Boosted Partial Least-Squares Regression

Jean-François Durand

Montpellier II University, France

E-Mail: support@jf-durand-pls.com

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. Boosted PLS regression
   – What is $L_2$ Boosting?
   – Ordinary PLS viewed as a $L_2$ 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
      * Example 1: Multi-collinearity and outliers, the orange juice data
      * Example 2: The Fisher iris data revisited by PLS boosting
   – MAPLSS to capture interactions
      * The ANOVA type model for main effects and interactions
      * The building-model stage
      * Example 3: Comparison between MAPLSS, MARS and BRUTO on simulated data
      * Example 4: 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 [19, 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 [11, Hastie et al.], [8, J.H. Friedman], by considering PLS components as the base learners.
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, [4], [5], [12].

The key point of this nonlinear approach was inspired by the book of A. Gifi [9] 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 $B = [B_1 | \ldots | B_p]$, while capturing main effects plus bivariate interactions leads to

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

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

II. Short introduction to splines

Few words on smoothing splines

Consider the signal plus noise model
\[ y_i = s(x_i) + \varepsilon_i, \quad i = 1 \ldots n, \quad x_1 < \ldots < x_n \in [0, 1] \]
\((\varepsilon_1 \ldots , \varepsilon_n) \sim N(0, \sigma^2 I_{n \times n}), \sigma^2\) is unknown and \(s \in W^m_2[0, 1]\)

\[ W^m_2[0, 1] = \{ s/s^{(l)} \mid l = 1, \ldots , m - 1 \text{ are absolutely continuous, and } s^{(m)} \in L_2[0, 1] \}. \]

The smoothing spline estimator of \(s\) is
\[
s_\lambda = \arg \min_{\mu \in W^m_2[0,1]} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu(x_i))^2 + \lambda \int_0^1 [\mu^{(m)}(t)]^2 dt; \quad \lambda > 0
\]

usually \(m = 2\), to penalize convexity and overfitting.

The solution \(s_\lambda\) 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 $\lambda$ 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 $\beta$ of the coordinate values is to be estimated by a regression method.
Two sets of basis functions

- The truncated power functions

\[ x \rightarrow (x - \tau)^d_+ \]

When knots are distinct, 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 $j = 1, \ldots, m + K$,

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

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

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]

$$B_j^1(x) = \begin{cases} 1 & \text{if } \tau_j \leq x \leq \tau_{j+1}, \\ 0 & \text{otherwise}, \end{cases}$$

For $k = 2, \ldots, m$,

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

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

**R-package:**

```r
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?
```

Boosted PLS Regression: Séminaire J.P. Fénelon 2008-10
The attractive $B$-splines family for coding data

\[ S(2, \emptyset, [0, 5]) \quad (a), \quad S(3, \emptyset, [0, 5]) \quad (b) \quad \text{and} \quad S(1, \{1, 2, 4\}, [0, 5]) \, . \]
B-splines for $S(2, \{1, 2, 4\}, [0, 5])$ and $S(3, \{1, 2, 4\}, [0, 5])$.
• **Local support:**

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

♥ → 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:**

\[
0 \leq B^m_i(x) \leq 1 \quad (1) \quad \text{and} \quad \sum_{i=1}^{m+K} B^m_i(x) = 1 \quad (2).
\]

♥ → \(B^m_i(x)\) measures the degree of membership of \(x\) to \([\tau_i, \tau_{i+m}]\).

The set \(\{[\tau_i, \tau_{i+m}] | i = 1, \ldots, m+K\}\) 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 \(\tau_i\), \(0 \leq m_i \leq m\), then the first \(m - 1 - m_i\) right and left derivatives are equal,

\[
s_-(j)(\tau_i) = s_+(j)(\tau_i), \quad j \leq m - 1 - m_i.
\]

\[
m_i = m \quad \Rightarrow \text{discontinuity at } \tau_i
\]

\[
m_i = 1 \quad \Rightarrow \text{locally } C^{m-2} \text{ at } \tau_i.
\]
• **Coding a variable** \( x \) **through** \( B \)-splines

