Simultaneous Selection of Variables and Smoothing Parameters by Genetic Algorithms

Rüdiger Krause[1] and Gerhard Tutz

Department of Statistics,
Ludwig-Maximilians University, Akademiestr.1, 80799 München, Germany

Summary. In additive models the problem of variable selection is strongly linked to the choice of the amount of smoothing used for components that represent metrical variables. Many software packages use separate toolsto solve the different tasks of variable selection and smoothing parameter choice. The combinationof these tools often leads to inappropriate results. In this paper we propose a simultaneous choice of variables and smoothing parameters based on genetic algorithms. Common genetic algorithms have to be modified since inclusion of variables and smoothing have to be coded separately but are linked in the search for optimal solutions. The basic tool for fitting the additive model is the penalized expansion in B-splines.

Keywords

Genetic algorithm, Additive model, Variable selection, Penalized regression splines, B-splines, Improved AIC, BIC.

1 Introduction

The problem of variable selection (Miller (2002)) arises when the relationship between a response variable and a subset of potential explanatory variables is to be modelled, but there is substantial uncertainty about the relevance of the variables. In many statistical applications (e.g. analysis of gene expression data) there are large sets of explanatory variables which contain many redundant or irrelevant variables. Hence these applications depend crucially on approaches of variable selection.

Beside variable selection we are also interested in appropriate estimation of the various terms in an additive model (e.g. Hastie & Tibshirani (1990)). In this paper we choose the approach of using a large number of basis functions with penalization of the coefficients. The danger of overfitting, resulting in wiggly estimated curves, is avoided by introducing a penalty term, characterized by a smoothing parameter λ (Eilers & Marx (1996)). The smoothing parameter controls the influence of the penalty term and hence the smoothness of the estimated function. A large parameter value yields smooth estimates (e.g. $\lambda \to \infty$ leads to a linear estimator). In contrast, a small parameter value yields wiggly estimated curves. To prevent over- respectively underfitting of data accurate choice of the smoothing parameter is essential.

[1] krause@stat.uni-muenchen.de

Many software packages have separate tools for variable selection and smoothing parameter choice which are applied successively. The disadvantage of separate tools can be described in the following way: if the user is interested e.g. in a variable selection smoothing parameters have to be chosen previously. Usually these "roughly" chosen parameters are unchanged during variable selection. Fine tuning of the smoothing parameters is subsequently done by another tool. The problem is that the selection of a variable subset is based on the default smoothing parameters; however other smoothing parameters usually yield different variable subsets. Thus a subsequent choice of smoothing parameters by another software tool can often lead to limited improvements, only. An approach which selects variables and smoothing parameters simultaneously should yield significantly improved results.

To our knowledge no common statistical software program contains a complete automatic procedure which simultaneously selects variables and smoothing parameters without restrictions. Here we propose the simultaneous selection of variables and smoothing parameters based on genetic algorithms.

The paper is organized as follows: in the next section we generally describe the class of additive models and the flexible representation of functions by expansions in B-spline basis functions. Section 3 presents the penalization concept of Eilers & Marx (1996) and adapts it to our problem. In section 4 we introduce the genetic algorithm for simultaneous selection of variables and smoothing parameters. In section 5 the suggested approach is compared to alternative methods proposed in literature. Finally in section 6 our approach is applied to a real dataset, the "rental guide" of Munich.

2 Expansion of Additive Models in Basis Functions

A very popular and flexible approach which assumes a rather weak structure in the predictor space is the additive model discussed in detail by Hastie and Tibshirani (1990). Suppose that we have observations $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, where each $\mathbf{x}_i$ is a vector of p components $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$. Then it is assumed that the response variable y_i depends on $\mathbf{x}_i$ by

$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \quad , \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $f_1, \ldots, f_p$ are unknown smooth functions which have to be estimated. It is obvious that the additive model replaces the problem of estimating a function f of a p-dimensional variable $\mathbf{x}_i$ by one of estimating p separate one-dimensional functions $f_j(x_{ij})$. The advantage of (1) is its potential as a data analytic tool: since each variable is represented separately one can plot the p co-ordinate functions separately and thus evaluate the roles of the single predictors.

The additive model in (1) is easily extended to categorical variables $\mathbf{z}_i = (z_{i1}, \ldots, z_{iq}), i = 1, \ldots, n$, as well as interactions between two (categorical or metrical) variables. Then the additive model has the form

$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \mathbf{z}_i^T \boldsymbol{\alpha}_i + \sum_{r=1}^{p-1} \sum_{s=r+1}^{p} f_{rs}(x_{ir}, x_{is}) + \sum_{k=1}^{q} z_{ik} \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i. \tag{2}$$

The term $\mathbf{z}_i^T \boldsymbol{\alpha}_i$ contains the categorical variables and possible interactions between two categorical variables.

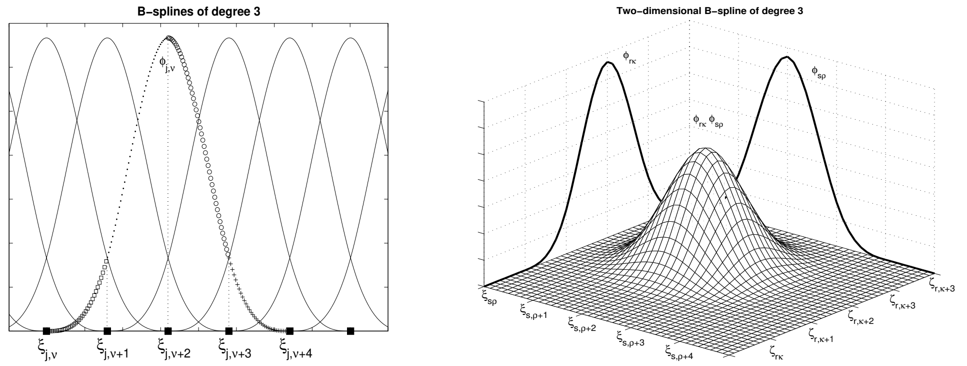

Figure 1. *The left panel shows one-dimensional B-splines of degree 3 respectively order* 4. *The different polynomials of one B-spline are exemplarily plotted. The right panel illustrates a two-dimensional cubic B-spline and the respective one-dimensional B-splines which are comparable with the corresponding ones in the left panel.*

An approach which allows flexible representations of the functions $f_j(x_{ij}), j = 1, \ldots, p$ in (2) is the *expansion in basis functions*, i.e. $f_j(x_{ij})$ is represented by a linear combination

$$f_j(x_{ij}) = \sum_{\nu=1}^{K_j} \beta_{j\nu}\, \phi_{j\nu}(x_{ij}) \tag{3}$$

where $\beta_{j\nu}$ are unknown coefficients and $\{\phi_{j\nu}(x_{ij}), \nu = 1, \ldots, K_j\}$ is a set of basis functions. Each basis function $\phi_{j\nu}(x_{ij})$ is characterized by a knot $\xi_{j\nu}$ from the range of the jth covariate.

As basis functions we use B-splines of degree 3 respectively order 4. A cubic B-spline is generated by four polynomials of degree 3 which are joint at the inner knots. The first and the second derivatives are equal at the joining points (Figure 1). Generally B-splines of degree d have the following general properties:

- B-splines consist of $d + 1$ polynomial pieces, each of degree d;
- they have d inner knots where the polynomial pieces become joined;
- B-splines have an overlap with $2d$ neighboring B-splines. Of course the leftmost and the rightmost B-splines have less overlap;
- at the joining points, derivatives up to order $d - 1$ are continuous;
- B-splines are positive on a domain spanned by $d+2$ knots; outside of this domain the B-spline is zero.

The basis functions $\phi_{j\nu}$ is characterized by one knot only. When using a knot to identify a specific B-spline we take the leftmost knot at which the spline becomes non-zero. For computation of B-splines see e.g. de Boor (1978).

