Boosted Partial Least-Squares Regression

Jean-François Durand

Montpellier II University, France

E-Mail: jf.durand@club-internet.fr

Web site: www.jf-durand-pls.com

I. Introduction

- Machine Learning versus Data Mining
- The data mining prediction process
- Partial Least-Squares boosted by splines

II. Short introduction to splines

- Few words on smoothing splines
- Regression splines
    - Two sets of basis functions
    - Bivariate regression splines
    - Least-Squares Splines
    - Penalized Splines

III. Multivariate Partial Least Squares Splines

- What is $\boldsymbol{L}_2$ Boosting?
- Ordinary PLS viewed as a $\boldsymbol{L}_2$ Boost algorithm
- The linear PLS regression: algorithm and model
- The building-model stage: choosing M
- PLS Splines (PLSS): a main effects additive model
    - The PLSS model
    - Choosing the tuning parameters
    - Multi-collinearity and outliers: the orange juice data
    - Nonlinear 2-D component scatterplots
- MAPLSS to capture interactions
    - The ANOVA type model for main effects and interactions
    - The building-model stage
    - Comparison between MAPLSS, MARS and BRUTO on simulated data
    - Multi-collinearity and bivariate interaction: the ”chem” data

IV. References

I. Introduction

Machine Learning versus Data mining

Machine Learning: machine learning is concerned with the design and development of algorithms and techniques that allow computers to "learn". The major focus of machine learning research is to extract information from data automatically, by computational and statistical methods. [http://www.wikipedia.org/]

Data mining: Data mining has been defined as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" and "the science of extracting useful information from large data sets or databases." [http://www.wikipedia.org/]

To be franc, I do not like very much the word "Machine Learning". I do prefer "Data Mining" that suggests helping humans to learn rather than machines... More concretely, many automatic black-box methods involve thresholds for decision rules whose values may be more or less well understood and controlled by the lazy user who mostly accepts the default values proposed by the author. In the R-package, called PLSS for Partial Least-Squares Splines, I tried to make the automatic part, inescapable to face with the huge amount of data and computation, easy to master by on-line conversational controls.

The data mining prediction process

The timing of the data mining prediction process to be followed with PLSS functions, can be split up into 3 steps:

1. Set up the aims of the problem and the associated schedule conditions.
2. The building-model phase: a 2.1-2.2 round-trip until obtaining a validated model.
   2.1 Build an evolutionary training data base following the retained schedule conditions.
   2.2 Process the regression and validate or not the model built on the data at hand.
3. Elaborate a scenario of prediction. A scenario allows the user to conveniently enter new real or fictive observations to test the validated models.

Partial Least-Squares boosted by splines

Partial Least-Squares regression [14, S. Wold et al.], in short PLS, may be viewed as a repeated Least-Squares fit of residuals from regressions on latent variables that are linear compromises of the predictors and of maximum covariance with the responses. This method is presented here in the framework of L_2 boosting methods [7, J.H. Friedman] by considering PLS components as the base learner.

Historically very popular in chemistry and now in many scientific domains, Partial Least-Squares regression of responses Y on predictors X, $PLS(X, Y)$, produces linear models and has been recently extended to ANOVA style decomposition models called PLS Splines, PLSS, and Multivariate Additive PLS Splines, MAPLSS, [3], [4], [10].

The key point of this nonlinear approach was inspired by the book of A. Gifi [8] who replaced in exploratory data analysis methods, the design matrix X by the super-coding matrix B from transforming the variables by B-splines. Let B_i be the coding of predictor i by B-splines, PLSS produces main effects additive models

$$PLSS(X, Y) \equiv PLS(B, Y)$$

where $B = [B_1| \dots |B_p]$, while capturing main effects plus bivariate interactions leads to

$$MAPLSS(X, Y) \equiv PLS(B, Y)$$

where $B = [B_1| \dots |B_p \,\|\, \dots |B_{i,i'}| \dots]$, $B_{i,i'}$ being the tensor product of splines for the two predictors i and i'.

The aim of this course is twofold, first to detail the theory of that way of boosting PLS that involves regression splines in the base learner, second to present real and simulated examples treated by the free PLSS package available at

http://www.jf-durand-pls.com .

II. Short introduction to splines

Few words on smoothing splines

Consider the signal plus noise model

$$y_i = s(x_i) + \varepsilon_i, \quad i = 1 \ldots n, \quad x_1 < \ldots < x_n \in [0,1]$$

$(\varepsilon_1 \ldots, \varepsilon_n)' \sim N(0, \sigma^2 I_{n \times n})$, σ^2 is unknown and $s \in W_2^m[0,1]$

$W_2^m[0,1] = \{s/s^{(l)}\ l = 1, \ldots, m-1$ are absolutely continuous, and $s^{(m)} \in L_2[0,1]\}$.

The smoothing spline estimator of s is

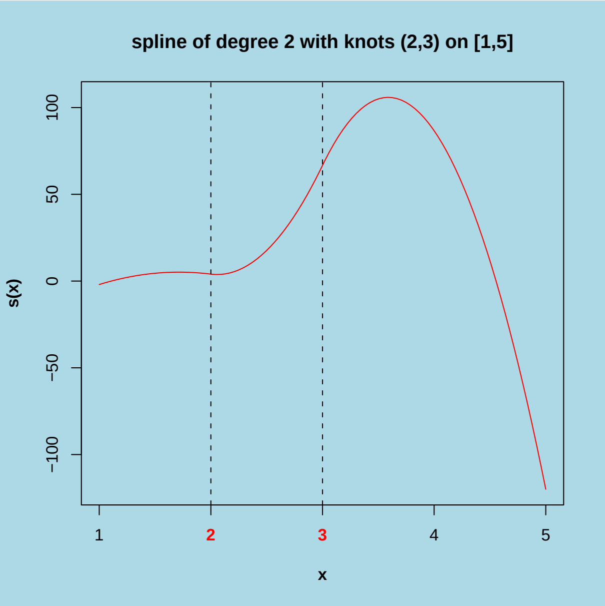
$$s_\lambda = arg \min_{\mu \in W_2^m[0,1]} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu(x_i))^2 + \lambda \int_0^1 [\mu^{(m)}(t)]^2 dt;\ \lambda > 0$$

usually $m = 2$, to penalize convexity and overfitting.

The solution s_λ is unique and belongs to the space of *univariate natural spline functions* of degree $2m-1$ (usually 3) with *knots* at distinct data points $x_1 < \ldots < x_n$.

Now is time to tell something about what splines are!

To transform a continuous variable x whose values range within $[a, b]$, a spline function s is made of adjacent polynomials of degree d that join end to end at points called "the knots", with continuity conditions for the derivatives [1, De Boor].



Two kinds of splines that mainly differ by the way they are used:

- Smoothing splines: the degree is fixed to 3 and knots are located at distinct data points, the tuning parameter λ is a positive number that controls the smoothness.
- Regression splines: few knots $\{\tau_j\}_j$ whose number ad location constitute, joined to the degree, the tuning parameters. Splines are computed according to a regression model.

Regression splines

A spline belongs to a functional linear space

$$S(m, \{\tau_{m+1}, \ldots, \tau_{m+K}\}, [a, b])$$

of dimension $m + K$ characterized by three tuning parameters

- the degree d or the order $m = d + 1$ of the polynomials,
- the number K and
- the location of knots $\{\tau_{m+1}, \ldots, \tau_{m+K}\}$

$$\tau_1 = \ldots = \tau_m = a < \tau_{m+1} \leq \ldots \leq \tau_{m+K} < b = \tau_{m+K+1} = \ldots = \tau_{2m+K}.$$

A spline $s \in S(m, \{\tau_{m+1}, \ldots, \tau_{m+K}\}, [a, b])$ can be written

$$s(x) = \sum_{i=1}^{m+K} \beta_i B_i^m(x)$$

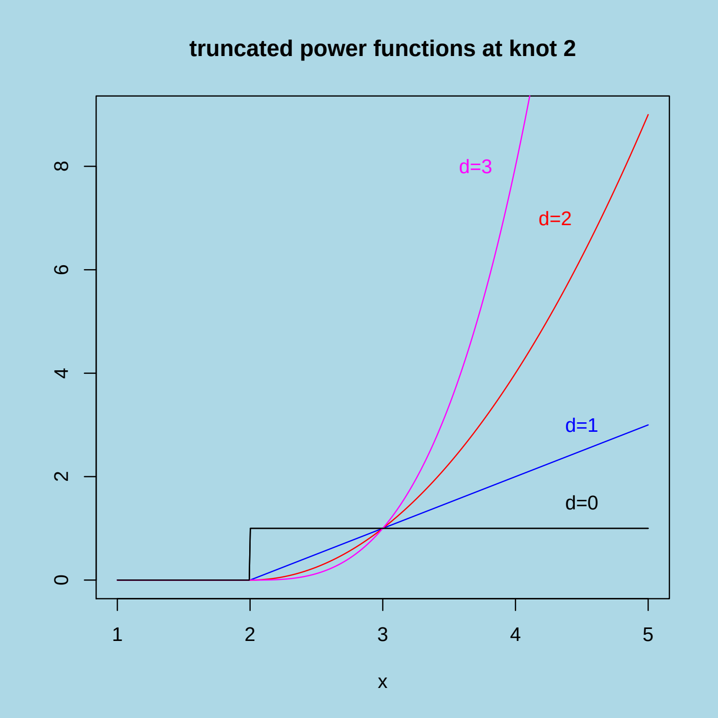
where $\{B_i^m(.)\}_{i=1,\ldots,m+K}$ is a basis of spline functions.

The vector β of the coordinate values is to be estimated by a regression method.

Two sets of basis functions

- **The truncated power functions**

$$\boldsymbol{x \longrightarrow (x-\tau)_+^d}$$



<u>When knots are distinct</u>, a basis of $S(m, \{\tau_{m+1}, \ldots, \tau_{m+K}\}, [a,b])$ is given by

$$1, x, \ldots, x^d, (x-\tau_{m+1})_+^d, \ldots, (x-\tau_{m+K})_+^d$$

Notice that, when $K = 0$,

$S(m, \emptyset, [a,b]) =$ the set polynomials of order m on $[a,b]$.

- **The B-splines**

B-splines of degree d (order $m = d + 1$): for $j = 1, \dots, m + K$,

$$B_j^m(x) = (-1)^m(\tau_{j+m} - \tau_j)[\tau_j, \dots, \tau_{j+m}](x - \tau)_+^d$$

where $[\tau_j, \dots, \tau_{j+m}](x - \tau)_+^d$ is the divided difference of order m computed at $\tau_j, \dots, \tau_{j+m}$ for the function $\tau \longrightarrow (x - \tau)_+^d$.

This basis is the most popular partly due to the next property that allows to compute recursively the values of B-splines, [1, De Boor]

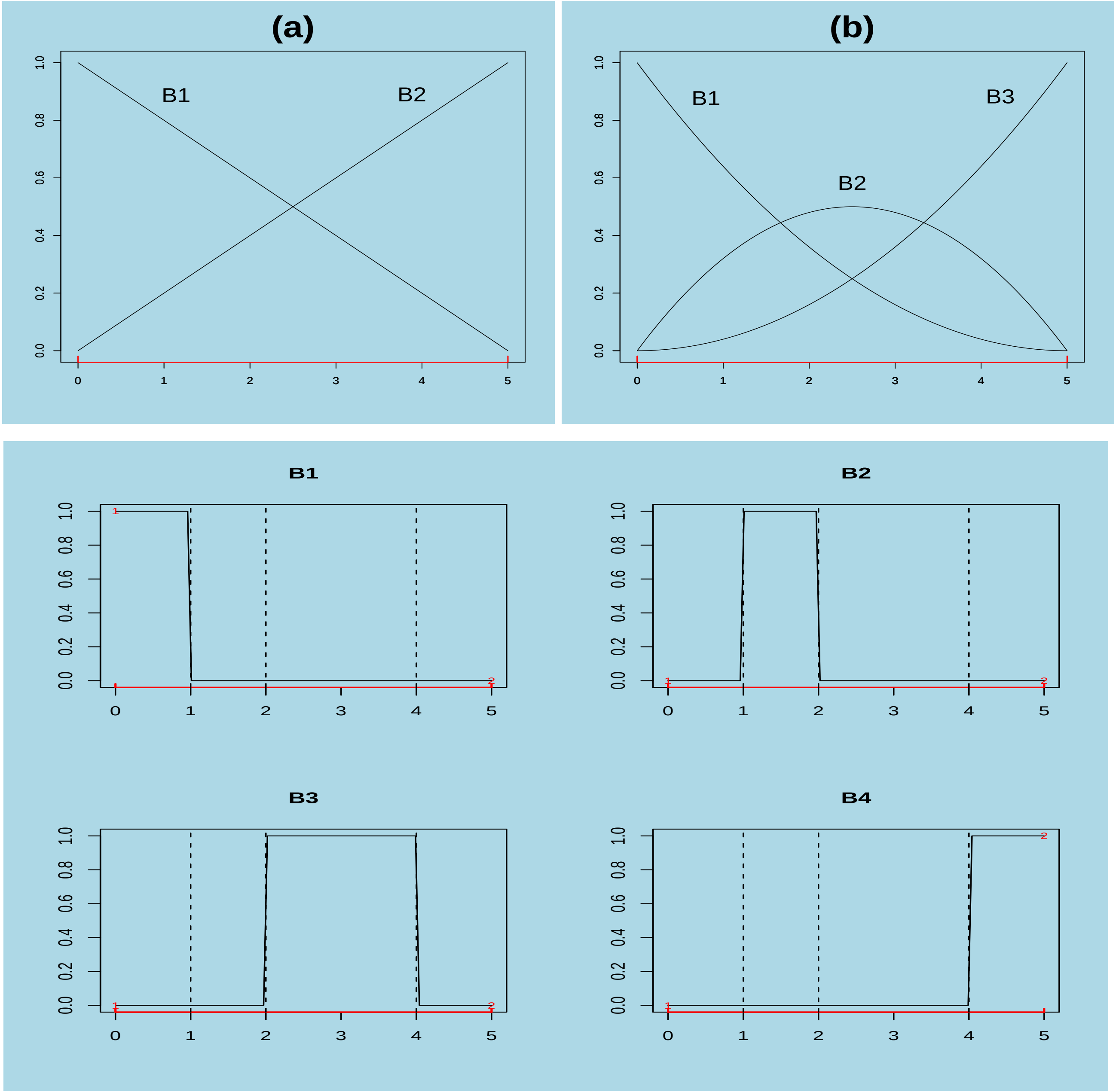
$B_j^1(x) = 1$ if $\tau_j \le x \le \tau_{j+1}$, 0 otherwise,
For $k = 2, \dots, m$,