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

\[
\{ B_j^m(x) \mid j = 1, \ldots, m + K \} \text{ usual basis}
\]

\[
\{ 1, B_j^m(x) \mid j = 2, \ldots, m + K \} \text{ , modified basis.}
\]

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

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

the \underline{complete} \( n \times (m + K) \), or \underline{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

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

A bivariate regression spline split into the ANOVA decomposition:

\[
s(x, z) = \beta_1 + \sum_{j \in I_1} \beta_1^j B_1^j(x) + \sum_{j \in I_2} \beta_2^j B_2^j(z) + \sum_{i \in I_1} \sum_{j \in I_2} \beta_{1,2}^{i, j} B_1^i(x) B_2^j(z)
\]

- **Main effects** \(s^1\) and \(s^2\) in \(x\) and \(z\)
  \[ s^1(x) = \sum_{j \in I_1} \beta_1^j B_1^j(x) \quad s^2(z) = \sum_{j \in I_2} \beta_2^j B_2^j(z) \]

- **Interaction part** \(s^{12}\)
  \[ s^{12}(x, z) = \sum_{i \in I_1} \sum_{j \in I_2} \beta_{1,2}^{i, j} B_1^i(x) B_2^j(z) \]

How to measure the importance of an ANOVA term?

- by the range of the transformation (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.

\(\text{ncol}(B_1) = 10, \quad \text{ncol}(B_2) = 10, \quad \Rightarrow \quad \text{ncol}(B_{1,2}) = 100.\)
The Least-Squares Splines (LSS) [14, 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 $X$, $B = [B_1| \ldots | B_p]$, leads to $q$ separate additive spline models, $j = 1, \ldots, q$, 

$$
\hat{y}^j = s_{j,1}(x^1) + \ldots + s_{j,p}(x^p)
$$

through 

$$
\text{LSS}(X, Y) \equiv \text{OLS}(B, Y) \iff \hat{Y} = HY = B\hat{\beta} = \sum_i B_i\hat{\beta}_i
$$

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 [7, Friedman] proposes to remedy ♠**: automatic ↗ ↘ procedure to select knots and high order interactions through linear truncated power functions.

♥* EBOK [13, Molinari et al.]: $K$ fixed, optimal location of knots.

○ others....

Boosted PLS Regression: Séminaire J.P. Fénelon 2008-16
The Penalized Least-Squares Splines
[6, 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 ♠* by using a large number of equidistant knots, [6] proposed to penalize the \(L_2\) cost by a penalty \((\lambda)\) 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)

example : \(r = ncol(B) = 5\)

\[
D_2 = \begin{bmatrix}
1 & -2 & 1 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 \\
0 & 0 & 1 & -2 & 1
\end{bmatrix}
\]

Following [10, T. Hastie and R. Tibshirani]

\[
trace(H) = \text{effective dimension of the smoother}
\]
so that the Generalized Cross-Validation index

\[ GCV(\lambda, k) = \frac{\text{var}(\text{residuals})}{(1 - \text{trace}(H/n))^2} \]

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

One example: \((x_i, y_i)_{i=1,100}, \quad x_i \sim U[0, 2\pi], \quad y_i = \sin(x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, 0.5)\)

\[ \text{GCV} = 0.2768 \]

\(L-S\) Splines (red), signal (dotted), vertical lines indicate the location of knots
$P$-splines (red) with $\lambda \in \{0.2, 1.5, 3, 30\}$. 
III. Boosted PLS regression

What is $L_2$ Boosting? [17, Tukey] ("Twicing"), [11, Hastie et al.], [8, Friedman], [2, Bühlmann and Bin Yu]