The interaction term $f_{rs}(x_{ir}, x_{is})$ in (2) can also be expanded in B-splines. In this case the two-dimensional function $f_{rs}(x_{ir}, x_{is})$ is represented as a tensor product of two one-dimensional B-splines, i.e.

$$f_{rs}(x_{ir}, x_{is}) = \sum_{\kappa=1}^{K_r} \sum_{\rho=1}^{K_s} \gamma_{rs,\kappa\rho}\, \phi_{r\kappa}(x_{ir}) \phi_{s\rho}(x_{is}) \quad , \tag{4}$$

where the numbers of B-splines K_r, K_s for the two metrical variables can be unequal.Figure 1 shows a two-dimensional cubic B-spline. For illustration we have also plotted the respective one-dimensional B-splines which generate the two-dimensional B-spline. For further details compare de Boor (1978) and Dierckx (1995).

3 Estimation with Penalized Shrinkage

In the case of an additive model without interactions and metrical variables $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, the parameters may be estimated by minimizing the *penalized residual sum of squares (pRSS)* criterion

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \sum_{\nu=1}^{K_j} \beta_{j\nu} \phi_{j\nu}(x_{ij}))^2 + \tau(\{\lambda^x\}) \right\} \quad , \tag{5}$$

where $\beta_{j\nu}, j = 1, \ldots, p, \nu = 1, \ldots, K_j$, are unknown coefficients and K_j is the number of B-splines for the jth covariate. The expression

$$\tau(\{\lambda^x\}) = \sum_{j=1}^{p} \sum_{\nu=k+1}^{K_j} \lambda_{j\nu} (\Delta^k \beta_{j\nu})^2 \tag{6}$$

represents the penalization term. Eilers & Marx (1996) suggested to penalize the difference of adjacent coefficients. Hence in (6) the expression $\Delta^k \beta_{j\nu}, k = 1, 2, \ldots,$ denotes the kth difference, e.g. the second difference has the form $\Delta^2 \beta_{j\nu} = \Delta^1(\beta_{j\nu} - \beta_{j\nu-1}) = (\beta_{j\nu} - 2\beta_{j\nu-1} + \beta_{j\nu-2})$. The parameters $\lambda_{j\nu} \geq 0, \nu = k+1, \ldots, K_j$, with $k = 1, 2, \ldots$, are *local* smoothing parameters that control the amount of shrinkage locally at knot $xi_{\nu,j}$: the larger the values of $\lambda_{j\nu}$, the larger the amount of local shrinkage. If $\lambda_{j,k+1} = \ldots = \lambda_{j,K_j} = \lambda_j$ we have a *global* smoothing parameter for the jth covariate.

Writing (5) in matrix form we obtain

$$pRSS(\mathbf{\Lambda}) = (\mathbf{y} - \mathbf{B}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{B}\boldsymbol{\beta}) + \boldsymbol{\beta}^T \mathbf{D}^T \mathbf{\Lambda} \mathbf{D} \boldsymbol{\beta}, \tag{7}$$

where $\mathbf{B}$ is a $n \times [(K_1 - 1) + \ldots + (K_p - 1)] + 1$-design matrix, $\mathbf{D}$ is a $[(K_1 - k) + \ldots + (K_p - k)] + 1 \times [(K_1 - 1) + \ldots + (K_p - 1)] + 1$- penalization matrix and $\mathbf{\Lambda} = diag(0, \lambda_{1,k+1}, \ldots, \lambda_{1,K_1}, \ldots, \lambda_{p,K_p})$ is a smoothing matrix of dimension $[(K_1 - k) + \ldots + (K_p - k)] + 1 \times [(K_1 - k) + \ldots + (K_p - k)] + 1$. It can be shown (Krause & Tutz (2003)) that the estimator $\hat{\beta}(\mathbf{\Lambda})$ which minimizes (7) has the form

$$\hat{\beta}(\mathbf{\Lambda}) = (\mathbf{B}^T \mathbf{B} + \mathbf{D}^T \mathbf{\Lambda} \mathbf{D})^{-1} \mathbf{B}^T \mathbf{y}. \tag{8}$$

For interactions between two metrical variables x_{ir} and x_{is} a comparable expression for the *pRSS* criterion in matrix form is

$$\begin{aligned} pRSS(\mathbf{\Lambda}_r, \mathbf{\Lambda}_s) = & (\mathbf{y} - \mathbf{B}\gamma)^T (\mathbf{y} - \mathbf{B}\gamma) + \\ & + \frac{1}{2} \underbrace{\gamma^T \mathbf{D}_r^T \mathbf{\Lambda}_r \mathbf{D}_r \gamma}_{\substack{\textit{penalization in} \\ \textit{first direction}}} + \frac{1}{2} \underbrace{\gamma^T \mathbf{D}_s^T \mathbf{\Lambda}_s \mathbf{D}_s \gamma}_{\substack{\textit{penalization in} \\ \textit{second direction}}} \; , \end{aligned} \tag{9}$$

with the penalization term splitting into two parts: the first term yields the penalization of B-splines in the direction of the rth variable. Here $\mathbf{\Lambda}_r$ is a diagonal matrix which has dimension $(K_r - k)K_s \times (K_r - k)K_s$. $\mathbf{B} = diag(\boldsymbol{\phi}_r \otimes \boldsymbol{\phi}_s)$ is a block matrix with tensor products $\boldsymbol{\phi}_r \otimes \boldsymbol{\phi}_s$ of two-dimensional B-splines and have dimension $n \times (K_r K_s - 1)$. The form of the penalization matrices $\mathbf{D}_1$ respectively $\mathbf{D}_2$ is given in Appendix A.

The estimator which minimized (9) is given by

$$\hat{\gamma}(\mathbf{\Lambda}_r, \mathbf{\Lambda}_s) = (\mathbf{B}^T\mathbf{B} + \mathbf{D}_r^T \mathbf{\Lambda}_r \mathbf{D}_r + \mathbf{D}_s^T \mathbf{\Lambda}_s \mathbf{D}_s)^{-1}\mathbf{B}\mathbf{y}. \tag{10}$$

If we have to estimate an additive model, containing metrical and categorical variables respectively diverse interactions, the $pRSS$ criterion can be generally written as

$$pRSS(\mathbf{\Lambda}) = (\mathbf{y} - \mathbf{A}\mathbf{w})^T(\mathbf{y} - \mathbf{A}\mathbf{w}) + \mathbf{w}^T\mathbf{P}^T\mathbf{\Lambda}\mathbf{P}\mathbf{w}. \tag{11}$$

The design matrix $\mathbf{A} = (\mathbf{1}, \mathbf{B}^x, \mathbf{B}^z, \mathbf{B}^{xx}, \mathbf{B}^{zz}, \mathbf{B}^{xz})$ has the form of a block matrix. Here $\mathbf{B}^x$ and $\mathbf{B}^z$ are the design matrices for metrical and categorical variables. $\mathbf{B}^{xx}, \mathbf{B}^{zz}$ and $\mathbf{B}^{xz}$ yield the respective interaction terms. The matrices $\mathbf{B}^z$ and $\mathbf{B}^{zz}$ only contain values 0 or 1. The vector $\hat{\mathbf{w}}$ contains the estimators of weights for the single terms of (2), i.e. $\hat{\mathbf{w}} = (\hat{\beta}_0, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\delta}})^T$. The penalization matrix $\mathbf{P}$ is a block matrix $\mathbf{P} = (0, \mathbf{D}^x, \mathbf{0}^z, (\mathbf{D}_1^{xx}, \mathbf{D}_2^{xx}), \mathbf{0}^{zz}, \mathbf{D}^{xz})$, where $\mathbf{0}^z$ and $\mathbf{0}^{zz}$ are zero matrices for the categorical variables (because they have no penalization terms). $\mathbf{D}_1^{xx}$ and $\mathbf{D}_2^{xx}$ are the penalization matrices (two directions) for the interactions between metrical variables. Finally $\mathbf{\Lambda} = diag(\mathbf{0}, \mathbf{\Lambda}^x, \mathbf{0}^z, \mathbf{\Lambda}^{xx}, \mathbf{0}^{zz}, \mathbf{\Lambda}^{xz})$ is a block matrix, where (similar to the penalization matrix $\mathbf{P}$) $\mathbf{\Lambda}^{xx}$ is splitted into two matrices $\mathbf{\Lambda}_1^{xx}$ and $\mathbf{\Lambda}_2^{xx}$.