$$B_j^k(x) = \frac{x - \tau_j}{\tau_{j+k-1} - \tau_j} B_j^{k-1}(x) + \frac{\tau_{j+k} - x}{\tau_{j+k} - \tau_{j+1}} B_{j+1}^{k-1}(x).$$

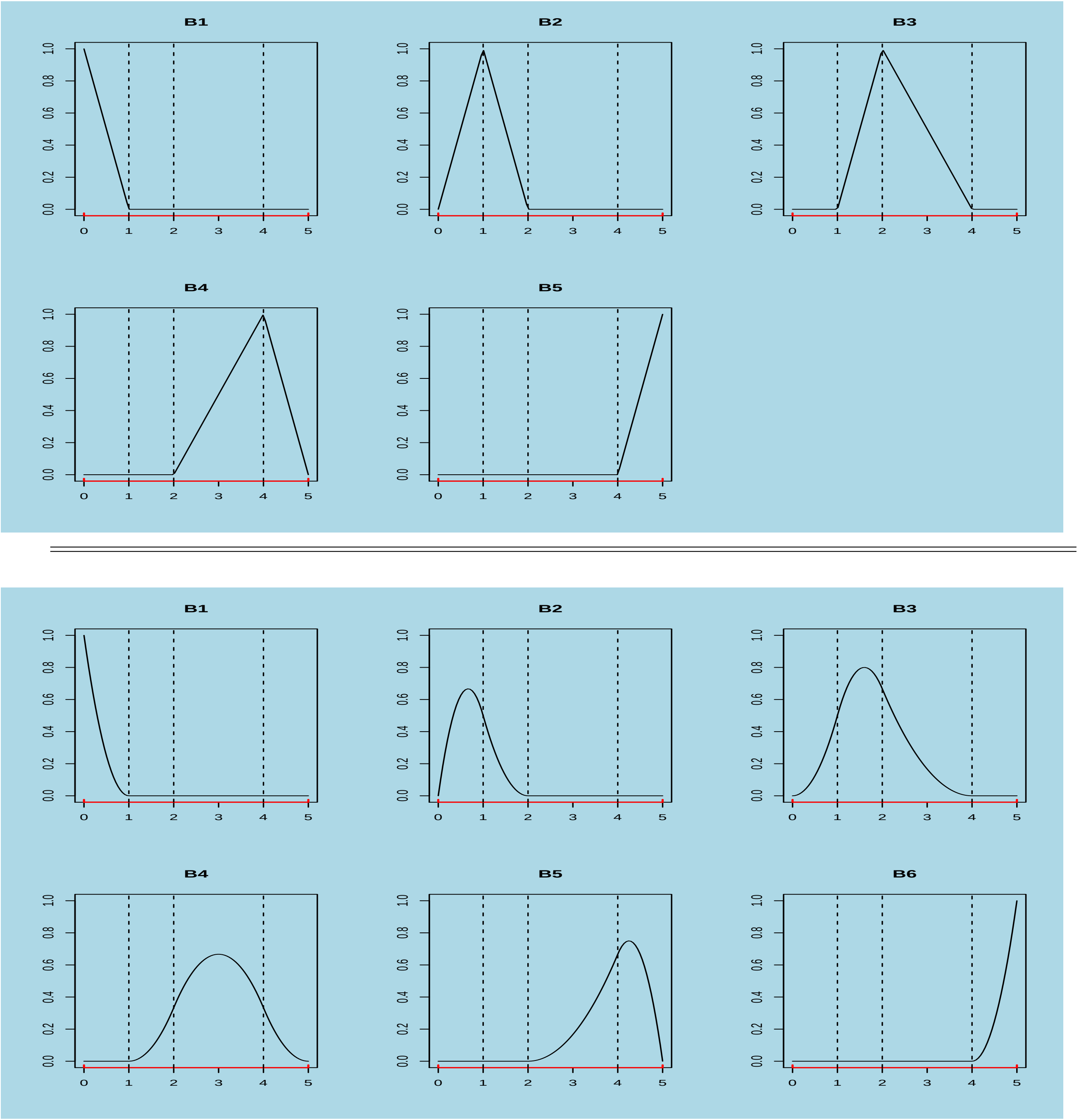
Many statistical packages implement those formulae to compute the $m + K$ values of the B-splines given an x sample
R-package:

```
library(splines)
x=seq(1,2*pi,length=100)
B=bs(x,degree=2,knots=c(pi/2,3*pi/2),intercept=T)
# What are the dimensions of the matrix B?
```

The attractive B-splines family for coding data



$S(2, \emptyset, [0,5])$ *(a)*, $S(3, \emptyset, [0,5])$ *(b) and* $S(1, \{1,2,4\}, [0,5])$.



B-splines for $S(2, \{1, 2, 4\}, [0, 5])$ *and* $S(3, \{1, 2, 4\}, [0, 5])$.

- **Local support**:

$$B_i^m(x) = 0, \quad \forall x \notin [\tau_i, \tau_{i+m}].$$

♡ → One observation x_i, has a local influence on $s(x_i)$ that depends only on the m basis functions whose supports encompass this data.

♠ → The counterpart is that $s(x) = 0$ outside $[a, b]$.

- **Fuzzy coding functions**:

$$0 \leq B_i^m(x) \leq 1 \ \mathbf{(1)} \text{ and } \sum_{i=1}^{m+K} B_i^m(x) = 1 \ \mathbf{(2)}.$$

♡ → $B_i^m(x)$ measures the degree of membership of x to $[\tau_i, \tau_{i+m}]$. The set $\{[\tau_i, \tau_{i+m}] \, | i = 1, ..., m+K\}$ is a fuzzy partition of $[a, b]$.

- **The multiplicity of knots controls the smoothness**: The multiplicity of a knot is the number of knots that merge at the same point. The multiplicity may vary from 1, a simple knot, to m, a multiple knot of order m.

Let m_i be the multiplicity of τ_i, $0 \leq m_i \leq m$,
then the first $m - 1 - m_i$ right and left derivatives are equal,

$$s_-^{(j)}(\tau_i) = s_+^{(j)}(\tau_i), \quad j \leq m - 1 - m_i.$$

$$\begin{aligned} m_i = m \quad &\Rightarrow \text{discontinuity at } \tau_i \\ m_i = 1 \quad &\Rightarrow \text{locally } C^{m-2} \text{ at } \tau_i. \end{aligned}$$

- **Coding a variable x through B-splines**

  Due to **(2)**, two B-splines bases are generally used:

  $\{B_j^m(x) \,|\, j = 1, \ldots, m+K\}$ usual basis

  $\{1, B_j^m(x) \,|\, j = 2, \ldots, m+K\}$, modified basis.

Let $X = (x_1, \ldots, x_n)'$ be a n-sample of the variable x, denote

$$B = [B^1(X) \ldots B^{m+K}(X)] \quad \text{or} \quad B = [B^2(X) \ldots B^{m+K}(X)]$$

the complete $n \times (m+K)$, or incomplete $n \times (d+K)$, coding matrix of the sample.

Notice that $d = 0$ provides a binary coding matrix B.

D-centering the coding matrix

When the columns of B are centered, then,

$$rank(B) \leq \min(n-1, d+K).$$

Bivariate regression splines for (x, z)

$\{1, B_1^j(x) \,|\, j \in I_1\}$ and $\{1, B_2^j(z) \,|\, j \in I_2\}$ univariate bases

$$s(x,z) \in span[\{1, B_1^j(x)|j \in I_1\} \bigotimes \{1, B_2^j(z)|j \in I_2\}]$$

A bivariate regression spline split into the ANOVA decomposition:

$$\boxed{s(x,z) = \beta_1 + \sum_{j \in I_1} \beta_j^1 B_1^j(x) + \sum_{j \in I_2} \beta_j^2 B_2^j(z) + \sum_{i \in I_1} \sum_{j \in I_2} \beta_{i,j}^{1,2} B_1^i(x) B_2^j(z)}$$

<u>main effects</u> s^1 and s^2 in x and z

$$s^1(x) = \sum_{j \in I_1} \beta_j^1 B_1^j(x) \qquad s^2(z) = \sum_{j \in I_2} \beta_j^2 B_2^j(z)$$

<u>interaction part</u> s^{12}

$$s^{12}(x,z) = \sum_{i \in I_1} \sum_{j \in I_2} \beta_{i,j}^{1,2} B_1^i(x) B_2^j(z)$$

<u>How to measure the importance of an ANOVA term?</u>

<u>by the range of the transformation</u> (when standardized variables).

$B = [B_1 | B_2 | B_{1,2}]$ column centered coding matrix from X and Z.

Curse of dimensionality: expansion of the column dimension.
ncol$(B_1) = 10$, ncol$(B_2) = 10$, ⇒ ncol$(B_{1,2}) = 100$.

The Least-Squares Splines (LSS) [12, Stone]

Denote X, $n \times p$, and Y, $n \times q$, the sample matrices for the p predictors and the q responses (centered).

The centered coding matrix of X, $B = [B_1 | \dots | B_p]$, leads to q separate additive spline models, $j = 1, \dots, q$,

$$\widehat{y}^j = s^{j,1}(x^1) + \dots + s^{j,p}(x^p)$$

through

$$\textbf{LSS}(X, Y) \equiv \textbf{OLS}(B, Y) \Longleftrightarrow \widehat{Y} = HY = B\widehat{\beta} = \textstyle\sum_i B_i \widehat{\beta}_i$$

where the so called linear smoother $H = B(B'B)^{-1}B'$.

Tuning parameters: The spline spaces used for the predictors

LSS drawbacks: numerical instability of $(B'B)^{-1}$ if it exists !

♠ needs a large ratio observations/(column dimension of B)

♠* very sensitive to knots' location

♠ perturbing concurvity effects due to correlated predictors

♠** no interaction terms involved

♡** MARS [6, Friedman] proposes to remedy ♠**: automatic $\nearrow$ $\searrow$ procedure to select knots and high order interactions through linear truncated power functions.

♡* EBOK [11, Molinari et al.]: K fixed, optimal location of knots.

∘ others....

The Penalized Least-Squares Splines

[5, P. Eilers and B. Marx]

Because of drawbacks cited above, L-S Splines are mostly efficient in the one-dimensional context ($p = 1$).

In that univariate case, to remedy $\spadesuit^*$ by using a large number of equidistant knots, [5] proposed to penalize the L_2 cost by a penalty (λ) on k-finite differences of the coefficients of adjacent B-splines.

$$S = \sum_{i=1}^{n} \left\{ y_i - \sum_{j=1}^{r} \beta_j B_j(x_i) \right\}^2 + \lambda \sum_{j=k+1}^{r} (\Delta^k \beta_j)^2$$

The linear smoother associated to the P-splines model becomes

$$H = B(B'B + \lambda {D_k}' D_k)^{-1} B'$$

Notice that $\lambda = 0$ leads to LSS and $k = 0$ to ridge regression.