The training sample: $\{y_i, x_i\}_1^n$, $y \in \mathbb{R}$, $x \in \mathbb{R}^p$, centered to estimate the function $F^*$ restricted to be of "additive" type,

$$F(x; \{\alpha_m, \theta_m\}_{1}^{M}) = \sum_{m=1}^{M} \alpha_m h(x, \theta_m),$$

that minimize the expected $L_2$ cost

$$\mathbb{E}[C(y, F(x))] \quad C(y, F) = (y - F)^2 / 2.$$

• the base learner $h(x, \theta)$

$\mapsto$ a function of the input variables $x$ characterized by parameters

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

$\mapsto$ neural nets, wavelets, splines, regression trees...

but also, latent variables,

$\theta \in \mathbb{R}^p \quad , \quad t = h(x, \theta) = < \theta, x >.$

• $M$, the "dimension" of the additive model.
Boosting:
a stagewise functional gradient descent method with respect to $F$.

$L_2$ boost algorithm:

$$F_0(x) = 0$$

For $m = 1$ to $M$ do

$$\tilde{y}_i = -\frac{\partial C(y_i,F)}{\partial F}|_{F=F_{m-1}(x_i)} = y_i - F_{m-1}(x_i), \quad i = 1, n,$$

$$(\alpha_m, \theta_m) = \arg\min_{\alpha, \theta} \sum_{i=1}^{n} [\tilde{y}_i - \alpha h(x_i; \theta)]^2 \quad (*)$$

$$F_m(x) = F_{m-1}(x) + \alpha_m h(x; \theta_m)$$

endFor

Extended $L_2$ boost algorithm:

Replace $(*)$ by

Criterion or procedure to construct $\theta_m$ from $\{(x_i, y_i)\}_{i=1,n}, \quad (*1)$

$$\alpha_m = \arg\min_{\alpha} \sum_{i=1}^{n} [\tilde{y}_i - \alpha h(x_i; \theta_m)]^2. \quad (\ast 2)$$

To summarize, $L_2$ boosting is characterized by

- choosing the type of the base learner
- repeated least-squares fitting of residuals
- fitting the data by a linear combination of the base learners.
**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

$$cov(x, y) = \langle x, y \rangle_D = y'Dx$$

and

$$var(x) = \|x\|^2_D.$$

**The algorithm** [18, H. Wold],[19, 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.

<table>
<thead>
<tr>
<th>PLS</th>
<th>$X_{(0)} = X$, $Y_{(0)} = Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step $m$</strong></td>
<td></td>
</tr>
<tr>
<td>$m=1, \ldots, M$</td>
<td></td>
</tr>
</tbody>
</table>
| **(1) base learner** | constructing $t^m \in \text{span}(X_{(m-1)})$ | $t = X_{(m-1)}w$, $u = Yv$

$(w^m, v^m) = \arg \max_{w, v} \text{cov}(t, u)$

$t^m = X_{(m-1)}w^m$, $u^m = Yv^m$

$X_{(m)} = X_{(m-1)} - H_m X_{(m-1)}$

| **(2) Y-residuals** | $Y_{(m)} = Y_{(m-1)} - H_m Y_{(m-1)}$ |
where the so-called linear PLS learner,

\[ H_m = \Pi_{t_m}^D = t_m t_m' D / \| t_m \|_D^2 \]

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 \( X'_{(m-1)} D Y \).

Let \( (\lambda_m, w_m, v_m) \) be the triple corresponding to the largest (first) singular value \( \lambda_m \), then

\[
  t_m = X_{(m-1)} w_m, \quad u_m = Y v_m, \quad \lambda_m = \text{cov}(t_m, u_m).
\]

In the one-response case, \( v_m = 1 \) and \( w_m = X'_{(m-1)} D Y / \| X'_{(m-1)} D Y \| \).
Components \( \{t^m\} \)

- belong to \( \text{span}(X) \), that is \( t^m = X \theta^m \)
- are mutually \( D \)-orthogonal \( t^{m_1'}Dt^{m_2} = 0, \ m_1 \neq m_2 \).
- Partial Regressions of pseudo variables give the same results as LS regressions of the original variables on the components.

\[
\hat{X}_m = H_m X_{(m-1)} = H_m X = t^m p^{m'} \\
\hat{Y}_m = H_m Y_{(m-1)} = H_m Y = t^m \alpha^{m'}
\]
The linear model with respect to \( \{ t^1, \ldots, t^M \} \)