The performance of the penalized estimate strongly depends on the choice of the smoothing parameters $\lambda_{j\nu}$. Two common used criteria are the *improved Akaike information criterion (AIC_{imp})* proposed by Hurvich & Simonoff (1998)

$$AIC_{imp} = log\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right] + 2 \cdot \frac{[tr(\mathbf{H}) + 1]}{n - tr(\mathbf{H}) - 2}, \tag{12}$$

and the *Bayesian information criterion (BIC)* of Schwarz (1978)

$$BIC = log\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right] + log(n) \cdot \frac{tr(\mathbf{H})}{n}, \tag{13}$$

where $\mathbf{H} = \mathbf{A}(\mathbf{A}^T\mathbf{A} + \mathbf{P}^T\mathbf{\Lambda}\mathbf{P})^{-1}\mathbf{A}^T$ is the hat matrix. The smoothing parameters have to be chosen such that the criterion becomes minimal. Compared with AIC_{imp} the BIC leads to a stronger penalization.

4 The Concept of Simultaneous Selection with Genetic Algorithms

Genetic Algorithms (Holland (1975), Goldberg (1989)) are originally based on Darwin's evolution theory which refers to the principle that better adapted (fitter) individuals win against their competitors under equal external conditions. Like their biological model, genetic algorithms use biological components (or operators) like selection, crossover, or mutation to model the natural phenomenon of genetic inheritance and Darwinian strife of survival. For some background on the biological processes of genetics and the origin of the terminology see Haupt and Haupt (1998) and Mitchell (1996).

4.1 Operators of a genetic algorithm for variable selection and smoothing parameter choice

The smallest units linked to relevant information of a genetic algorithm are called *genes*. The genes are either single units or short blocks of adjacent units and the information is coded in form of numbers, characters, or other symbols. Usually several genes are arranged in a linear succession which is called *string* (also *chromosome, individual*). The genetic algorithm always uses several strings as a potential solution of an optimization problem. This collection of strings is called *population*. If we apply operators to strings we generate a population with new different strings. This new population of strings is called *offspring*. We denote the particular populations as *generations*, or more precisely as parent- respectively offspring generation.

Accurate coding is of high interest for genetic algorithms. In our case of simultanous selection the strings of the population are a combination of a $0-1$ string δ coding the presence of the diverse variables and a real-valued string λ of smoothing paramters. Suppose we have p metrical variables $\mathbf{x}_1, \ldots, \mathbf{x}_p$ and q categorical variables $\mathbf{z}_1, \ldots, \mathbf{z}_q$. Then the coding of the inclusion of metrical variables is given by

$$\delta_j^x = \begin{cases} 1 & if\ variable\ \mathbf{x}_j\ is\ included \\ 0 & else \end{cases} \qquad j = 1, \ldots, p.$$

In case of categorical variables we have

$$\delta_j^z = \begin{cases} 1 & if\ variable\ \mathbf{z}_j\ is\ included \\ 0 & else \end{cases} \qquad j = 1, \ldots, q.$$

Interactions are coded in the same way by $\delta_{jk}^{xx}, \delta_{jk}^{zz}, \delta_{jk}^{xz}$ and thus for example δ_{jk}^{xx} is given by

$$\delta_{jk}^{xx} = \begin{cases} 1 & if\ the\ interaction\ between\ \mathbf{x}_j\ and\ \mathbf{x}_k\ is\ included \\ 0 & else \end{cases},$$

where $j, k = 1, \ldots, p, j \neq k$. It should be noted that only interactions with $\delta_{jk}^{xx}, j < k$ and $\delta_{jk}^{zz}, j < k$ are used. For interaction between metrical and categorical variables all combinations $\delta_{jk}^{xz}, j = 1, \ldots, p,\ k = 1, \ldots, q$, have to be considered. The indicators may be collected into one string

$$\delta = (\{\delta_j^x\}, \{\delta_j^z\}, \{\delta_{jk}^{xx}\}, \{\delta_{jk}^{zz}\}, \{\delta_{jk}^{xz}\}).$$

Each component of δ can only take the value 0 or 1. For the sake of interpretability hierarchical models are preferred. Thus the model term is restricted by

$$\delta_{jk}^{xx} \leq \delta_j^x \delta_k^x \tag{14}$$

which implies that an interaction can only be included if both variables $\mathbf{x}_j$ and $\mathbf{x}_k$ are included. The same is postulated for categorical variables respectively their interactions with metrical variables.

Each indicator string δ in the population is connected to a smoothing parameter string

$$\lambda = (\{\lambda_j^x\}, \{\lambda_{jk}^{xx}\}, \{\lambda_{jk}^{xz}\}) \ ,$$

which contains the smoothing parameters for the corresponding variables. $\{\lambda_j^x\}, j = 1, \dots, p$, describes the set of smoothing parameters corresponding to the metrical variables (without interactions) $\mathbf{x}_1, \dots, \mathbf{x}_p$. In case of interactions we have similar expressions $\{\lambda_{jk}^{xx}\}$ and $\{\lambda_{jk}^{xz}\}$. In contrast to δ, λ only contains three elements. Since categorical variables as well as their interactions are not connected to smoothing parameters which have to be optimized. In the following the combined string is denoted as (δ, λ).

For the design of a powerful genetic algorithm operators like crossover and mutation are important. Many authors (e.g. Oliveira, Benahmed, Sabourin, Bortolozzi & Suen (2001); Wallet, Marchette, Solka & Wegman (1996); Yang & Honavar (1997)) use operators which have constant influence during the whole application of the genetic algorithm. However, better results are obtained if different aspects of the search are differently weighted at various times: first we are generally interested in exploring the search space and aquire information about the nature of the space. Later we try to obtain information near the global optimum by utilizing the local possibilities of upgrade. Therefore diverse adaptive and non-uniform operators. In context with variable selection we use the following two operators

(i) *Adaptive binary crossover (ABC)* operator: suppose we have two $0-1$ strings with indicator variables $\delta = (\delta_1 \dots \delta_i \dots \delta_k)$ and $\bar{\delta} = (\bar{\delta}_1 \dots \bar{\delta}_i \dots \bar{\delta}_k)$. A pair of bits $(\delta_i, \bar{\delta}_i)$ of the parent strings swap their places if for a random number r_i from $r \in [0,1]$

$$r_i < p_{cv}(\underbrace{1 - r^{(1-\frac{t}{T})^b}}_{\equiv g(t)}). \tag{15}$$

Here $r \in [0,1]$ is a uniform random number which does not depend on the component, p_{cv} is the crossover probability (of the variables), t is the number of the current generation, T is the maximum number of generations and b is a user-dependent system parameter which determines the degree of non-uniformity. Which strings are selected for crossover process is controlled by a similar expression as (15).

In contrast to the conventional crossover operators the ABC operator considers the diverse objectives which have different relevance during the application of the genetic algorithm. We can distinguish between two extreme cases (compare also Figure 2): if t is small the exponent of r is close to zero and hence $g(t)$ is primarily influenced by a suitable choice of the random number r. Figure 2 illustrates this context: for generation number $t = 5$ we receive approximatively a straight line with slope -1. As random number r is uniformly distributed each value $g(t)$ can be (approximatively) accepted with the same probability. If we have chosen $p_{cv} = 1$, diverse strings show many swaps of corresponding bits (if r is small) during crossover process. By suitable choice of p_{cv} the number of swaps between corrsponding bits can be varified. A small value of p_{cv} also decreases the number of swaps between two strings (e.g. a decrease of 0.5 reduces the number of swaps by a half during crossover process).