Default : $k = 2$ (strong connection with second derivatives)

$$\text{example : } r = ncol(B) = 5 \qquad D_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}.$$

Following [9, T. Hastie and R. Tibshirani]

$$trace(H) = \text{effective dimension of the smoother}$$

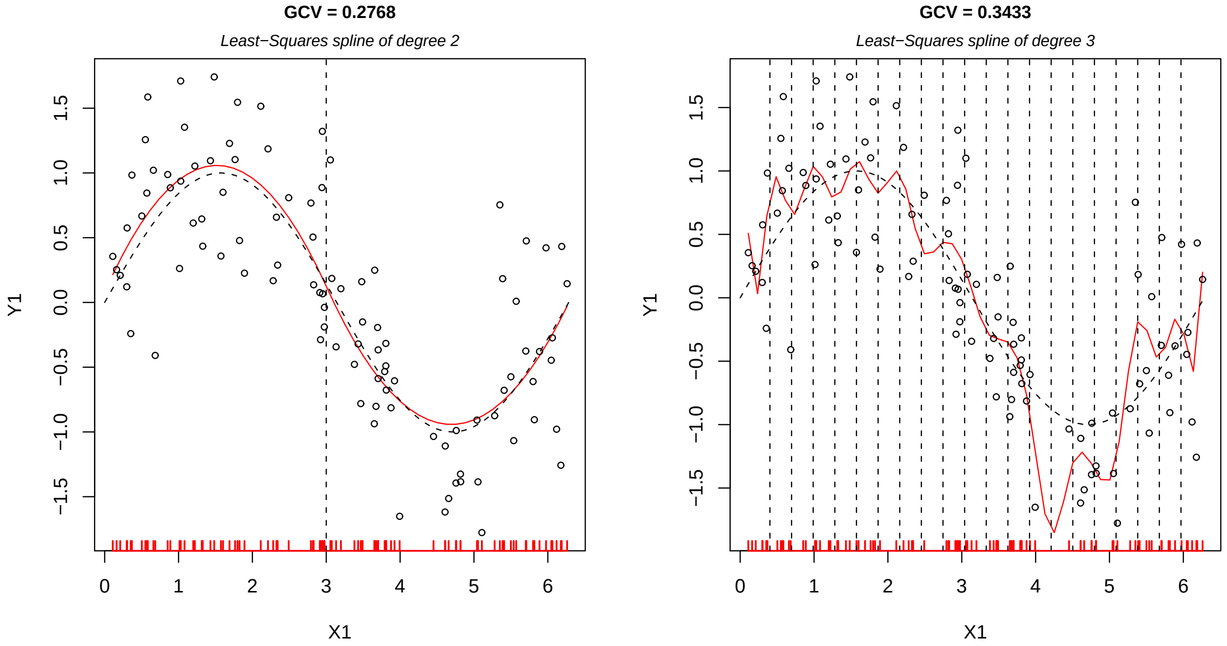
so that the Generalized Cross-Validation index

$$GCV(\lambda, k) = var(residuals)/(1 - trace(H/n))^2$$

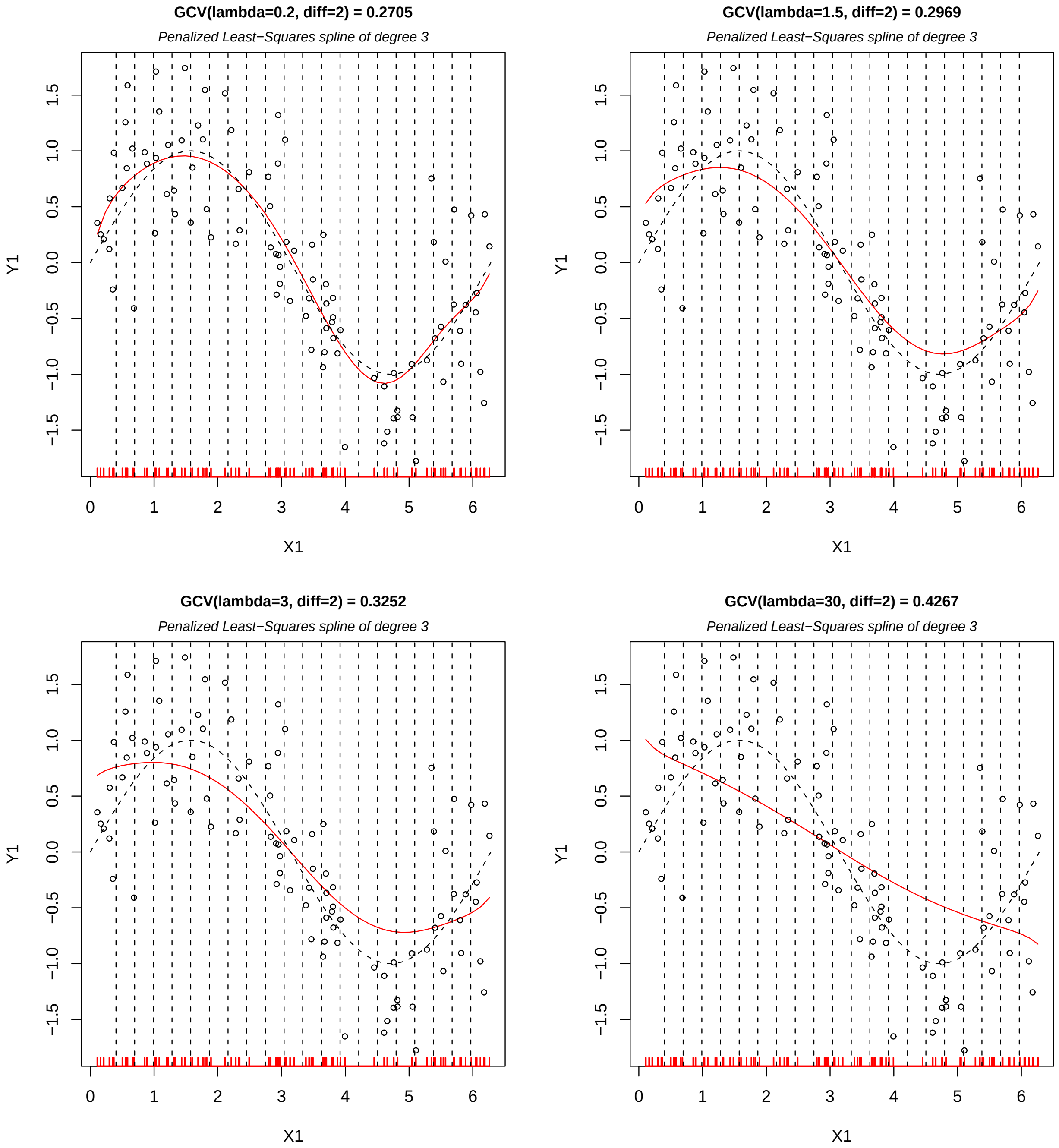
is a surrogate to the cross validation Predictive Error Sum of Squares (PRESS)

One example : $(x_i, y_i)_{i=1,100}, \qquad x_i \sim U[0, 2\pi],$

$y_i = sin(x_i) + \varepsilon_i, \qquad \varepsilon_i \sim N(0, 0.5)$



L-S Splines (red), signal (dotted), vertical lines indicate the location of knots



P-splines (red) with $\lambda \in \{0.2, 1.5, 3, 30\}$.

II. Multivariate Additive PLS Splines

What is L_2 Boosting? [7, J.H. Friedman]

The training sample: $\{y_i, \underline{x}_i\}_1^n$, $\quad y \in I\!R$, $\underline{x} \in I\!R^p$, centered

to estimate the function F^* restricted to be of "additive" type,

$$F(\underline{x}\,;\,\{\alpha_m, \underline{\theta}_m\}_1^M) = \sum_{m=1}^{M} \alpha_m h(\underline{x}, \underline{\theta}_m),$$

that minimize the expected L_2 cost

$$I\!E[C(y, F(\underline{x}))] \qquad C(y, F) = (y - F)^2/2 \; .$$

- the base learner $h(\underline{x}, \underline{\theta})$

↦ a function of the input variables $\underline{x}$ characterized by parameters

$$\underline{\theta} = \{\theta_1, \theta_2, \ldots\} \; .$$

↦ neural nets, wavelets, splines, regression trees...

but also, latent variables,

$$\underline{\theta} \in I\!R^p \qquad , \qquad t = h(\underline{x}, \underline{\theta}) = <\underline{\theta}, \underline{x}> \; .$$

- M, the "dimension" of the additive model.

Boosting:

a stagewise functional gradient descent method with respect to F.

Strict L_2 boost algorithm:

```
$F_0(\underline{x}) = 0$
For $m = 1$ to $M$ do
    $\tilde{y}_i = -\frac{\partial C(y_i, F)}{\partial F}|_{F=F_{m-1}(\underline{x}_i)} = y_i - F_{m-1}(\underline{x}_i), \quad i = 1, n,$
    $(\alpha_m, \underline{\theta}_m) = \arg\min_{\alpha, \underline{\theta}} \sum_{i=1}^{n} [\tilde{y}_i - \alpha h(\underline{x}_i; \underline{\theta})]^2 \qquad (*)$
    $F_m(\underline{x}) = F_{m-1}(\underline{x}) + \alpha_m h(\underline{x}; \underline{\theta}_m)$
endFor
```

Extended L_2 boost algorithm:

```
Replace $(*)$ by
Criterion or procedure to construct $\underline{\theta}_m$ from $\{(\underline{x}_i, y_i)\}_{i=1,n}, \quad (*1)$
$\alpha_m = \arg\min_{\alpha} \sum_{i=1}^{n} [\tilde{y}_i - \alpha h(\underline{x}_i; \underline{\theta}_m)]^2. \quad (*2)$
```

To summarize, L_2 boosting is characterized by

- choosing a base learner
- repeated least-squares fitting of residuals

Ordinary PLS as a L_2 Boost algorithm

The context of $PLS(X, Y)$:

- $X_{n\times p}$ for the p predictors, $Y_{n\times q}$ for the q responses.
- $D = diag(p_1, \ldots, p_n) = n^{-1} I_n$ weights of the observations.

All variables are centered (standardized) with respect to D so that

$$cov(x, y) = < x, y >_D = y' Dx \text{ and } var(x) = \|x\|_D^2.$$

The algorithm [13, H. Wold],[14, S. Wold et al.]

PLS constructs components $\{t^m\}_{m=1,\ldots,M}$ as linear compromises of X, on which LS residuals are repeatedly regressed.

PLS	$X_{(0)} = X, \quad Y_{(0)} = Y$	
Step m m=1,...,M	**1) base learner** constructing $\boldsymbol{t}^m \in span(\mathbf{X}_{(m-1)})$ update $\boldsymbol{X}_{(m)}$ (deflation)	$t = X_{(m-1)} w, \; u = Y v$ $(w^m, v^m) = arg \max_{w'w=1=v'v} cov(t, u)$ $t^m = X_{(m-1)} w^m, \; u^m = Y v^m$ $X_{(m)} = X_{(m-1)} - H_m X_{(m-1)}$
	2) Y-residuals	$Y_{(m)} = Y_{(m-1)} - H_m Y_{(m-1)}$

where the so-called linear PLS learner,

$$H_m = \Pi^D_{t^m} = t^m t^{m\prime} D / \|t^m\|^2_D$$

is the $n \times n$ D-orthogonal Least-Squares projector onto t^m.

Notice that the projector H_m depends also on Y since t^m is of maximum covariance with a Y-compromise .

To solve the optimization problem 1) with two constraints, the Lagrange multipliers technique leads to

Proposition : The PLS base learner problem is solved by the first term in the singular value decomposition of the $p \times q$ covariance matrix

$$X'_{(m-1)} DY.$$

Let (λ_m, w^m, v^m) be the triple corresponding to the largest (first) singular value λ_m, then

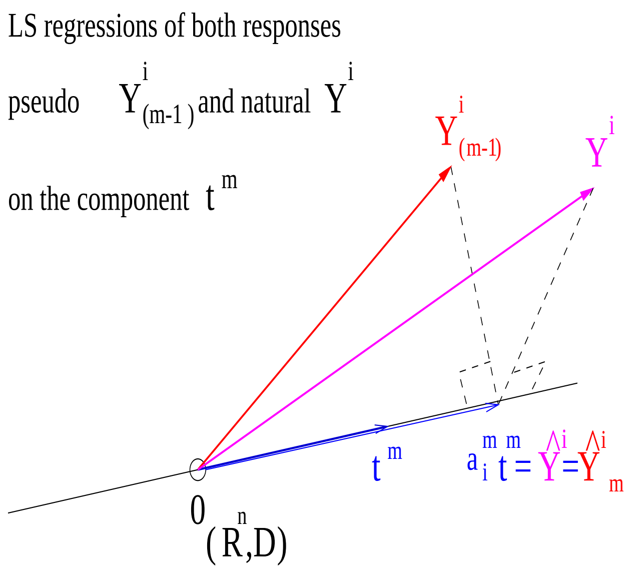
$$t^m = X_{(m-1)} w^m, \qquad u^m = Y v^m, \qquad \lambda_m = cov(t^m, u^m).$$

In the one-response case, $v^m = 1$ and $w^m = X'_{(m-1)} DY / ||X'_{(m-1)} DY||$.

