B-spline optimization with genetic algorithms for a non-linear PLS : Application to chemometrics and drug design

Christelle REYNES

Agrostat 2012 › 2 *Mars*

Context : PLS and non-linearity

PLS = a powerful method for regression
used in numerous applicative fields

Limitation : only *linear* relationships between inputs and outputs can be modelled.

$\Rightarrow$ need to introduce non-linearity for some datasets

Several way to introduce non-linearity, more or less Plusieurs façons d'introduire cette non-linéarité plus ou moins adaptive.

Chosen solution : *Individual and simultaneous* transformation of inputs thanks to B-spline functions.

Several parameters to be optimized for each inputs
(number and location of knots, spline degree)

$\Rightarrow$ Combinatorial optimization problem

$\Rightarrow$ Possibility to perform feature selection

Reminder about PLS : notations

Let $\mathbf{Y}$ be the $n \times q$ ($q \geq 1$) matrix containing the q response values for n observations.
Let $\mathbf{X}$ be the $n \times p$ ($p \geq 1$) matrix containing the p predictor values for n observations.

Purpose : Find successive linear combinations of predictors $\mathbf{t}_k$ ($k = 1, \ldots, A$) and responses $\mathbf{u}_k$ maximizing $cov(\mathbf{t}_k, \mathbf{u}_k)$, for instance with NIPALS algorithm (Tenenhaus, 1998).
A : number of retained components usually chosen by cross validation.

Reminder about B-spline functions

B-splines = piecewise polynomials

Three parameters to be chosen for each input x^i : d_i the polynomial degree, K_i the number of knots and the location of knots.

x^i transformation :

$$x^i \rightarrow \mathbf{B}^i$$

where $\mathbf{B}^i = [B_1^i, \ldots, B_{r_i}^i]$ is a $n \times r_i$ matrix with $r_i = (d_i + K_i + 1)$.

$\Rightarrow$ can fit various shapes

Reminder about B-spline functions

B-splines = piecewise polynomials

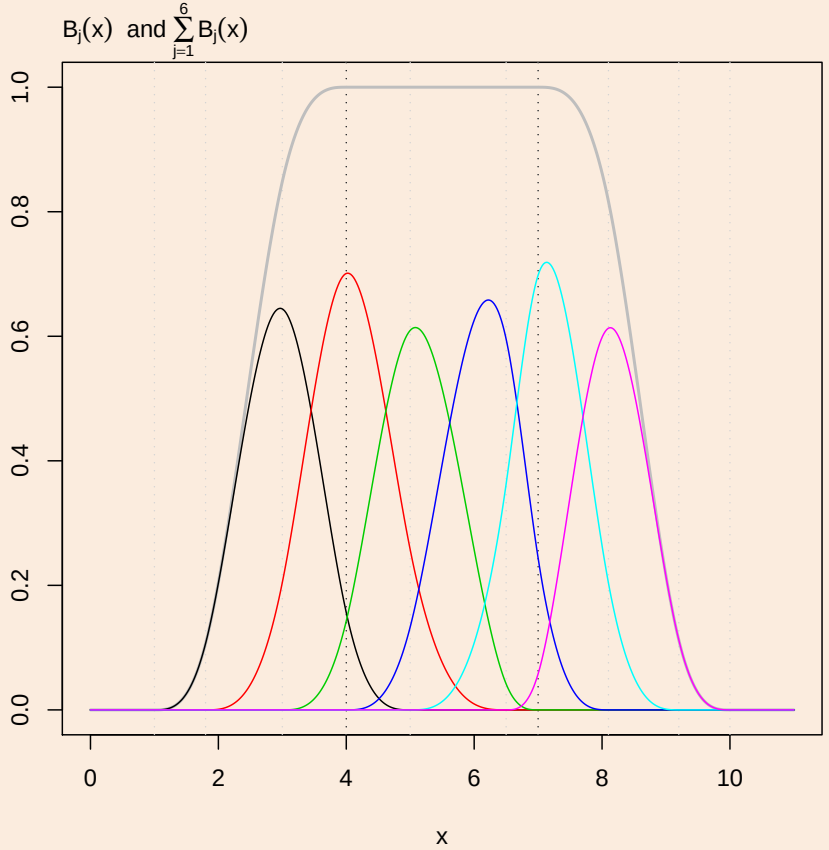

Illustration of B-spline basis for $d = 3$ and $K = 10$

Reminder about genetic algorithms

Genetic algorithm = metaheuristic mimicing processes of
natural evolution

Need to define a function allowing the quantification of solution quality
=fitness

Works on a *population* of potential solutions to the problem.

Three main and iterative steps :

- mutation : introduce hazard allowing to explore the solution space
- crossover : combines previously obtained characteristics
- selection : preferentially retains the most interesting solutions for next generations

AG-PLSS method

Purpose : Optimize the B-spline transformation to obtain a more efficient PLS model.

The usual PLS method is applied to $\mathbf{B}$, the juxtapostion of the $\mathbf{B}^i$ for $i \in \{1, \ldots, p\}$.