If the generation number t is large, the exponent of $g(t)$ is close to zero and hence $g(t)$ also yields values close to zero for a wide range of random numbers r. This fact is illustrated in Figure 2 for $t = 95$. At the end of the genetic algorithms' application there are only a few swaps between corresponding bits. In addition a decrease of p_{cv} increases the effect.

(ii) *Adaptive binary mutation (ABM)* operator: for each bit of a string we generate a random number r_i and if

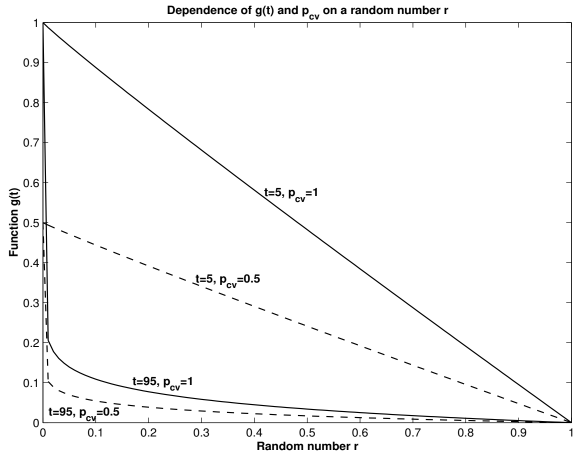

Figure 2. Here function $g(t)$ is shown subject to a uniformly distributed random number r for two sizes of the generation number t and crossover probabilities $p_{cv} = 1$ respectively $p_{cv} = 0.5$. The user-dependent system parameter b is chosen as 1.

$$r_i < p_m v(1 - r^{(1-\frac{t}{T})^b}) \tag{16}$$

holds, the bits mutate, i.e. 0 is changed to 1 and vice versa. Here p_{mv} is the mutation probability (of the variables). The idea and functionality of this operator are the same as described for the ABC operator.

Here we only introduce the binary operators. For optimization of a real-valued string λ of smoothing parameters we use the *modified improved arithmetical crossover (modIAC)* operator and the *non-uniform mutation* operator as described in Krause & Tutz (2004).

4.2 Structure of the combined genetic algorithm

In order to receive a genetic algorithm for simultaneous selection of variables and smoothing parameters the tools and operators from section 4.1 have to be combined appropriately into a selection procedure.

Generally an indicator string δ contains of elements with values 0 and 1, i.e. only some variables (expressed by value 1) are contained in the string. Hence for the smoothing parameter string λ smoothing parameters are used only in case the corresponding indicator takes value 1. In the genetic algorithm presented here in case of $\delta_j^x = 0$ the smoothing parameter λ_j^x is not chosen as 0 but retains the value of the former string. This has the advantage: if the indicator variable again changes from 0 to 1 (e.g. by mutation) the respective smoothing parameter has not been randomly selected which generally leads to results far away from any optima. Instead the actual smoothing parameter which is already determined in former iterations should be close to an optimum. Further application of the genetic algorithm tries to find more fit offsprings which are more close to the optimum. Thus we do not permanently have to explore the whole search space for better solutions.

In the simultaneous genetic algorithm the mutation operators for selection of variables respectively smoothing parameters are linked. In the mutation procedure first randomly chosen elements of the indicator string δ are mutated by use of the adaptive binary mutation (ABM). Then the non-uniform mutation operator is applied to the elements of the smoothing parameters λ which correspond to the mutated elements of δ. The rest of the smoothing parameters remains unchanged. Different from mutation the crossover operators (ABC and IAC) run simultaneously but independently from each other. Simulation trials have shown that it is favourable to use different crossover rates for variables and parameters. Here the crossover rate for the variables is lower than that for smoothing parameters, i.e. the number of crossover processes for variables is lower.

As selection procedure we use a modification of the stochastic universal sampling method (Baker (1985)), called *modified selection procedure (modSP)*. This procedure consists of 9 steps and is illustrated in Figure 3:

Step 1: *In iteration step t population $P(t)$ of $m = r + s$ strings (δ, λ) is generated by selecting from the previous population. Then the worst u percent strings of $P(t)$.*

Step 2: *From the remaining strings of step 1 randomly r strings (δ, λ) are selected. These strings do not necessarily have to be distinct.*

Step 3: *From the remaining strings of step 1 randomly select $s = m - r$ parent strings (δ, λ) are selected. These have not to be distinct from the r selected strings in step 2.*

Step 4: *If identical strings are in the population (i.e. all genes of the strings are identical) the copies will be mutated by using the ABM operator on the indicator strings δ. How many genes of a string are randomly selected and mutated is controlled by a random number (at least one gene is mutated). After mutation there are r different indicator strings. This operation is also executed for the s parent strings.*

Step 5: *Check of the restriction $\delta^{xx}_{jk} \leq \delta^x_j \delta^x_k$ (respectively their equipollent for categorical variables) and deletion of illegal interactions.*

Step 6: *The non-uniform mutation operator is applied to copies of parameter strings λ which correspond to the indicator strings δ. Here only smoothing parameters are mutated for which the value is 1. How many genes of a string are randomly selected and mutated is controlled by a random number (at least one gene is mutated). After mutation, there are r different parameter strings. This operation will also be executed for the s parent strings.*

Step 7: *The ABC operator is applied to the r indicator strings δ and thus generate r indicator offsprings. Apply the modIAC operator to the r parameter strings λ simultaneously and thus generate r parameter offsprings. Both crossover operators run independently.*

Step 8: *Check of the restriction $\delta^{xx}_{jk} \leq \delta^x_j \delta^x_k$ (respectively their equipollent for categorical variables) and deletion of illegal interactions.*

Step 9: *Let r offsprings and s parent strings form the new population $P(t+1)$. Hence their are again $r+s$ indicator strings and $r+s$ parameter strings in the new population $(\tilde{\delta}, \tilde{\lambda})$.*

The selection in step $2, 3, 4, 6$ and 7 is implemented with respect to a probability distribution based on the strings' fitness (see e.g. Michalewicz (1996) for further details).

5 Simulation Study

In this section we present two simulations which base on additive models containing different numbers of components:

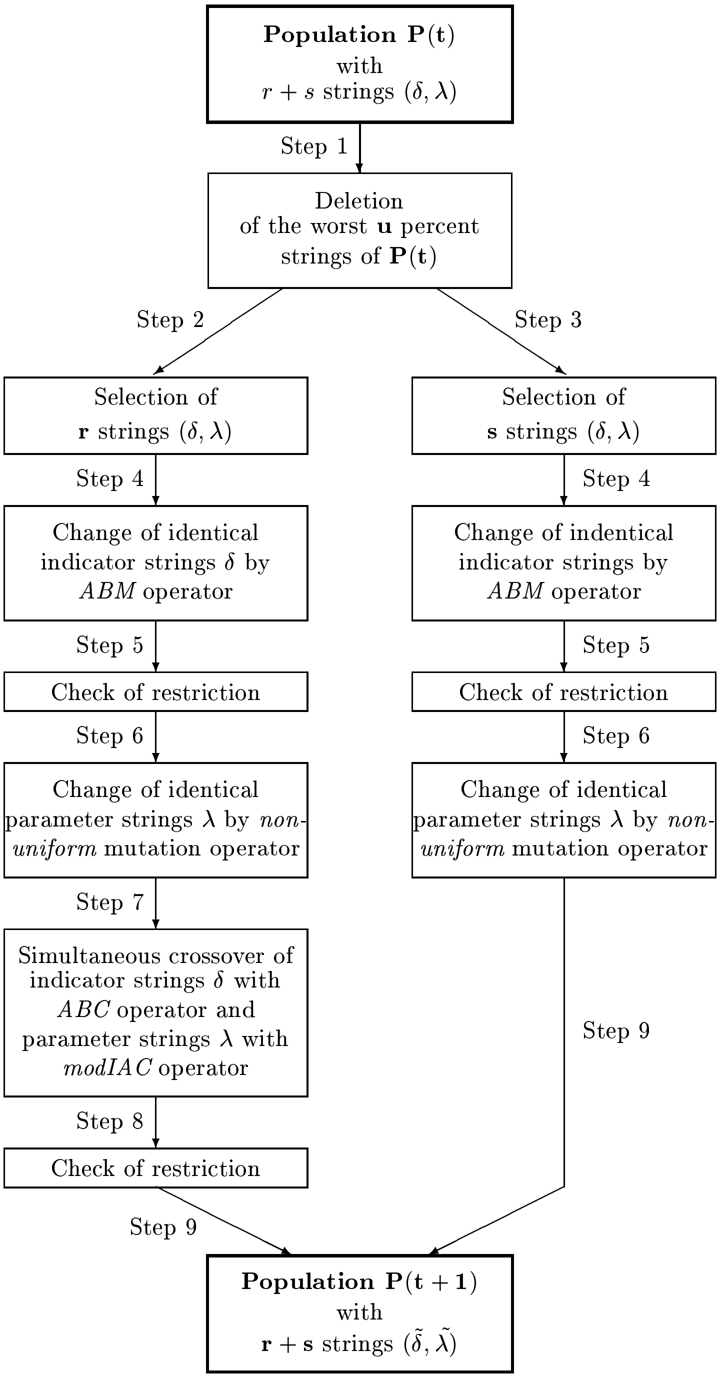

Figure 3. *The flowchart shows the structure of the modified selection procedure (modSP) which has been adapted to the problem of simultaneous selection of variables and smoothing parameters. Details in the text.*

(i) In the first simulation we use an additive model containing 18 different components: 10 functions $f_j(x_{ij}), j = 1, \ldots, 10$, depend on metrical covariates where 5 functions (see Figure 4) have no effect, i.e. $f_j(x_{ij}) = 0$. Furthermore 8 functions $f_j(z_{ij}), j = 11 \ldots, 18$, depend on binary covariates, where 5 functions have no effect. The default parameters of the genetic algorithm used are: *popsize* $= 38$

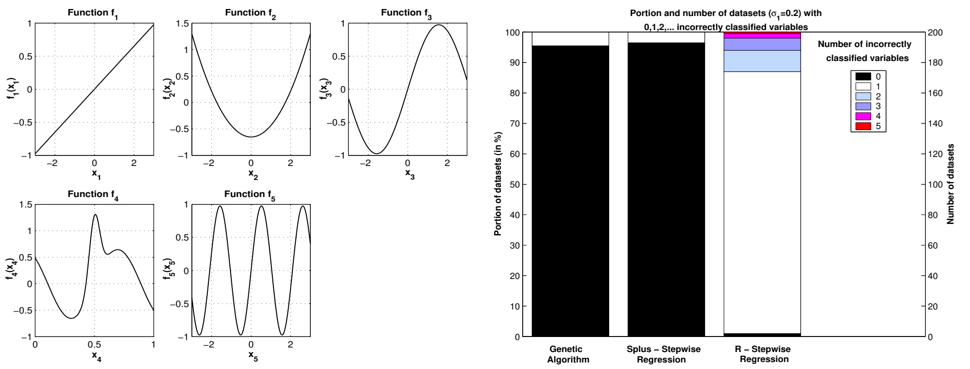

Figure 4. *The left panel shows the five original functions for metrical variables with effect. For the simulation of the additive model with $n = 200$ observations and noise of $\sigma = 0.2$ the right panel yields the number (respectively portion) of datasets with incorrectly specified variables.*

strings, crossover probability (of the variables) $p_{cv} = 0.25$, crossover probability (of the parameters) $p_c = 0.5$, mutation probability (of the variables) $p_{mv} = 0.1$, deletion of $u = 60$ percent of the worst strings, selection of $r = 28$ and $s = 10$ strings, $\nu = 0.5$, $T = 1000$, and $b = 1$. As information criterion we use BIC.

(ii) In the second simulation we analyse an additive model consisting of 4 metrical and 4 categorical variables. Furthermore we have 6 interactions between metrical variables as well as 6 interactions between categorical variables. Altogether 8 variables respectively interactions have an effect (hence the other variables and interactions are without any effect). As default parameters we have chosen: $popsize = 32$, $p_{cv} = 0.25$, $p_c = 0.5$, $p_{mv} = 0.5$, $u = 60$, $r = 22$, $s = 10$, $\nu = 0.5$, $T = 1000$, and $b = 1$. As information criterion we use the improved AIC.

In all cases we simulate 200 datasets each one consisting of 200 independently and uniformly distributed observations with noise of $\sigma = 0.2$. For estimation the single functions $f_j(x_{ij})$ are expanded in 20 (first simulation) respectively 15 (second simulation) one-dimensional cubic B-splines. For the interactions terms $f_{rs}(x_{ir}, x_{is})$ we choose two-dimensional cubic B-splines on a grid of 10 by 10 knots. The smoothing parameters are chosen from the interval $[10^{-4}, 10^4]$ and the penalty is of third difference order.

To compare the performance of the genetic algorithm to alternative approaches we have chosen software tools implemented in S-Plus and R:

- The software package S-Plus offers a restricted possibility of variable selection and simultaneous function estimation. First one calculates AIC for an initial model which contains each covariate as a linear term. Then one has to specify a list with other modelling alternatives. Each covariate can be dropped or integrated in the model as a linear term or as a cubic smoothing spline with a specified smoothing parameter. Starting with the initial model the implemented function `step` successively calculates the AIC for all alternative models. If a current model yields a better AIC the previous model is replaced. Because of its implementation S-Plus can only run a relatively small number of different models. In the simulation each covariate is modelled linearly or as a cubic smoothing spline with degrees of freedom $df = 2, 6, 10, 14$. In case of the interaction terms $f_{rs}(x_{ir}, x_{is})$ the S-Plus procedure uses locally weighted regression smoothers with parameters automatically chosen (for further details see the manual of S-Plus).

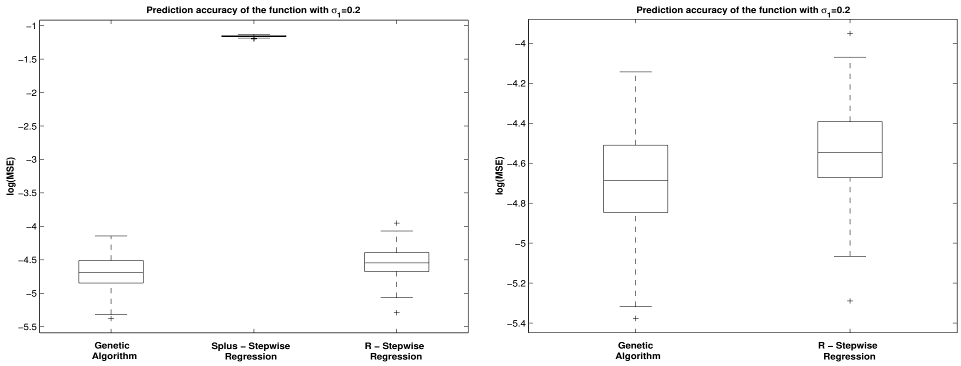

Figure 5. *These two panels show the prediction accuracy of the approaches for the simulation of the additive model. In the panel to the right the S-Plus approach is left out.*

- The software package R offers the following approach to variable selection. The function `stepAIC` implemented in the package `MASS` chooses a model by AIC in a stepwise algorithm. This procedure is comparable with the `step`-function in S-Plus. But in R each covariate can be dropped or integrated in the current model as a linear term or as a polynomial up to degree 4. The user has the possibility to choose BIC as criterion. In the simulations below we have applied AIC and BIC. After variable selection the R-package `mgcv` (Wood (2000)) yields an automatic smoothing parameter selection based on a method first proposed by Gu & Wahba (1991).