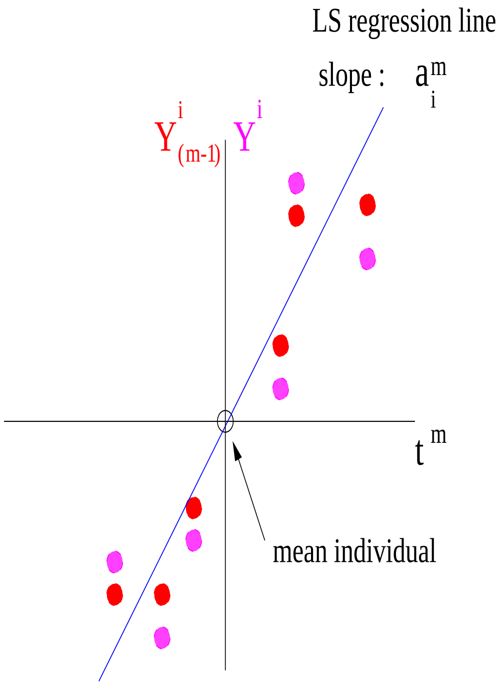
Components $\{t^m\}$

- belong to $span(X)$, that is $\qquad t^m = X\theta^m$
- are mutually D-orthogonal $\qquad t^{m_1\prime}Dt^{m_2} = 0, \quad m_1 \neq m_2$.
- Partial Regressions of pseudo variables give the same results as LS regressions of the original variables on the components.

$$\hat{X}_m = H_m X_{(m-1)} = H_m X = t^m p^{m\prime}$$
$$\hat{Y}_m = H_m Y_{(m-1)} = H_m Y = t^m \alpha^{m\prime}$$



LS regressions of both responses pseudo $Y^i_{(m-1)}$ and natural Y^i on the component t^m



LS regression line

slope : a^m_i

The linear model with respect to $\{t^1, \ldots, t^M\}$

Denoting $\quad T^M = [t^1 ... t^M]_{n \times M}$

After M steps, the fit from both X and Y sides, is

$$\hat{X}(M) = \sum_{m=1}^{M} \hat{X}_m = \Pi_{T^M}^{D} X = (H_1 + \ldots + H_M) X \, ,$$

$$\hat{Y}(M) = \sum_{m=1}^{M} \hat{Y}_m = \Pi_{T^M}^{D} Y = (H_1 + \ldots + H_M) Y \, , \qquad (1)$$

so that, the set $\{H_m\}_1^M$ tries to sequentially reconstruct X

$$X = \hat{X}(M) + X_{(M)}$$

and provides a typical L_2-Boosting additive model (1) whose base learners are the PLS components

$$Y = t^1 \alpha^{1\prime} + \ldots + t^M \alpha^{M\prime} + Y_{(M)} \, .$$

If $M = rank(X)$, then, $span(T^M) = span(X)$, $X_{(M)} = 0$ and

$$PLS(X, Y) \equiv OLS(X, Y),$$

thus providing an upper bound for the number of components

$$1 \leq M \leq rank(X).$$

PCA of X is the "self"-PLS regression of X onto itself

$$PLS(X, Y = X) \equiv PCA(X).$$

The linear model with respect to X

Recall that a component is a linear combination of the natural predictors

$$t^m = X\theta^m.$$

The $\{\theta^m\}_m$ set provides a ($\mathbb{V} = X'DX$)-orthogonal basis to $span(X')$

$$< t^{m_1}, t^{m_2} >_D = < \theta^{m_1}, \theta^{m_2} >_{\mathbb{V}} = \ 0$$

The use of $\{\theta^m\}_m$ is twofold

- $\mathbb{V}$-project the X-samples on $\{\theta^m\}$ and look at 2-D scatterplots of the X observations.
- stepwise build the linear PLS model with respect to the natural predictors

$$\hat{Y}(M) = (H_1 + \ldots + H_M)Y = X\beta(M) \qquad (2)$$

where $\beta(M)$ is the $p \times q$ matrix of the linear model, recursively computed:

$$\begin{aligned} \beta(0) &= 0 \\ \beta(m) &= \beta(m-1) + \theta^m \theta^{m\prime} X' DY / \|\theta^m\|^2_{\mathbb{V}}. \end{aligned}$$

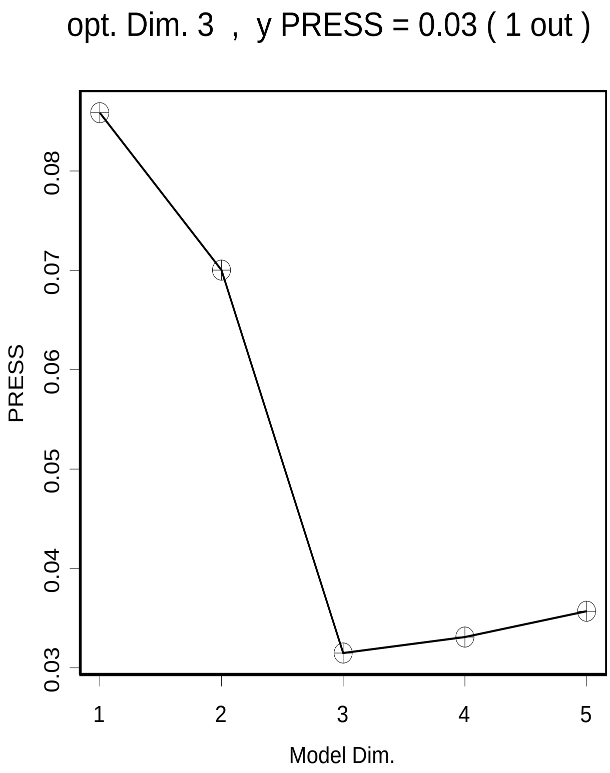
The building-model stage: choosing M

- Cross-Validation criterion: PRESS

$m = 1, \ldots, rank(X)$; $prop =$ proportion of samples out, default 0.1.

When $prop$ is such that 1 observation out at a time,

$$PRESS(prop, m) = \sum_{j=1}^{q} \frac{1}{n} \sum_{i=1}^{n} (Y_i^j - X_i \hat{\beta}(m)_{(-i)}|^j)^2 \,.$$

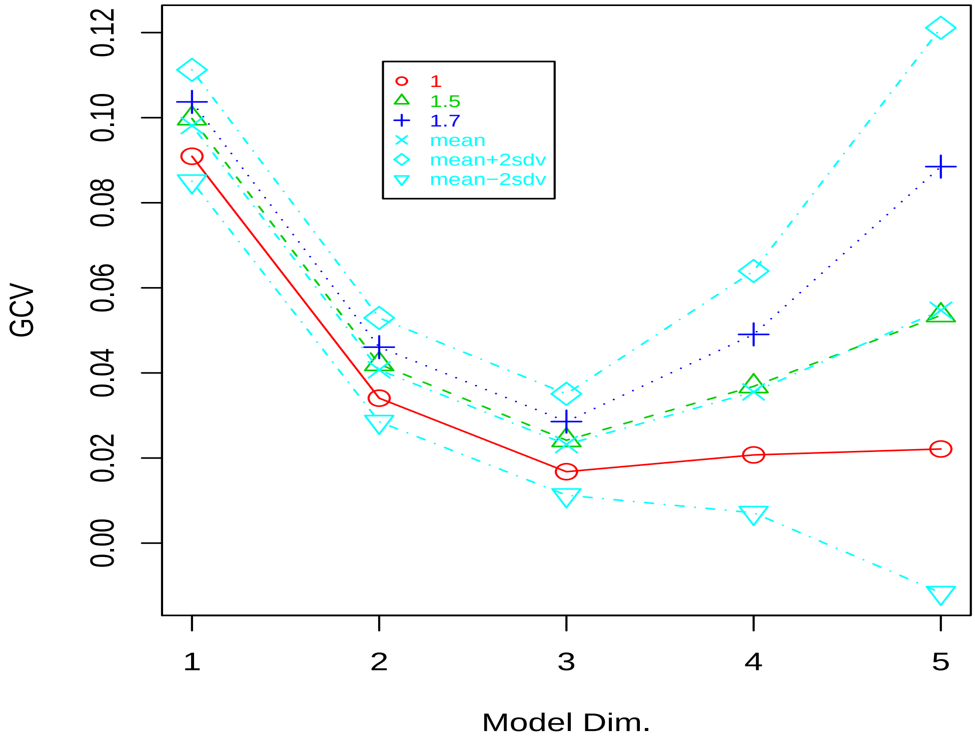


- A surrogate to CV : Generalized Cross-Validation (GCV)

  $m = 1, \ldots, rank(X)$; $\quad \alpha =$ penalty term (default 1)

$$GCV(\alpha, m) = \frac{\sum_{j=1}^{q} \frac{1}{n} \|Y^j - \hat{Y}(m)^j\|^2}{[1 - \alpha \frac{m}{n}]^2}$$

Calibration of the penalty α: $GCV(\alpha = 1.7, M = 3) = 0.29$



PLS Splines (PLSS): a main effects additive model [3, J.F. Durand]

The PLSS model

The centered coding matrix of X being $B = [B_1|\ldots|B_p]$

PLS through Splines (PLSS) is defined as

$$\mathbf{PLSS}(X, Y) \equiv \mathbf{PLS}(B, Y)$$

- Tuning parameters: the spline spaces for each predictor

  the model dimension $M \mapsto$ crossvalidation

- The PLSS additive model: $j = 1, \ldots, q,$

$$\widehat{y}^j_M = s^{j,1}_M(x^1) + \ldots + s^{j,p}_M(x^p)$$

The method inherits the advantages of PLS and B-splines:

♡ if $M = rank(B)$ then PLSS(X, Y) = LSS(X, Y) if it exists
♡ PLSS$(X, Y = B) \equiv$ PLS(B, B) = NL-PCA(X), [8, Gifi]
♡ efficient with low ratio observations/(column dimension of B)
♡ efficient in the multi-collinear context for predictors (concurvity)
♡ robust against extreme values of predictors (local polynomials)
♠ ♡ no automatic procedure for choosing spline parameters
♠ no interaction terms involved

Choosing the tuning parameters

Recall the tuning parameters:

1. The spline space for each predictor:
   degree, number and location of knots
2. the number of PLS components

1. Two strategies for choosing the spline spaces

- **The ascending strategy**
  - First, take $d = 1$ with no knots $(K = 0) \mapsto$ linear model.
  - Increase the degree d, keeping $K = 0$, $\mapsto$ polynomial model.
  - for fixed d, add knots, $\mapsto$ local polynomial model
    ”adding a knot increases the local flexibility of the spline and then, the freedom of fitting the data in this area.”
- **The descending strategy**
  - First, take a high degree, $d = 3$, and more knots than necessary.
  - Remove superfluous knots and decrease the degree as much as possible

2. To stop a strategy : find a balance between

thriftiness (M and the total spline dimension) and goodness-of-prediction, $PRESS$ and/or GCV.

Multi-collinearity, nonlinearity and outliers: the orange juice data

- $(x^1, \ldots, x^p)$, p predictors, $X = [X^1|\ldots|X^p]$ $n \times p$
- $(y^1, \ldots, y^q)$, q responses, $Y = [Y^1|\ldots|Y^q]$ $n \times q$