Denoting \( T^M = [t^1 \ldots t^M]_{n \times M} \)

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

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

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

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

\[
Y = t^1 \alpha^1' + \ldots + t^M \alpha^M' + Y(M) .
\]

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

\[
\text{PLS}(X, Y) \equiv OLS(X, Y) ,
\]

thus providing an upper bound for the number of components

\[
1 \leq M \leq \text{rank}(X) .
\]

PCA of \( X \) is the ”self”-PLS regression of \( X \) onto itself

\[
\text{PLS}(X, Y = X) \equiv \text{PCA}(X) .
\]
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 ($V = X'DX$)-orthogonal basis to $\text{span}(X')$

$$<t^m_1, t^m_2>_D = <\theta^m_1, \theta^m_2>_V = 0$$

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

- $V$-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

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

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

$$\beta(0) = 0$$

$$\beta(m) = \beta(m-1) + \theta^m \theta^{m'} X'DY / \|\theta^m\|^2_V.$$
The building-model stage: choosing $M$

- Cross-Validation criterion: PRESS

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

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

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

![Graph showing PRESS vs. model dimension](image-url)

opt. Dim. 3 , $y$ PRESS = 0.03 (1 out)
• A surrogate to CV: Generalized Cross-Validation (GCV)

\[ m = 1, \ldots, \text{rank}(X); \quad \alpha = \text{penalty term (default 1)} \]

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

Calibration of the penalty \( \alpha \): \( GCV(\alpha = 1.7, M = 3) = 0.29 \)
PLS Splines (PLSS): a main effects additive model [4, 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

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

$$t = h(x, \theta) = \sum_{i=1}^{p} \sum_{k=1}^{r_i} \theta^i_k B^i_k(x^i) = \sum_{i=1}^{p} h_i(x^i).$$

- Tuning parameters: the spline spaces for each predictor
  the model dimension $M \mapsto \text{crossvalidation}$

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

$$\hat{y}_M^j = s_{M1}^{j1}(x^1) + \ldots + s_{Mp}^{jp}(x^p)$$

$\heartsuit$ if $M = \text{rank}(B)$ then PLSS$(X, Y) = \text{LSS}(X, Y)$ if it exists

$\heartsuit$ PLSS$(X, Y = B) \equiv \text{PLS}(B, B) = \text{NL-PCA}(X)$, [9, Gifi]

$\heartsuit$ efficient with low ratio observations/(column dimension of $B$)

$\heartsuit$ efficient in the multi-collinear context for predictors (concurvity)

$\heartsuit$ robust against extreme values of predictors (local polynomials)
no automatic procedure for choosing spline parameters
no interaction terms involved

Choosing the tuning parameters
1. For each predictor: degree, number and location of knots
2. the number of PLS components $\mapsto$ cross validation

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 $GCV$. 

Boosted PLS Regression: Séminaire J.P. Fénelon 2008-30
Example 1: Multi-collinearity, nonlinearity and outliers, the orange juice data

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

Both sample matrices are standardized.

Here,
\(n = 24\) orange juices: A, B, C, ..., X
\(p = 10, q = 1\),

<table>
<thead>
<tr>
<th>PREDICTORS</th>
<th>sensorial RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COND (Conductivity)</td>
<td>Heavy</td>
</tr>
<tr>
<td>SiO2</td>
<td></td>
</tr>
<tr>
<td>Na</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
</tr>
<tr>
<td>Ca</td>
<td></td>
</tr>
<tr>
<td>Mg</td>
<td></td>
</tr>
<tr>
<td>Cl</td>
<td></td>
</tr>
<tr>
<td>HCO3</td>
<td></td>
</tr>
<tr>
<td>SO4</td>
<td></td>
</tr>
<tr>
<td>Sum</td>
<td></td>
</tr>
</tbody>
</table>

collinearity: \(SUM = SiO2 + \ldots + SO4\)
biplots between Heavy and the predictors with 'lowess'