In case of the first simulation Figure 4 shows the number respectively portion of datasets with incorrectly specified variables (where incorrectly means the variable has an effect but is not chosen and vice versa). While the genetic algorithm and the stepwise procedure in S-Plus yield comparable results in variable selection the stepwise procedure in R leads to significantly worse error rates: 88% of the datasets have at most one misclassified variable and only 1% of the datasets are completely correctly classified. The stepwise procedure in R also contains datasets with up to five misclassified variables. It should be noted that in case of all three approaches in each dataset all variables with effect are correctly classified. Thus the errors occur in the variables without effect, i.e. more variables than necessary are included in the model.

The two panels of Figure 5 show the prediction accuracy of the diverse approaches. The left panel illustrates all three methods, i.e. the genetic algorithm and the stepwise procedures. The right panel is restricted to the genetic algorithm and the stepwise procedure in R. It is obvious that the worst performance is found for the S-Plus approach. This result depends on the limited choice of the models. Genetic algorithm and stepwise procedure in R lead to comparable estimations. However the right panel of Figure 5 shows that the genetic algorithm outperforms the procedure in R.

In case of the second simulation Figure 6 shows the results for the additive model with interactions between metrical respectively categorical variables. In this simulation the S-Plus procedure also yields one misclassified variable in each dataset. It should be noticed that in this context the expression variable includes main effect variables and interactions. In approximately 35% of the datasets the genetic algorithm shows no misclassified variable and in only 15% of the datasets we have more than 2 incorrectly classified variables. The procedure in R generates significantly worse results, because approximately 60% of the datasets have more than 2 mis-

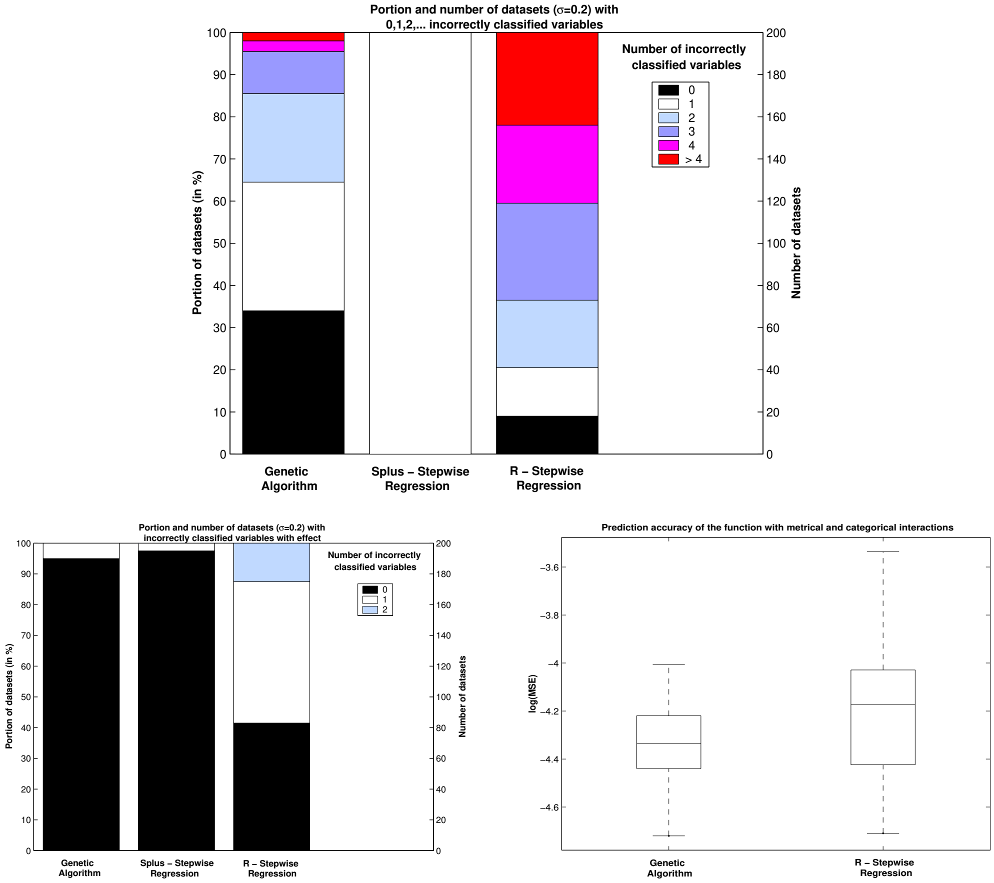

Figure 6. *The simulation results of the additive model with interactions ($n = 200$ observations, noise of $\sigma = 0.2$) are shown here. The top panel yields the number (respectively portion) of datasets with incorrectly specified variables. The left panel below yields the respective results of incorrectly classified variables with effect . The right panel below shows the prediction accuracy of the genetic algorithm and the stepwise procedure in R.*

classified variables. Furthermore in only 8% of the datasets the stepwise procedure in R leads to completely correctly classified datasets.

The left panel of Figure 6 below yields the number respectively portion of datasets with incorrectly classified variables with effect. We realize comparable results for genetic algorithm and S-Plus procedure. Hence for these two approaches the errors essentially occur in the variables without effect, i.e. more variables than necessary are included in the model. In case of the stepwise procedure in R significantly more errors occur for variables with effect: only 40% datasets yield completely correctly classified variables with effect.

The right panel of Figure 6 below shows the prediction accuracy for the genetic algorithm and the procedure in R. Because S-Plus leads to worse prediction accuracy we restrict ourselves to a comparison between genetic algorithm and stepwise procedure in R. Again we realize a significant difference between the two approaches. Furthermore the genetic algorithm shows a smaller variance in the estimations compared with the stepwise procedure in R.

6 Application of the Simultaneous Genetic Algorithm to Rents for Flats

In the last years many large cities have published "rental guides" assisting renter respectively owner of flats to calculate their rents. Furthermore, according to the German rental law, owners are only allowed to increase the rents in dependence on "average rents" of comparable flats. To generally determine "average rents" several thousends of owners and renters are randomly chosen and interviewed in reference to the special equipment of the flat (e.g. bath equipment, kitchen, quality of heating or warm water system). Using further informations like e.g. rent, location of the flat or year of construction we have the possibility of determine the "average rent" (after specification of the respective criteria of the flat).

As basis of the statistical analysis in this chapter we have a random sample of 2055 flats from the census of the rental guide of the year 1998 in Munich. As response variable we choose

$y_i \equiv$ monthly net rent per square meters in Euro (this is calculated by the difference between the monthly rent and the estimated utility costs),

where $i = 1, \dots, 2055$. Out of the approximately 200 variables of the original sample we use 3 metrical variables $x_{ij}, j = 1 \dots, 3$ and 7 categorical variables $z_{ij}, j = 1, \dots, 7$, as described in the Table 1. In context with this dataset we assume an additive model

$$y_i = \beta_0 + \sum_{j=1}^{3} f_j(x_{ij}) + \mathbf{z}_i^T \boldsymbol{\alpha}_i + \sum_{r=1}^{2} \sum_{s=r+1}^{3} f_{rs}(x_{ir}, x_{is}) + \epsilon_i \quad ,$$

where $\epsilon_i, i = 1, \dots, 2055$, is independently and normally distributed.