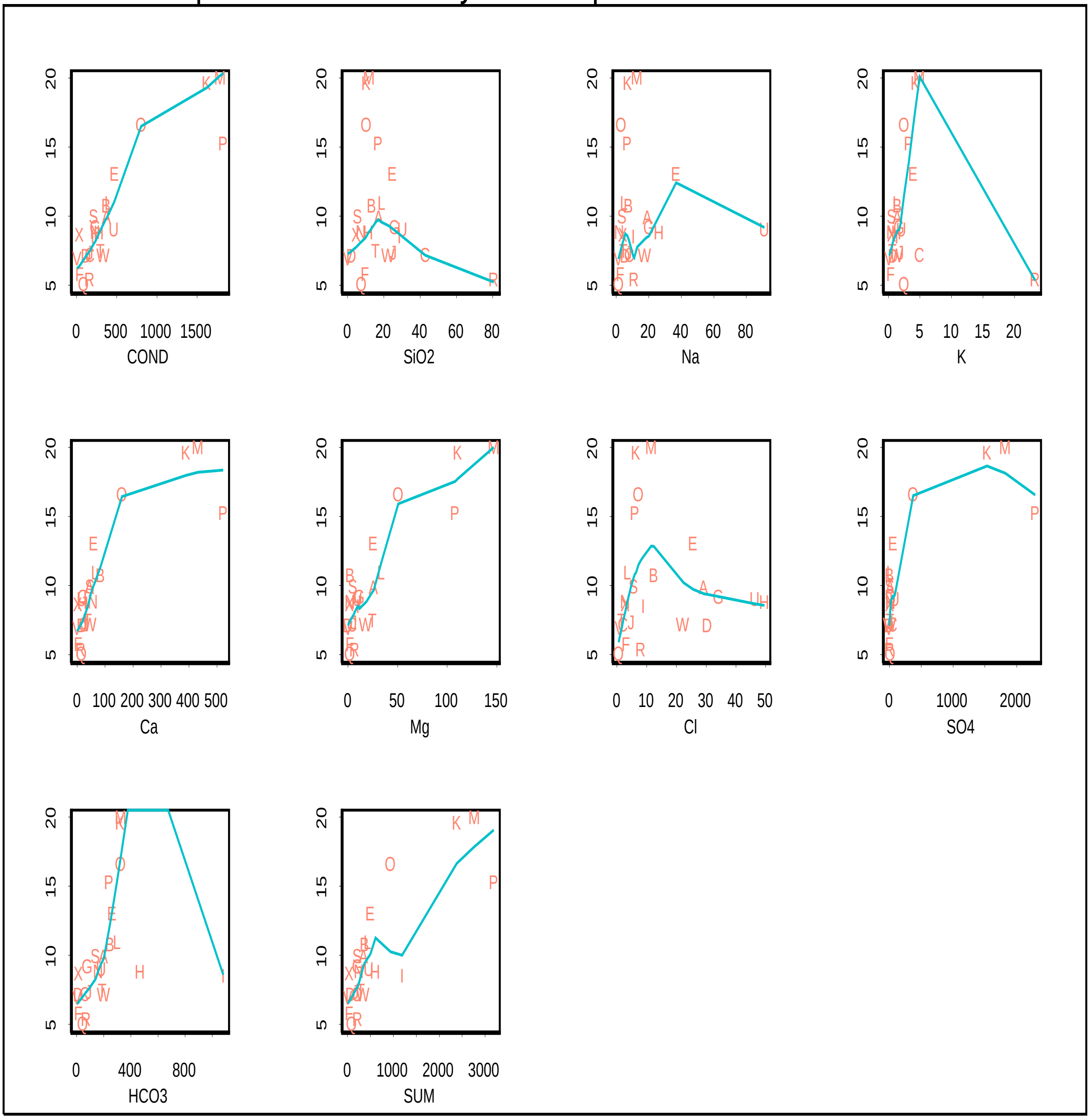
Both sample matrices are standardized

$n = 24$ orange juices: A, B, C, ..., X
$p = 10$, $q = 1$,

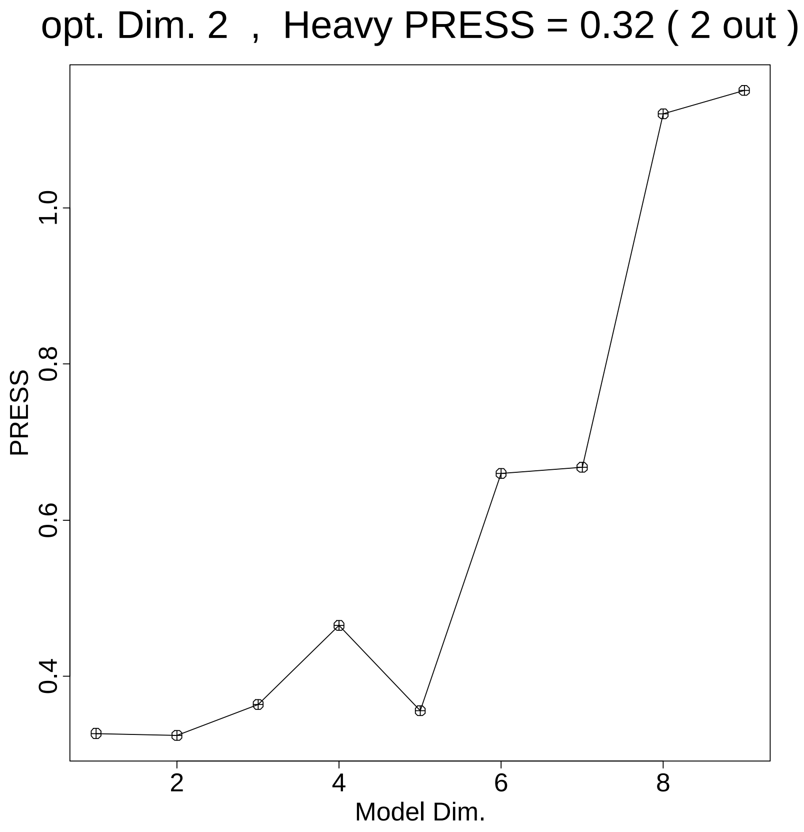
PREDICTORS	sensorial RESPONSE
COND (Conductivity) SiO2 Na K Ca Mg Cl HCO3 SO4 Sum	Heavy

collinearity : $SUM = SiO2 + ... + SO4$

biplots between Heavy and the predictors with ’lowess’

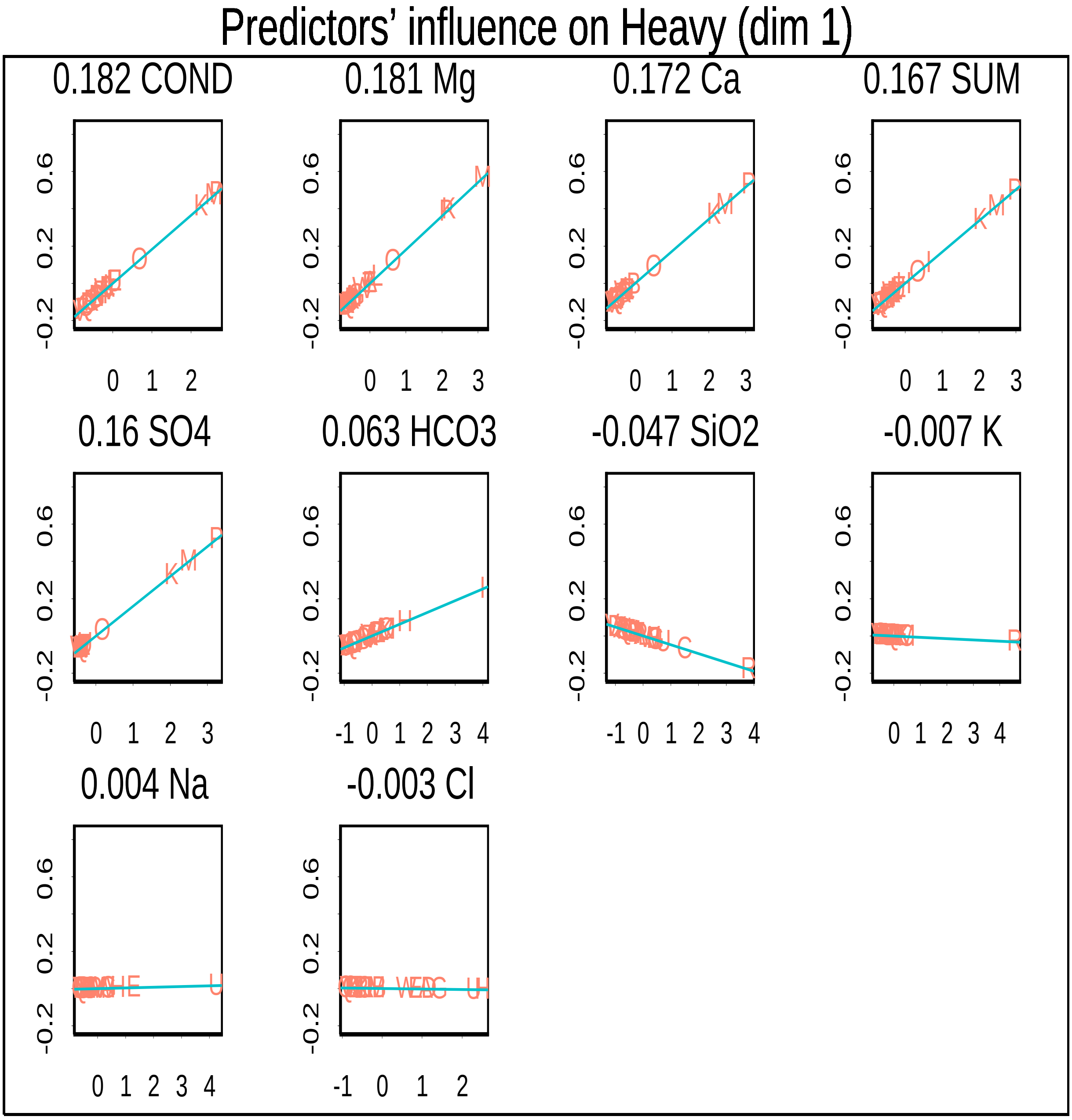


Linear PLS on the orange juice data



$$R^2(1) = 0.754 \qquad PRESS(0.1, 1) = 0.32$$

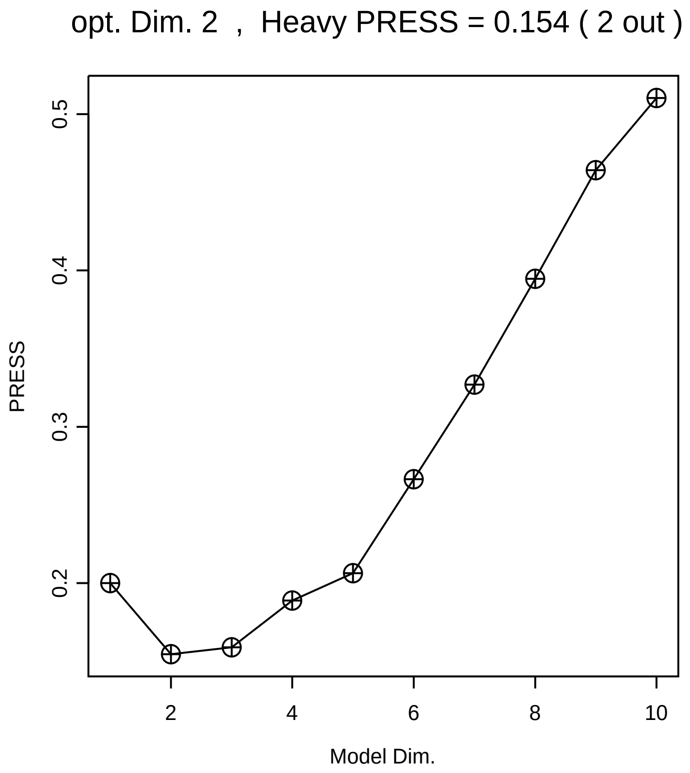
Retained dimension : $M = 1$


Predictors' influence on Heavy (dim 1)
0.182 COND
0.181 Mg
0.172 Ca
0.167 SUM
0.16 SO4
0.063 HCO3
-0.047 SiO2
-0.007 K
0.004 Na
-0.003 Cl
-0.2
0.2
0.6
0 1 2
0 1 2 3
-1 0 1 2 3 4
0 1 2 3 4
-1 0 1 2

PLSS on the orange juice data

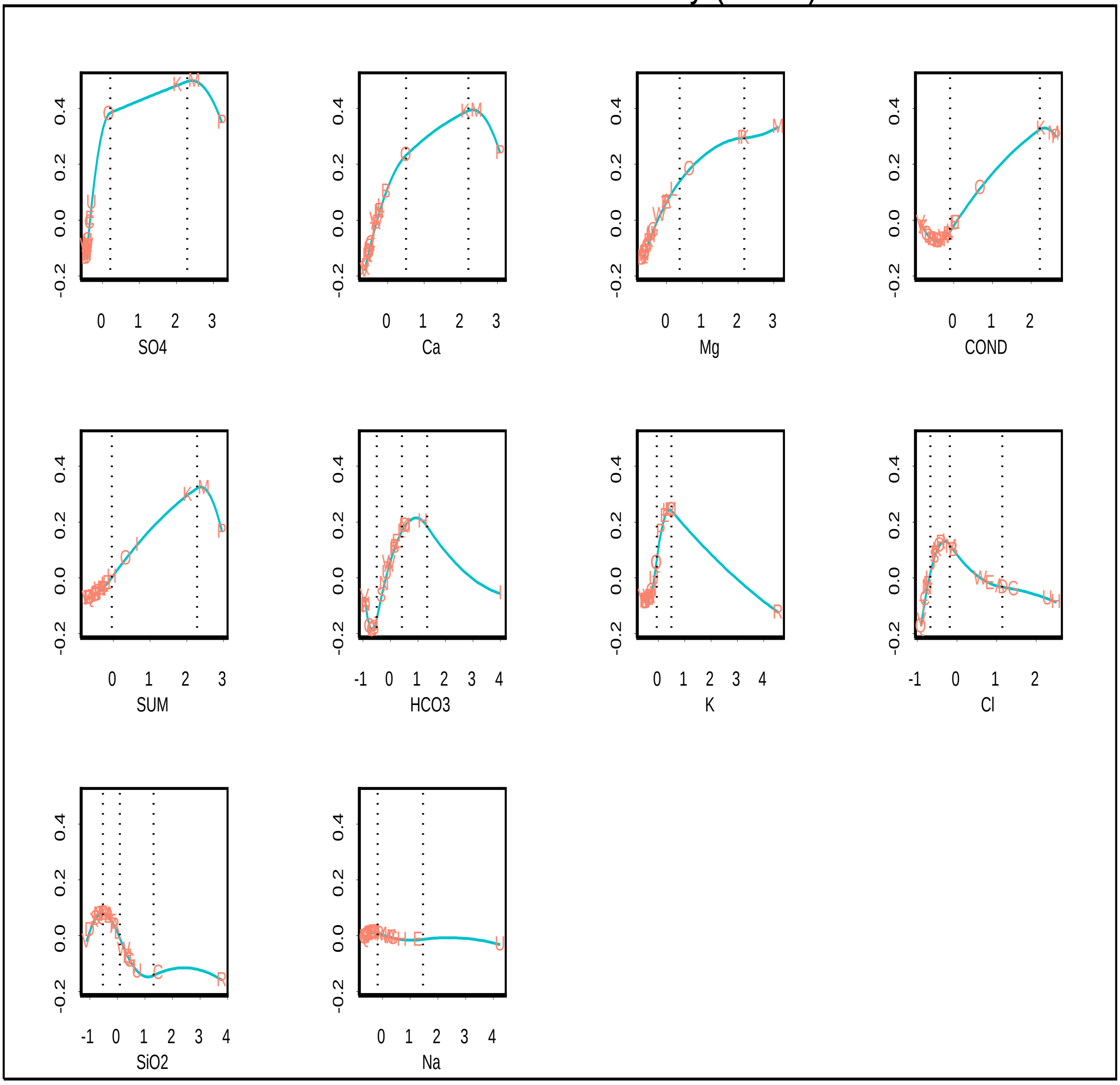
Table 1: Selected knots for the predictors