Boosted PLS Regression: Séminaire J.P. Fénelon 2008-32
Linear PLS on the orange juice data

opt. Dim. 2 , Heavy PRESS = 0.32 ( 2 out )

\[ R^2(1) = 0.754 \quad PRESS(0.1, 1) = 0.32 \]

Retained dimension : \( M = 1 \)
Boosted PLS Regression: Séminaire J.P. Fénelon 2008-34
**PLSS on the orange juice data**

degree=2 for all predictors

Table of selected knots for the predictors

<table>
<thead>
<tr>
<th>COND</th>
<th>SiO_2</th>
<th>Na</th>
<th>K</th>
<th>Ca</th>
<th>Mg</th>
<th>Cl</th>
<th>SO_4</th>
<th>HCO_3</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>10</td>
<td>10</td>
<td>2.5</td>
<td>160</td>
<td>40</td>
<td>4</td>
<td>400</td>
<td>100</td>
<td>600</td>
</tr>
<tr>
<td>1600</td>
<td>20</td>
<td>40</td>
<td>5</td>
<td>400</td>
<td>110</td>
<td>11</td>
<td>1700</td>
<td>300</td>
<td>2600</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>500</td>
</tr>
</tbody>
</table>

opt. Dim. 2 , Heavy PRESS = 0.154 ( 2 out )

\[
R^2(2) = 0.917 \quad PRESS(0.1, 2) = 0.154
\]
Boosted PLS Regression: Séminaire J.P. Fénelon 2008-36
Nonlinear 2-D component scatterplots

A PLSS component $t$ is additively decomposed by ANOVA terms

$$t = B\theta = \sum_{i=1}^{p} B_i \theta_i = \sum_{i=1}^{p} h_i(X^i)$$

where $\theta_i$ is the sub-vector of $\theta$ associated to the block $B_i$, thus allowing to interpret $t$ by the predictors.

Because $r(t^1, Y) = 0.924$, then $h_i^1(X^i) \approx s^i_M(X^i)$ and one can use the Heavy ANOVA plots to explain the $t^1$ coordinates.
Example 2: The Fisher iris data revisited by PLS boosting

A multi-response PLS discriminant Analysis context, [15, Sjöström et al.], [16, Tenenhaus]:

\( p = 4 \) predictors: sepal width and length, petal width and length

\( q = 3 \) responses: the binary indicators of 3 Iris species, setosa (s), versicolor (c), virginica (v)

\( n = 150 \) Iris specimen (50 per species).

The \((t^1, t^2)\) scatterplot and the table of PLS diagnostics:

\[
\text{PRESS}(3) = 1.379, \quad 10\% \text{ out.}
\]
The splines: degree=2 and 3 equally spaced knots.

No relevant interaction was detected and the pure main effects model provides efficient diagnostics.

The \((t^1, t^2)\) scatterplot and table of PLSS diagnostics:

\[
\begin{array}{ccc}
\text{PRESS}(4) = 0.3551, & 10\% \text{ out.} \\
\end{array}
\]

How to interpret a latent variable with respect to the predictors?
The spline functions are ordered, from left to right and up to down, according to the range of the vertical values. Data are coded: setosa (1), versicolor (2), virginica (3). Dashed vertical lines indicate the location of knots.
To summarize: Petal Length and Width are the relevant factors for discrimination as well as the Sepal Length to a lesser degree.
Multivariate Additive PLS Splines (MAPLSS): capture of interactions
[5, Durand & Lombardo][12, Lombardo, Durand, De Veaux ]

The ANOVA type model for main effects and interactions

The centered main effects + interactions coding matrix:

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

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

\[ \text{MAPLSS}(X, Y) \equiv \text{PLS}(B, Y) \]

\[ t = h(x, \theta) = \sum_{i=1}^{p} \sum_{k=1}^{r_i} \theta_k^i B_k^i(x^i) + \sum_{\{i,i'\} \in I} \left[ \sum_{k=1}^{r_i} \sum_{l=1}^{r_i'} \theta_{k,l}^{i,i'} B_k^i(x^i) B_l^{i'}(x^{i'}) \right] . \]