To receive a selection of necessary variables respectively simultaneous estimation of the dataset we use the genetic algorithm for simultaneous selection of variables and parameters. Here the default parameters of the genetic algorithm are chosen as: $popsize = 32$, $p_{cv} = 0.25$, $p_c = 0.5$, $p_{mv} = 0.5$, $u = 60$,$r = 22$, $s = 10$, $\nu = 0.5$, $T = 1000$, and $b = 1$. The main effects of the 3 metrical variables are modelled by cubic B-splines with 20 knots; for the respective 3 interactions between metrical variables we choose two-dimensional cubic B-splines on a grid of 10 by 10 knots. In both cases the penalty is of third difference order. As model selection criterion we use improved AIC and BIC.

Variable	Brief description	Scale
x_{i1}	floor space (in square meters)	metrical
x_{i2}	year of construction	metrical
x_{i3}	term of tenancy (in months)	metrical
z_{i1}	good location	binary
z_{i2}	best location	binary
z_{i3}	simple warm water supply	binary
z_{i4}	no warm water supply	binary
z_{i5}	no central heating	binary
z_{i6}	special auxiliary equipment in the bath	binary
z_{i7}	bath not tiled	binary

Table 1. *Used variables in the real dataset which basis on the rental guide of Munich (1998).*

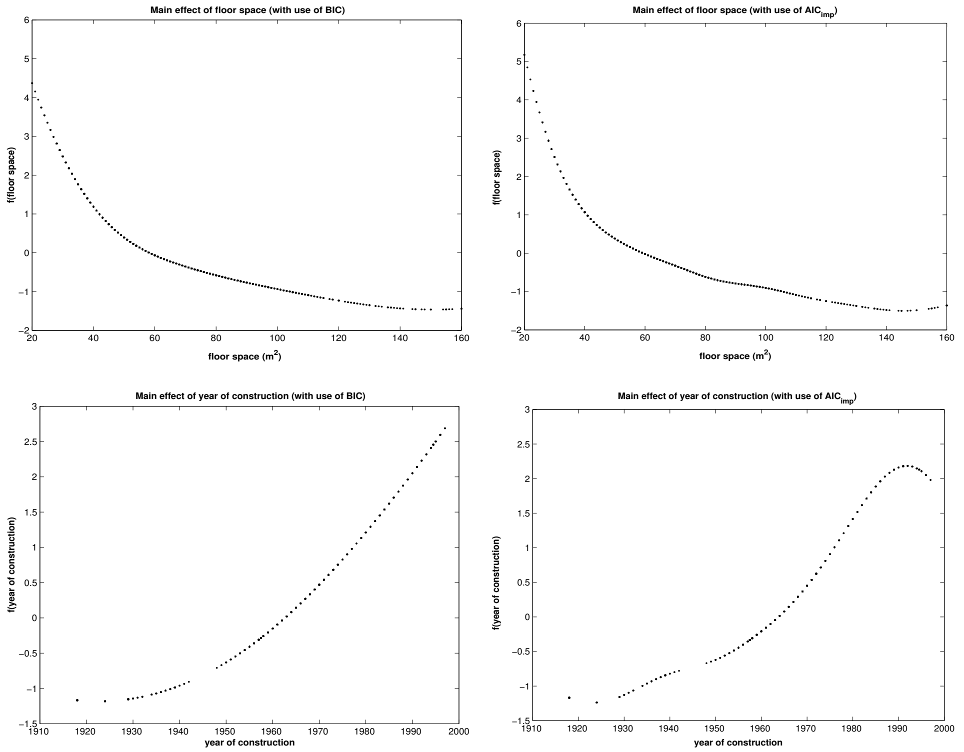

Figure 7. *Here the estimations of the main effects "floor space" and "year of construction" are shown. The plots on the left side yield the results for the genetic algorithm with BIC; the plots on the right side show the respective results for the genetic algorithm including the improved AIC.*

Application of the genetic algorithm with improved AIC (respectively BIC) yields the following results: variables $x_{ij}, j = 1, \ldots, 3$ and $z_{ij}, j = 1, \ldots, 4, 6, 7$ are contained in the model in both cases. Hence the categorical variable z_{i5} and the interactions $x_{i1} \cdot x_{i3}$ respectively $x_{i2} \cdot x_{i3}$ are not included in the model. However in case of the interaction term between x_{i1} and x_{i2} the two criteria lead to a different result: only the genetic algorithm with improved AIC selects this interaction.

Figure 7 shows the effects of "floor space" and "year of construction" for the genetic algorithm with BIC respectively improved AIC. We realize that the main effect "floor space" shows comparable curves for both criteria: small flats are more expensive than larger ones but this effect becomes smaller with increasing floor space.

In case of the main effect "year of construction" the genetic algorithm also yields similar estimations for the two information criteria. The effect on the rents increases with more modern flats. Compared with flats before 1960 the effect on the rent is significantly larger for flats which have been built after 1960. With a year of construction later than 1990 there is a difference for flats between the respective curves of the genetic algorithm with BIC and improved AIC: while BIC yields further increase of the rent the improved AIC stabilizes the effect on the high level (respectively even decreases). This little difference in the curves is given by the stronger penalization of the BIC. Comparable results can also be found in Lang & Brezger (2003).

In Figure 8 we illustrate the effect of interaction between "floor space" and "year of construction" which is included in the model by application of the genetic algorithm

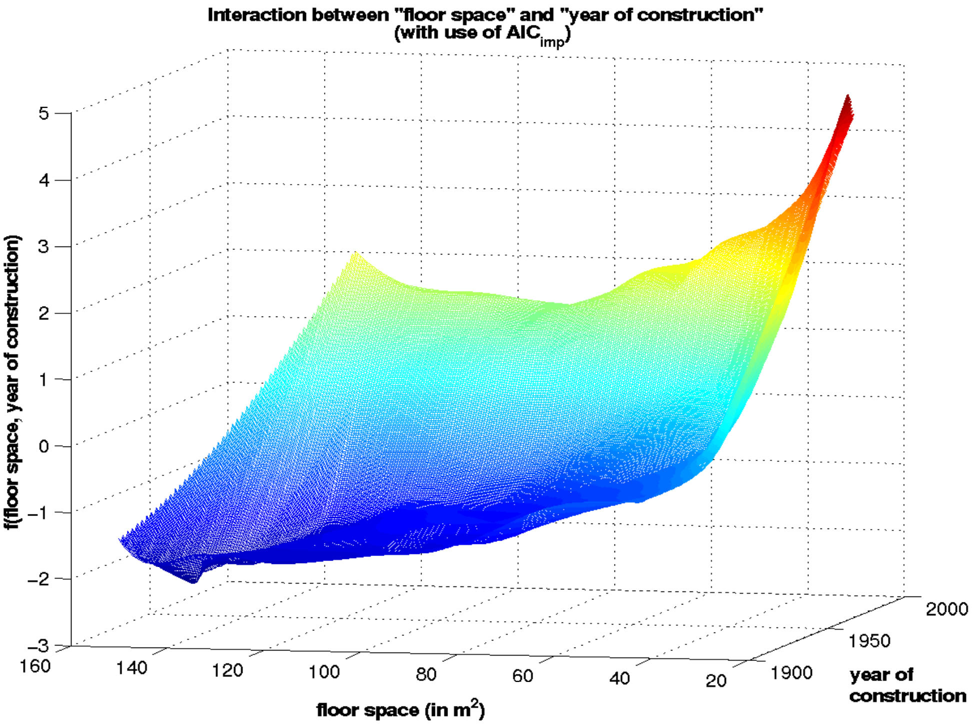

Figure 8. *Here is shown the effect of interaction between "floor space" and "year of construction" which is included in the model by application of the genetic algorithm with improved AIC.*

with improved AIC. The plot shows that the monthly net rent per square meters has a significant dependence on floor space. However the monthly net rent depends slightly on the year of construction, only. Because the effect "year of construction" is relatively small we can understand that the BIC with its stronger penalization has not included this interaction in the model.