COND	SiO_2	Na	K	Ca	Mg	Cl	SO_4	HCO_3	Sum
400	10	10	2.5	160	40	4	400	100	600
1600	20	40	5	400	110	11	1700	300	2600
	40					30		500	

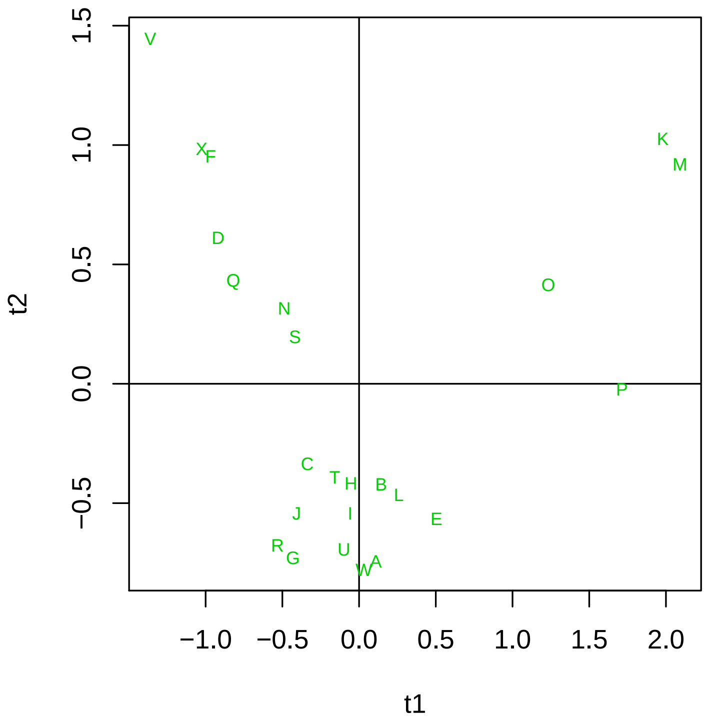


$$R^2(2) = 0.917 \qquad PRESS(0.1, 2) = 0.154$$

Predictors’ influence on Heavy (2 dim.)



Nonlinear 2-D component scatterplots



A PLSS component t^i is additively modeled by ANOVA terms

$$t^i = Bw^{*i} = \sum_{j=1}^{p} B_j w_j^{*i} = \sum_{j=1}^{p} \phi_j^i(X^j)$$

where w_j^{*i} is the sub-vector of w^{*i} associated to the block B_j, thus allowing to interpret t^i by the predictors.

Because $r(t^1, Y) = 0.924$, then $\phi_j^1(X^j) \approx s_M^j(X^j)$ and one can use the $Heavy$ ANOVA plots to explain the t^1 coordinates.

Multivariate Additive PLS Splines (MAPLSS): capture of interactions

[4, Durand & Lombardo][10, Lombardo, Durand, De Veaux]

The ANOVA type model for main effects and interactions

The centered main effects + interactions coding matrix:

$$B = [B_1|\dots|B_p\,\|\dots|B_{i,i'}|\dots]$$

where (i, i') belong to the set $\mathcal{I}$ of accepted couples of interactions

$$\mathbf{MAPLSS}(X, Y) = \mathbf{PLS}(B, Y)$$

q simultaneous models casted in the ANOVA decomposition

$$j = 1, \dots, q, \qquad \hat{y}^j_M = \sum_{i=1}^{p} s^{j,i}_M(x^i) + \sum_{(i,i')\in\mathcal{I}} s^{j,ii'}_M(x^i, x^{i'})$$

PLSS with interactions shares the preceding ♡ properties

♠ ♡ no automatic procedure for choosing spline parameters

The building-model stage

Inputs: $threshold = 20\%$, M_{max} = dim. maximum to explore

0 Preliminary phase: the main effects model (mandatory).

Decide on the spline parameters as well as on M_m for the main effects model (m): denote $GCV_m(M_m)$ and $R^2_m(M_m)$.

1 Individual evaluation of all interactions.

Each interaction i is separately added to the main effects.

$$crit(M_i) = \max_{M \in \{1, M_{max}\}} \frac{R^2_{m+i}(M) - R^2_m(M_m)}{R^2_m(M_m)} + \frac{GCV_m(M_m) - GCV_{m+i}(M)}{GCV_m(M_m)}$$

Eliminate interactions i such that $CRIT(M_i) < 0$

Order decreasingly the accepted candidate interactions.

2 Add successively interactions to main effects (forward phase).

```
Set $GCV_0 = GCV_m(M_m)$ and $i = 0$.
REPEAT
    □ $i \leftarrow i + 1$
    □ add interaction i and index the new model with $i$
    □ $GCV_i \leftarrow$ optimal $GCV$ among all dimensions
UNTIL $(GCV_i < GCV_{i-1} - threshold * GCV_{i-1})$
```

3 prune ANOVA terms of lowest influence (backward phase)

criterion: the range of the values of the ANOVA function

Comparison between MAPLSS, MARS and BRUTO on simulated data

BRUTO [9, Hastie & Tibshirani]: A multi-response additive model fitted by adaptive back-fitting using smoothing splines.

MARS[7, Friedman]: Multivariate Adaptive Regression Splines. A two-phase process to build ANOVA style decomposition models by fitting truncated power basis function. Variables, knots and interactions are optimized simultaneously by using a GCV criterion.

Campaign of simulations:

Three signals (pure additive, one and two interactions respectively)

$$f_1(\boldsymbol{x}) = 0.1\, exp(4\, x_1) + \frac{4}{1 + exp(-20\,(x_2 - 0.5))} + 3\, x_3 + 2\, x_4 + x_5 + 0 \sum_{i=6}^{10} x_i$$

$$f_2(\boldsymbol{x}) = 10\, sin(\pi x_1 x_2) + 20\,(x_3 - 0.5)^2 + 10\, x_4 + 5\, x_5 + 0 \sum_{i=6}^{10} x_i$$

$$f_3(\boldsymbol{x}) = 10\, sin(\pi x_1 x_2) + 20\,(x_3 - 0.5)^2 + 20\, x_4 x_5 + 0 \sum_{i=6}^{10} x_i$$

not depending on the last five variables.

The model associated to the signal f_j

$$y_i = f_j(\boldsymbol{x}_i) + \varepsilon_i, \qquad i = 1, \ldots, n$$

$$\boldsymbol{x}_i \sim U[0,1]^{10} \qquad \text{and} \qquad \varepsilon_i \sim N[0,1].$$

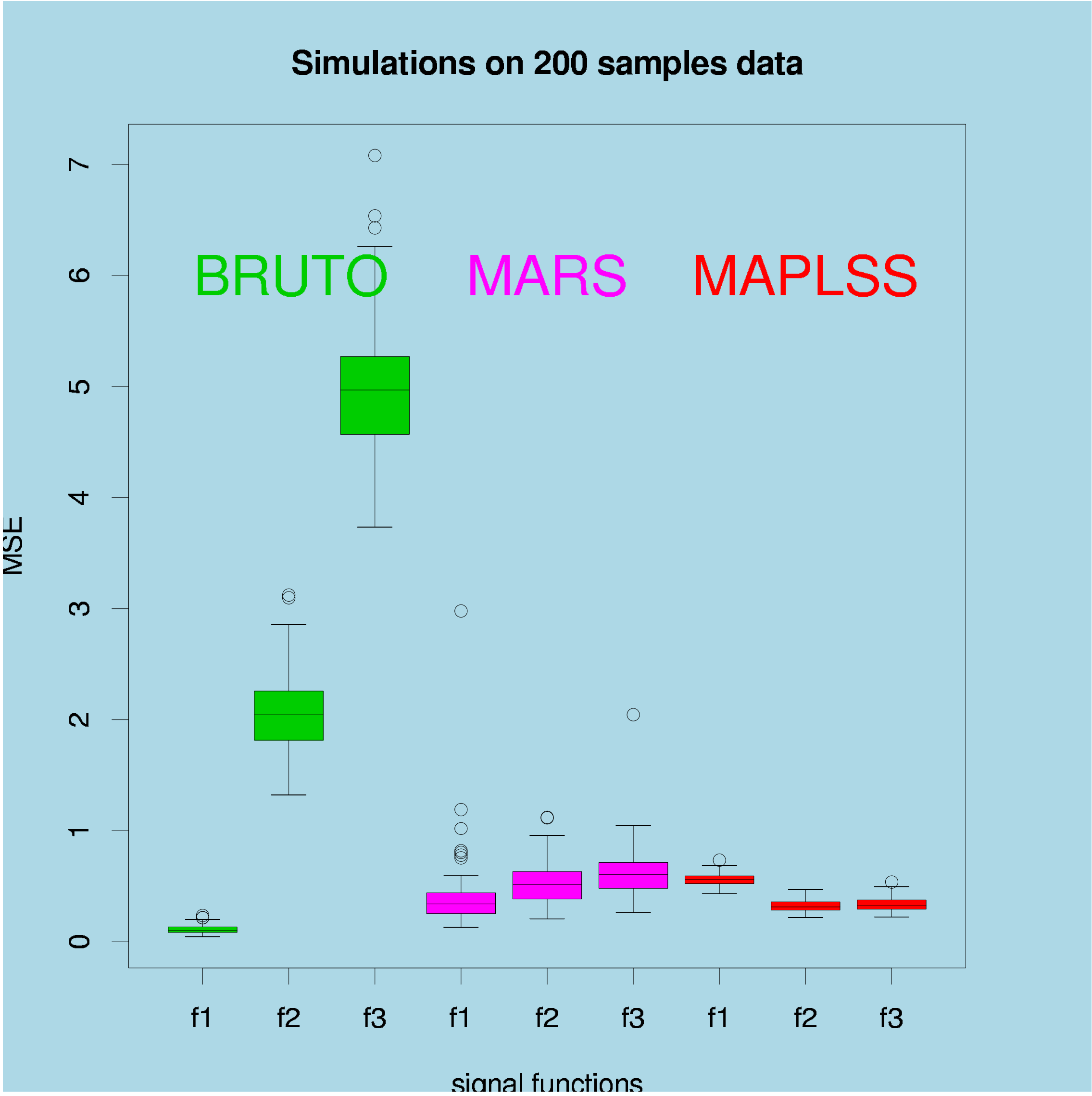
100 training data sets $(\boldsymbol{x}_i, y_i)_{i=1,n}$ were generated according to $n = 50,\ 100,\ 200$ and $j = 1, 2, 3$.

