The elements of statistical learning, data mining, inference and prediction, Springer, New York.

Householder AS (1964) The theory of matrices in Numerical Analysis. Dover, New York

Ingrassia S (1999) Geometrical aspects of discrimination by multilayer perceptrons. J Multivar Analy 68: 226–234

Ingrassia S, Morlini I (2005) Neural network modelling for small data sets. Technometrics, 47(3): 297–312

Michie D, Spiegelhalter DJ, Taylor CC, (eds) (1994) Machine learning, neural and statistical classification, Ellis Horwood Series in Artificial Intelligence, UK
Ripley BD, (1996). Pattern recognition and neural networks. Cambridge University Press, Cambridge, UK
Stewart GW (1992) Collinearity and least squares regression. Stat Sci, 2: 68–100
Venables WN, Ripley BD (1994) Modern applied statistics with S-Plus. Springer, Berlin Heidelberg, New York