From the plot we realize that old flats built before 1940 with a floor space below $50m^2$ are cheaper than the average. Otherwise modern flats built in the year 1970 and later are more expansive than the average. The maximal rent have to be paid for small ($50m^2$) respectively modern (year of construction: 1992) flat; otherwise large flats ($160m^2$) built before World War II are the cheapest ones.

7 Conclusions

In this paper we have presented an approachfor the simultaneous selection of variables and parameters. The approach is based on a combination of genetic algorithms for continous and binary parmeters. In section 5 the genetic algorithm is applied to different additive models for which a simultaneous selection of variables and smoothing parameters is executed. Compared with packages in the statistic programs S-Plus and R in all simulations the genetic algorithm shows distinctly better results with respect to the error rate of the selected variables as well as prediction accuracy.

For function estimation we followed the concept of penalization of regression splines. Here each function is expanded in a generally large number of basis functions (in our case we use B-splines as basis functions). A possible overfit is prevented by a penalization term. Another possibility is the estimation of functions by adaptive selection of knots and hence the use of respective genetic algorithms for knot selection. First approaches in this direction are published by Pittman (2002). In this context a genetic algorithm for simultaneous selection of variables and knots could be an interesting alternative to the concept presented in section 4.

A Penalized Regression Splines with Interactions

The matrices $\mathbf{D}_1 \equiv diag(\mathbf{D}_{r,k})$ and $\mathbf{D}_2 \equiv diag(\mathbf{D}_{s,k})$ with $r = 1, \ldots, p-1, s = r+1, \ldots, p, r \neq s$ in formula (9) are diagonal matrices. Here k yields the difference order and r respectively s symbolise the two directions of the metrical variables x_{ir} and x_{is}. The penalization matrices $\mathbf{D}_{r,1}$ and $\mathbf{D}_{s,1}$ for differences of first $(k = 1)$ order can be generally written as

$$\mathbf{D}_{r,1} = \left[\begin{array}{ccccccc} & \tilde{\mathbf{D}}_{r-1} \otimes \mathbf{I}_s & & & \vdots & \mathbf{0}_{[(K_r-2)K_s \times (K_r-1)]} \\ \hdashline \mathbf{0}_{[(K_s-1)\times(K_r-2)K_s]} & \vdots & -\mathbf{I}_{s-1} & \vdots\ \mathbf{0}_{[(K_s-1)\times 1]} & \vdots & \mathbf{I}_{s-1} \\ \hdashline & -\mathbf{1}^T_{[1\times(K_r-1)K_s]} - \mathbf{e}^T_{[1\times(K_r-1)K_s]} & & & \vdots & -\mathbf{1}^T_{[1\times(K_s-1)]} \end{array}\right],$$

$$\mathbf{D}_{s,1} = \left[\begin{array}{ccc} \mathbf{I}_{r-1} \otimes \tilde{\mathbf{D}}_s & \vdots & \mathbf{0}_{[(K_r-1)(K_s-1)\times(K_s-1)]} \\ \hdashline \mathbf{0}_{[(K_s-2)\times(K_r-1)K_s]} & \vdots & \tilde{\mathbf{D}}_{s-1} \\ \hdashline -\mathbf{1}^T_{[1\times(K_r-1)K_s]} & \vdots & -\mathbf{1}^T_{[1\times(K_s-1)]} - \mathbf{e}^T_{[1\times(K_s-1)]} \end{array}\right].$$

Here $\mathbf{I}_j$ denotes the $j \times j$ identity matrix and $-\mathbf{e}^T_{[1\times j]} = (0, \ldots, 0, 1)$ is a unit vector of length j with 1 at position $(1, j)$. The matrices $\tilde{\mathbf{D}}_r$ and $\tilde{\mathbf{D}}_s$ have dimensions $(K_r - 1) \times K_r$ respectively $(K_s - 1) \times K_s$ and have the form

$$\tilde{\mathbf{D}}_r = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \ldots\ldots & & -1 & 1 \end{bmatrix} \qquad \tilde{\mathbf{D}}_s = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \ldots\ldots & & -1 & 1 \end{bmatrix}.$$

Using the matrices $\mathbf{D}_{r,1}$ and $\mathbf{D}_{s,1}$ we can also specify penalization matrices for differences of order $1 \leq k \leq min\{K_r, K_s\} - 1$, in fact

$$\mathbf{D}_{r,k} = \left[(\tilde{\mathbf{D}}_{r-k+1} \cdot \tilde{\mathbf{D}}_{r-k+2} \cdot \ldots \cdot \tilde{\mathbf{D}}_{r-1}) \otimes \mathbf{I}_s\right] \cdot \mathbf{D}_{r,1}$$

which has dimension $(K_r - k)K_s \times (K_r K_s - 1)$ and

$$\mathbf{D}_{s,k} = \left[\mathbf{I}_r \otimes (\tilde{\mathbf{D}}_{s-k+1} \cdot \tilde{\mathbf{D}}_{s-k+2} \cdot \ldots \cdot \tilde{\mathbf{D}}_{s-1})\right] \cdot \mathbf{D}_{s,1}$$

with dimension $K_r(K_s - k) \times (K_r K_s - 1)$.

References

Baker, J. (1985). Adaptive selection methods for genetic algorithm. In J. Greffenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms*, pp. 101–111. Hillsdale, NJ: Lawrence Erlbaum Associates.

de Boor, C. (1978). *A Practical Guide to Splines*. New York, Heidelberg, Berlin: Springer.

Dierckx, P. (1995). *Curve and Surface Fitting with Splines*. Oxford: Clarendon Press.

Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with b-splines and penalties. *Stat. Science 11*(2), 89–121.

Gu, C. and Wahba, G. (1991). Minimizing gcv/gml scores with multiple smoothing parameters via the newton methods. *SIAM Journal of Scientific and Statistical Computing* **12**, 383–398.

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

Hastie, T., Tibshirani, R. J., and Friedman, J. (2001). *The Elements of Statistical Learning*. New York: Springer.

Hurvich, C. and Simonoff, J. (1998). Smoothing parameter selection in nonparametric regression using an improved akaike information criterion. *Journal of the Royal Statistical Society B 60*(2), 271–293.

Krause, R. and Tutz, G. (2003). Additive modelling with penalized regression splines and genetic algorithms. *Discussion Paper 312, SFB 386, University of Munich*.

Krause, R. and Tutz, G. (2004). Simultaneous selection of variables and smoothing parameters in additive models. In D. Baier & K.-D. Wernecke (Eds.), *Proceedings of the 27th Annual GfKl Conference, University of Cottbus*. Heidelberg-Berlin: Springer-Verlag (In preparation).

Lang, S. and Brezger, A. (2003). Bayesian p-splines. *Journal of Computational and Graphical Statistics*, to appear.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Heidelberg: Springer.

Miller, A. (2002). *Subset Selection in Regression*. Boca Raton, London, New York: Chapman & Hall/CRC.

Oliveira, L. S., Benahmed, N., Sabourin, R., Bortolozzi, F., and Suen, C. Y. (2001). Feature subset selection using genetic algorithms for handwritten digit recognition. In *Proceedings of the 14^{th} Brazilian Symposium on Computer Graphics and Image Processing*, pp. 362–369. Floriano'polis-Brazil: IEEE Computer Society.

Pittman, J. (2002). Adaptive splines and genetic algorithms. *Journal of Computational and Graphical Statistics 11*(3), 1–24.

Schwarz, G. (1978). Estimating the dimensions of a model. *Annals of Statistics* **6**, 461–464.

Wallet, B. C., Marchette, D. J., Solka, J. L., and Wegman, E. J. (1996). A genetic algorithm for best subset selection in linear regression. In *Proceedings of the 28th Symposium on the Interface.*

Wood, S. (2000). Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society B 62*(2), 413–428.

Yang, J. and Honavar, V. (1997). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems* **13**, 44–49.