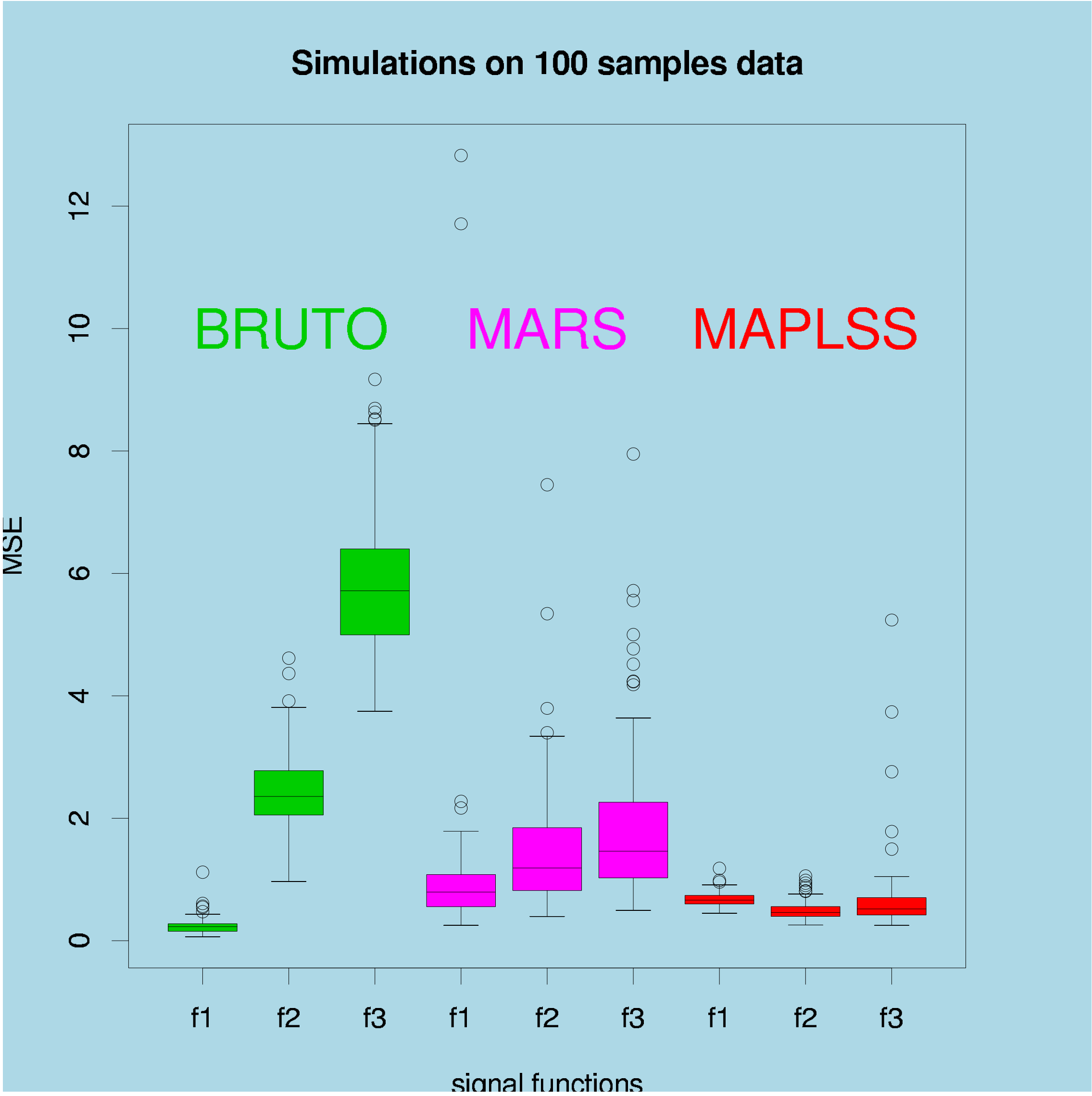
For each of the 900 data sets, the goodness-of-prediction of BRUTO, MARS and MAPLSS were measured on a new test set $(n_{test} = n)$ by computing the Mean Squared Error

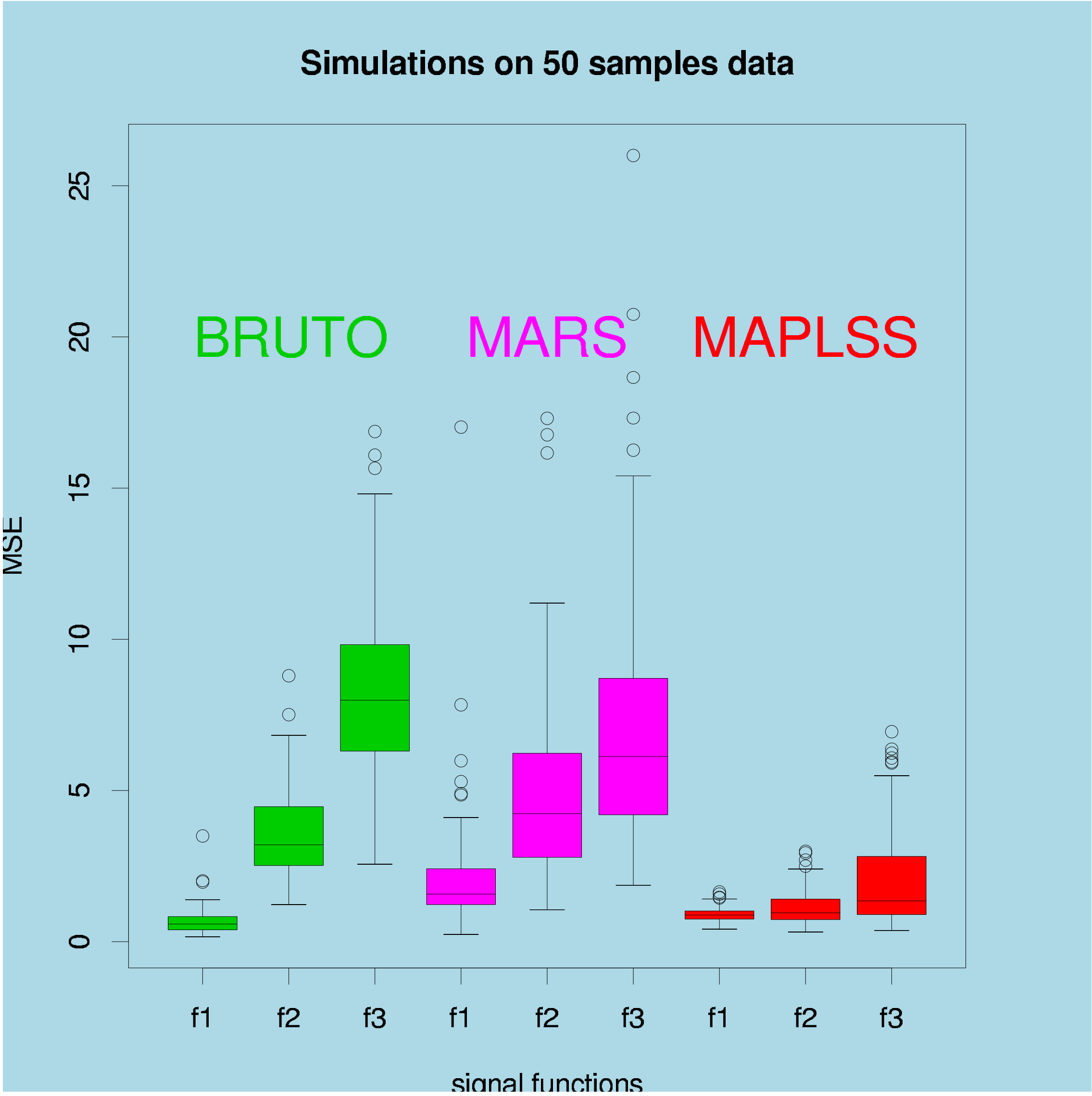
$$MSE = n_{test}^{-1} \sum_{i=1}^{n_{test}} (f_j(\boldsymbol{x}_i) - \hat{y}_i)^2.$$

These data sets were not only used to compare the domain of efficiency of the three methods but also to calibrate the MAPLSS tuning parameter allowing to accept or not one interaction

$$threshold = 20\%.$$


Simulations on 200 samples data
MSE
0
1
2
3
4
5
6
7
BRUTO
MARS
MAPLSS
f1
f2
f3
f1
f2
f3
f1
f2
f3
signal functions


Simulations on 100 samples data
MSE
0
2
4
6
8
10
12
BRUTO
MARS
MAPLSS
f1
f2
f3
f1
f2
f3
f1
f2
f3
signal functions

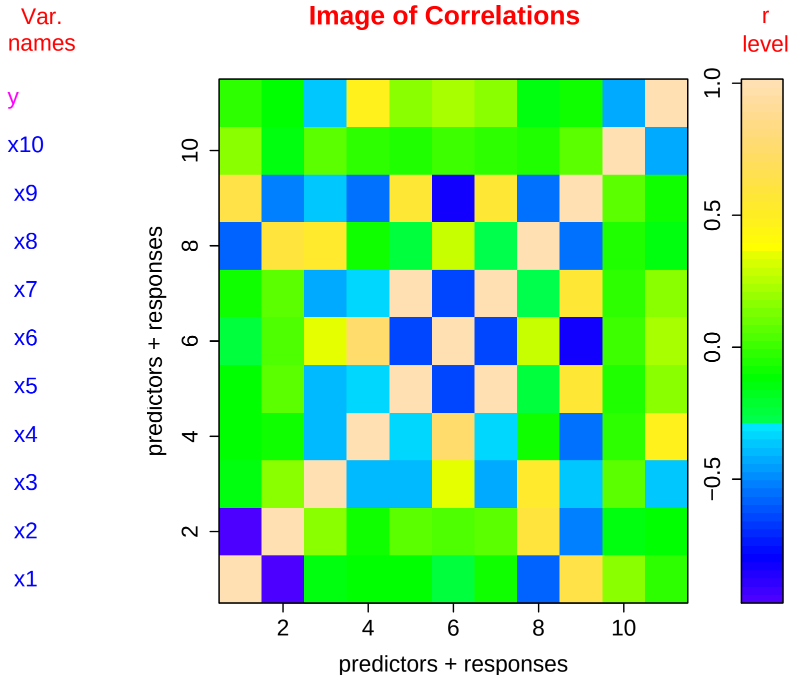

Simulations on 50 samples data
MSE
0
5
10
15
20
25
BRUTO
MARS
MAPLSS
f1
f2
f3
f1
f2
f3
f1
f2
f3
signal functions

Multi-collinearity and bivariate interaction: the ”chem” data [2, De Veaux et al.]

61 observations from a proprietary polymerization process.

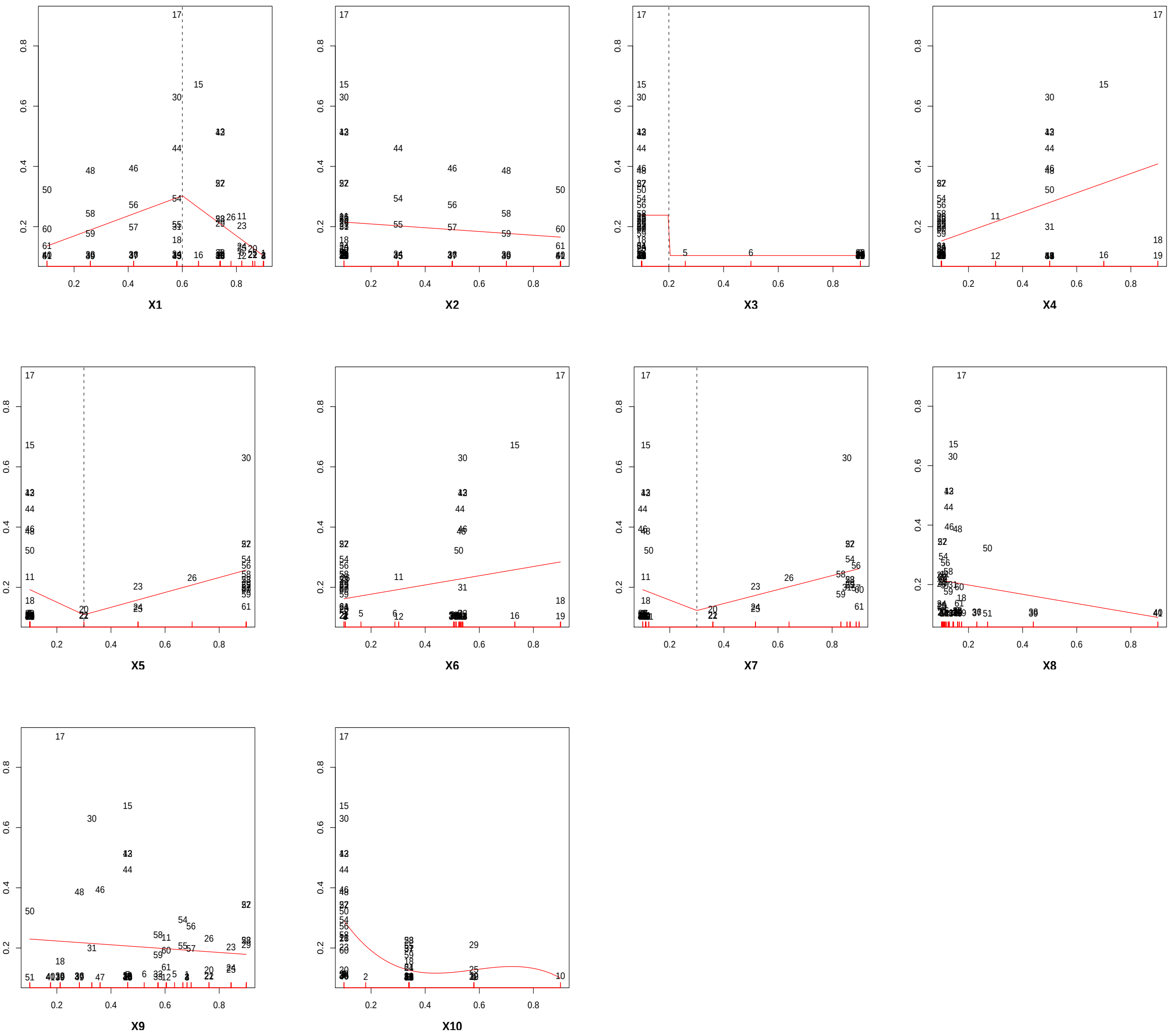
10 explanatory variables, $x_1, \ldots, x_{10}$ (inputs) and 1 response, y (output)

Due to the proprietary nature of the process, no more information could be disclosed.