\(q\) simultaneous models casted in the ANOVA decomposition

\[ j = 1, \ldots, q, \quad \hat{y}_M^j = \sum_{i=1}^{p} s_{M,i}^j(x^i) + \sum_{(i,i') \in I} s_{M,ii'}^j(x^i, x^{i'}) \]

PLSS with interactions shares the preceding \(\heartsuit\) properties

\(\spadesuit \heartsuit\) no automatic procedure for choosing spline parameters

Boosted PLS Regression: Séminaire J.P. Fénelon 2008-42
The building-model stage

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

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.

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

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 ($GCV_i < GCV_{i-1} - threshold \times GCV_{i-1}$)

3 prune ANOVA terms of lowest influence (backward phase)

criterion: the range of the values of the ANOVA function
Example 3: Comparison between MAPLSS, MARS and BRUTO on simulated data

BRUTO [10, Hastie & Tibshirani]: A multi-response additive model fitted by adaptive back-fitting using smoothing splines.

MARS[8, 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)

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

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

\[
f_3(x) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 20x_4x_5 + 0 \sum_{i=6}^{10} x_i
\]

not depending on the last five variables.
The model associated to the signal $f_j$

$$y_i = f_j(x_i) + \varepsilon_i, \quad i = 1, \ldots, n$$

$$x_i \sim U[0, 1]^{10} \quad \text{and} \quad \varepsilon_i \sim N[0, 1].$$

100 training data sets $(x_i, y_i)_{i=1,n}$ 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

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

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

\[ \text{threshold} = 20\%. \]
Simulations on 200 samples data

Boosted PLS Regression: Séminaire J.P. Fénelon 2008-46
Simulations on 100 samples data

Boosted PLS Regression: Séminaire J.P. Fénélon 2008-47
Example 4: Multi-collinearity and bivariate interaction, the "chem" data [3, 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.

\[
cor(x_5, x_7) = 1, \quad cor(x_1, x_2) = -0.95, \quad cor(x_6, x_9) = -0.82
\]
\{(x_i, y_i)\}_{i} \text{ 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: $x_4 \times x_{10}$ ACCEPTED at 20 % relative GCV gain

\begin{tabular}{cccc}
  i & j & M & GCV \\
  \hline
  4 & 10 & 9 & 0.04820853
\end{tabular}

\begin{tabular}{cccc}
  \% rel. GCV gain \\
  \hline
  89.22
\end{tabular}

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

\begin{tabular}{cccc}
  i & j & M & GCV \\
  \hline
  6 & 10 & 9 & 0.04757621
\end{tabular}

\begin{tabular}{cccc}
  \% rel. GCV gain \\
  \hline
  1.31
\end{tabular}

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

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

Range of the transformed predictors in descending order

\begin{tabular}{cccccccccccc}
  $x_4 \times x_{10}$ & $x_{10}$ & $x_8$ & $x_5$ & $x_7$ & $x_2$ & $x_6$ & $x_4$ & $x_9$ & $x_1$ & $x_3$ \\
  \hline
  3.8238 & 1.7266 & 0.6202 & 0.5647 & 0.4616 & 0.4163 & 0.266 & 0.2106 & 0.188 & 0.1457 & 0.0762
\end{tabular}

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

\begin{tabular}{cccccccc}
  $x_4 \times x_{10}$ & $x_8$ & $x_2$ & $x_5$ & $x_7$ \\
  \hline
  5.11868 & 0.55858 & 0.55846 & 0.53523 & 0.42331
\end{tabular}

Boosted PLS Regression: Séminaire J.P. Fénelon 2008-51
Leave-one-out predicted versus observed $y$ samples.

PLS PRESS = 0.5709

PLSS PRESS = 0.4835

MAPLSS PRESS = 0.0569
$y$-ANOVA plots ordered from left to right and up to down according to the vertical ranges.
References


Boosted PLS Regression: Séminaire J.P. Fénelon 2008-54