One obtains :

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

with $\hat{\mathbf{y}}_A^j$, rank A modelling of $\mathbf{Y}$ j-th column

$$\hat{f}_A^{j,i}(\mathbf{x}^i) = \sum_{k=1}^{r_i} \hat{a}_{k,A}^{j,i} \mathbf{B}_k^i(\mathbf{x}^i).$$

$\hat{a}_{k,A}^{j,i}$ are the usual coefficients of PLS applied to $\mathbf{B}$.

AG-PLSS method

The genetic algorithm allows to optimize the degree, the number and location of knots.
The PRESS is used to choose A the number of components.
Chosen fitness function for a solution s :

$$fit(s) = \frac{R^2_{train}(s) + R^2_{CV}(s)}{2} - \left(R^2_{train}(s) - R^2_{CV}(s)\right).$$

First part : model precision
Second part : over-fitting measurement.

Applications

Chemometrics example (Durand, 2001)

The data :

24 orange juices

10 mineralogical predictors

one sensory response

AG-PLSS results (Kmax=10, Dmax=3) :

6 components are obtained

TABLE: Obtained transformations

Var	1	2	3	4	5	6	7	8	9	10
Knot nb	**0**	**3**	1	**0**	2	**0**	**3**	0	0	0
Degree	1	1	1	1	2	1	1	1	1	1
$\rho(X_i, Y)$	**0.89**	**-0.23**	0.02	**-0.03**	0.84	**0.89**	**-0.01**	0.78	0.31	0.82
VIPmax	1.66	0.92	1.09	1.8	1.6	1.6	**2.5**	1.45	0.98	0.37

$\Rightarrow$ *parsimonious*, relevant and useful transformations

Chemometrics example (Durand, 2001)

Comparison with other *PLS-like* methods :

TABLE: Comparison of prediction quality for *Orange juices*.

	OLS	PLS	PLSS	AG-PLSS
R^2	0.9140	0.9044 ($A = 6$)	0.9139 ($A = 2$)	0.9953 ($A = 6$)
R^2_{CV}	0.5898	0.6047 ($A = 6$)	0.7189 ($A = 2$)	0.8175 ($A = 6$)

$\Rightarrow$ best results for training and cross-validation
$\Rightarrow$ least loss between training and cross-validation
$\Rightarrow$ interest and effectiveness of B-spline transformation optimisation

Chemometrics example (Durand, 2001)

Comparison with other non-linear methods :

TABLE: Comparison of prediction quality for *Orange juices*.

optimisation of parameters thanks to cross-validation

	AG-PLSS	SVM (gaussian)	Neural Network	Random Forest
R^2	0.9953	0.9253	0.7622	0.9562
R^2_{CV}	0.8175	0.6156	0.6989	0.7135

$\Rightarrow$ best results for training and cross-validation

$\Rightarrow$ interest of a specific transformation for each input

A *drug design* example : Vss prediction

Les données :

Vss : volume of distribution in steady state to be predicted

177 molecules (Gleeson *et al.*, 2006) :

- 138 molecules for training
- 39 molecules for testing

1666 descriptors of their 1D, 2D and 3D structure provided by e-Dragon software (http ://www.vcclab.org/lab/edragon/start.html)

A *drug design* example : Vss prediction

Feature selection process

- Non-specialized selection :
  - constant inputs removal
  - redundant inputs removal

  $\Rightarrow$ 526 predictors remain
- specialized selection :
  - selection of the most efficient inputs in PLS (VIPmax>2)
  - optimisation of an individual B-spline transformation of each input and selection of inputs achieving a real benefit through transformation

  $\Rightarrow$ 25 predictors remain

A *drug design* example : Vss prediction

AG-PLSS results :

degree 1 for each input

number of knots :

- no knots for 7 inputs
- 1 knot for 8 inputs
- 2 knots for 7 inputs
- 3 knots for 3 inputs

one component

results :

- $\rho(Y, \hat{Y})^2 = 0.6059$
- $\rho(Y, \hat{Y}_{CV})^2 = 0.5765$
- $\rho(Y_{test}, \hat{Y}_{test})^2 = 0.5695$

A *drug design* example : Vss prediction

Comparison with other methods :

parameters optimized thanks to cross-validation

	AG-PLSS	PLS	SVM	NN	RF
R^2	0.6059	0.6384	0.7956	0.8879	0.9078
R^2_{CV}	0.5765	0.5571	0.6212	0.5391	0.6066
R^2_{test}	0.5695	0.4710	0.2097	0.2821	0.3166

SVM : gaussian kernel, NN : neural network, RF : random forest.

$\Rightarrow$ best results on test sample

$\Rightarrow$ least loss of efficiency in validation

Conclusion and perspectives

- Difficulty of introducing appropriate non-linearity
- Interesting but complex to optimize individual transformations
- AG-PLSS really appropriate :
  - Chosen fitness $\Rightarrow$ parsimony and generalisability
  - Transformation only when required
  - Price : Computationaly intensive $\Rightarrow$ prior feature selection may be required
- Perspective : feature selection process to be optimized