$cor(x_5, x_7) = 1$, $cor(x_1, x_2) = -0.95$, $cor(x_6, x_9) = -0.82$

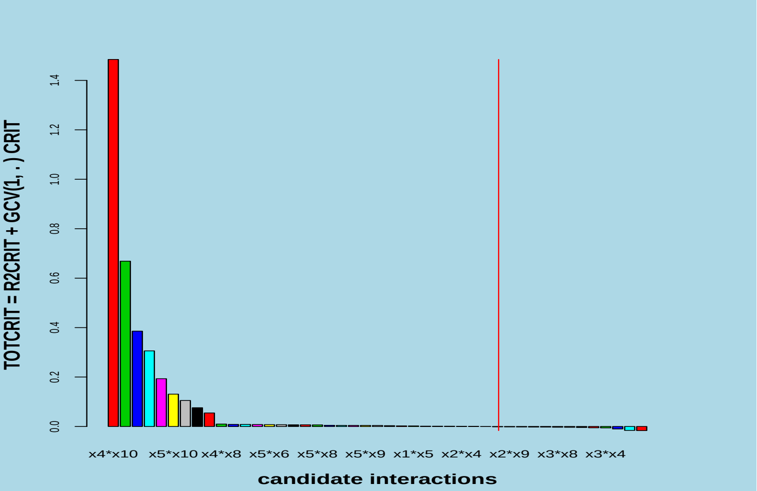
$\{(x_i,y)\}_i$ plots of chem data smoothed with L-S splines.



Towards a final model...

Phase 0: build a pure main effects model (PLSS)

Phase 1: select interactions candidate in decreasing order



Phase 2: Add interactions to the main effects model

candidate 1 : $x_4 * x_{10}$ ACCEPTED at 20 % relative GCV gain

	i	j	M	GCV	% rel. GCV gain
$x_4 * x_{10}$	4	10	9	0.04820853	89.22

candidate 2 : $x_6 * x_{10}$ NOT ACCEPTED at 20 % relative GCV gain

	i	j	M	GCV	% rel. GCV gain
$x_6 * x_{10}$	6	10	9	0.04757621	1.31

Continue anyway exploring (y/n)?1: n

Phase 3: Evaluate the PRESS(0.02,7)=0.0584 and ANOVA terms

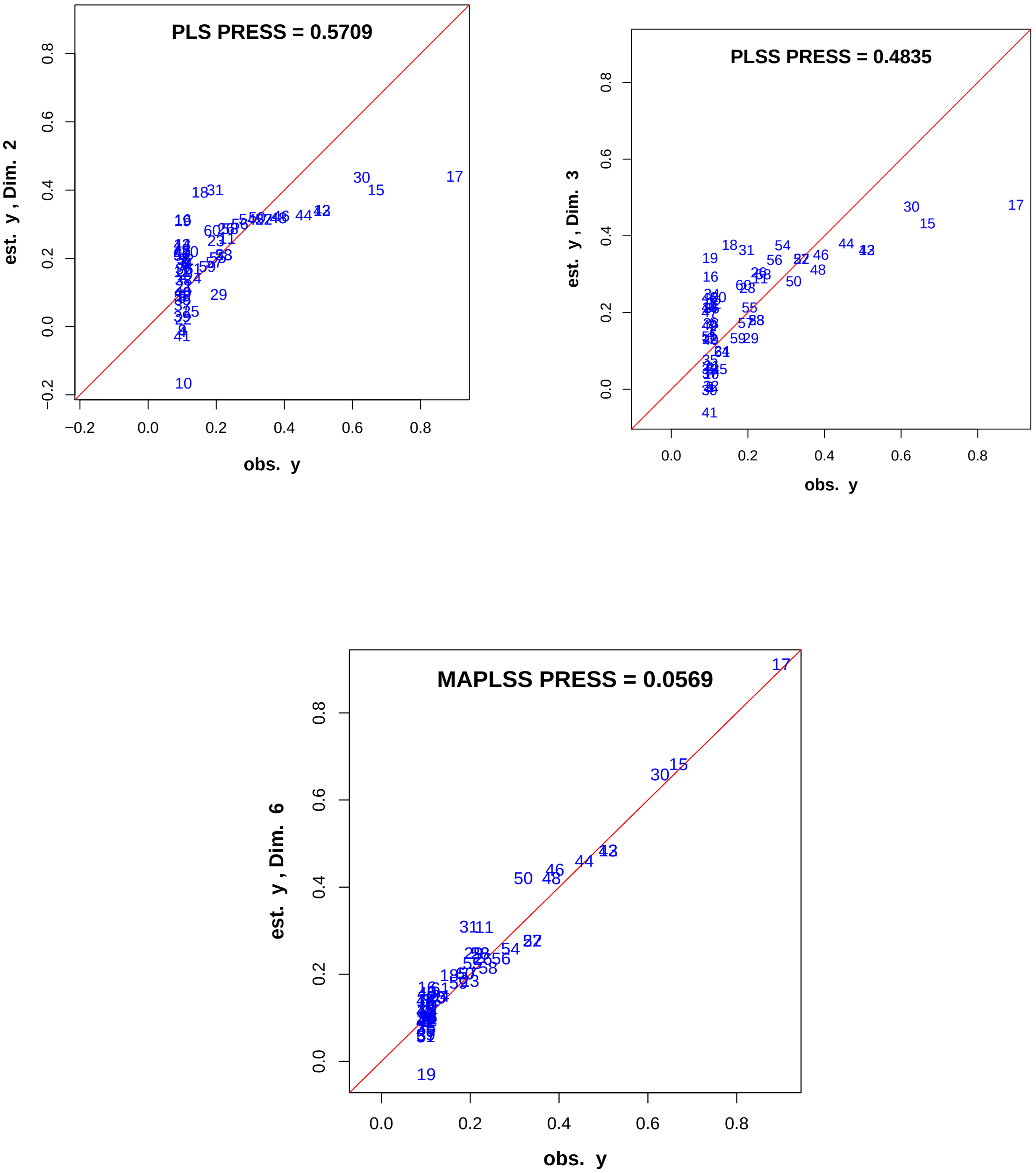
Range of the transformed predictors in descending order

$x_4 * x_{10}$	x_{10}	x_8	x_5	x_7	x_2	x_6	x_4	x_9	x_1	x_3
3.8238	1.7266	0.6202	0.5647	0.4616	0.4163	0.266	0.2106	0.188	0.1457	0.0762

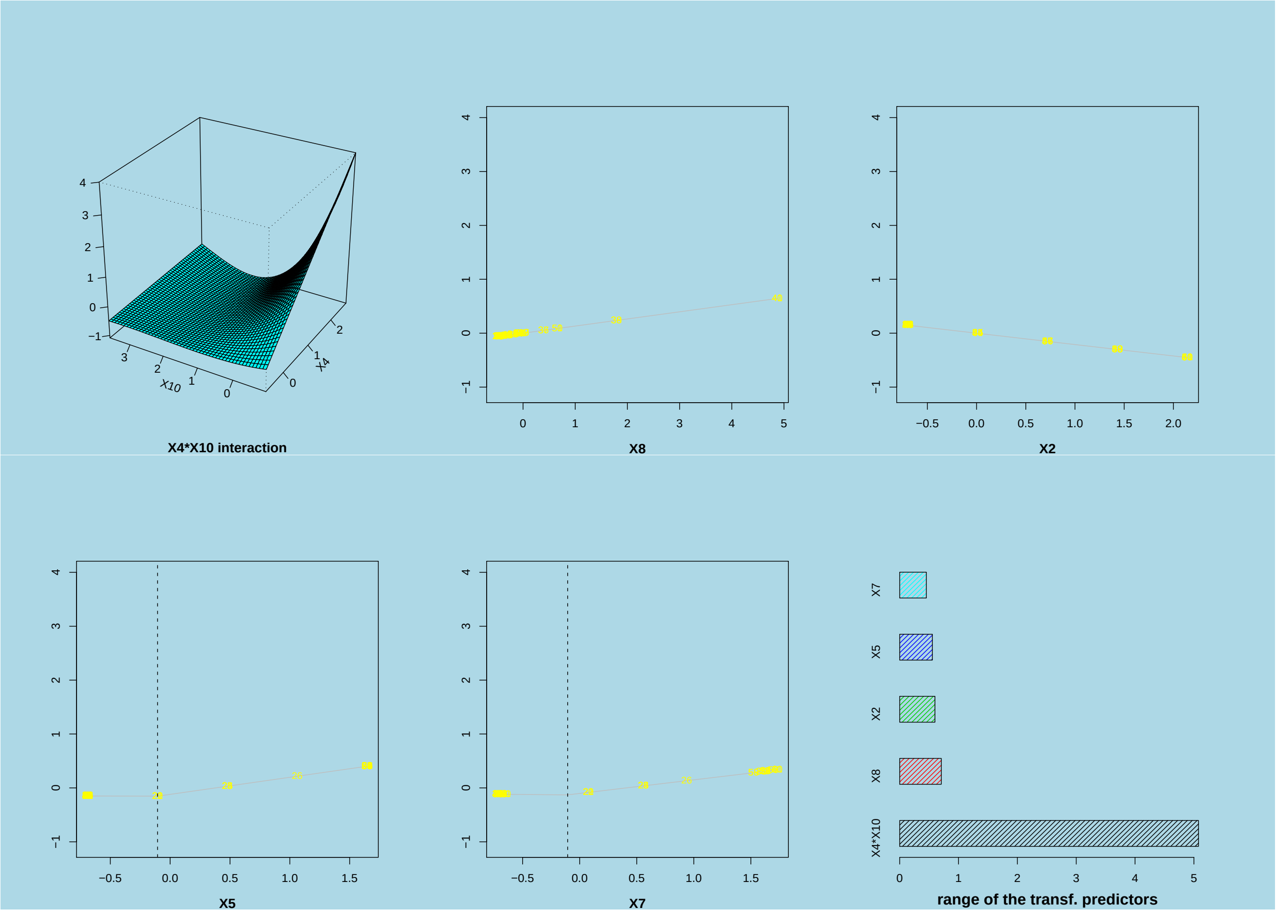
Phase 4: Prune low ANOVA terms. Final PRESS(0.02,6)=0.0569

$x_4 * x_{10}$	x_8	x_2	x_5	x_7
5.11868	0.55858	0.55846	0.53523	0.42331

Leave-one-out predicted versus observed y samples.



y-ANOVA plots ordered from left to right and up to down according to the vertical ranges.



References

[1] C. De Boor. *A Practical Guide to Splines*, Springer-Verlag, Berlin, 1978.

[2] R.D. De Veaux, D.C. Psichoggios and L.H. Ungar. *A comparison of two nonparametric estimation schemes: MARS and neural networks*,Computers chem. Engng, Vol. 17, No. 8, pp. 819-837, 1993

[3] J.F. Durand. *Local Polynomial Additive Regression through PLS and Splines: PLSS*, Chemometrics and Intelligent Laboratory Systems 58, 235-246, 2001.

[4] J. F. Durand and R. Lombardo. *Interactions terms in nonlinear PLS via additive spline transformations.* In "Studies in Classification, Data Analysis, and Knowledge Organization", Springer-Verlag, 2003.

[5] P.H.C. Eilers and B.D. Marx *Flexible smoothing with B-splines and Penalties, (with discussion).* Statistical Science, 19, 89-121, 1996.

[6] J.H. Friedman. *Multivariate Adaptive Regression Splines, (with discussion).* The Annals of Statistics, 19, 1-123, 1991.

[7] J.H. Friedman. *Greedy function approximation: a gradient boosting machine.* The Annals of Statistics, 29, 5, 1189-1232, 2001.

[8] A. Gifi. *Non Linear Multivariate Analysis*, J. Wiley & Sons, Chichester, 1990.

[9] T. Hastie and R. Tibshirani. Generalized additive models. Chapman and Hall, London, 1990.

[10] R. Lombardo, J. F. Durand and D. De Veaux. *Multivariate Additive Partial Least-Squares Splines, MAPLSS.* Submitted.

[11] N. Molinari, J.F. Durand and R. Sabatier. *Bounded Optimal knots for Regression Splines.* Computational Statistics & Data Analysis, 2, 159-178, 2004.

[12] C.J. Stone. *Additive regression and other nonparametric models.* The Annals of Statistics, 13, 689-705, 1985.

[13] H. Wold. *Estimation of principal components and related models by iterative least squares.* In Multivariate Analysis, (Eds.) P.R. Krishnaiah, New York: Academic Press, 391-420, 1966.

[14] S. Wold., H. Martens and H. Wold. *The mutivariate calibration problem in chemistry solved by PLS method.* In: A. Ruhe, B. Kagstrom (Eds), Lecture Notes in Mathematics, Proceedings of the Conference on Matrix Pencils, Springer-Verlag, Heidelberg, 286-293, 1983.