Talanta Open 7 (2023) 100235

Contents lists available at ScienceDirect

Talanta Open

journal homepage: www.sciencedirect.com/journal/talanta-open

Two sides of the same coin: Kernel partial least-squares (KPLS) for linear and non-linear multivariate calibration. A tutorial

Franco Allegrini [a], Alejandro C. Olivieri [b,*]

[a] Calle 9 de Julio 2045 Dto. 6B, Rosario (2000), Argentina
[b] Departamento de Química Analítica, Facultad de Ciencias Bioquímicas y Farmacéuticas, Universidad Nacional de Rosario, Instituto de Química Rosario (CONICET-UNR), Suipacha 531 (2000) Rosario, Argentina

ARTICLE INFO

Original content: Data, MATLAB codes and R codes for "Kernel partial least-squares (KPLS) for multivariate calibration. A tutorial." (Original data)

Keywords:
Kernel partial least-squares
Multivariate calibration
Near infrared spectra
Fluorescence data
Detection of non-linearity

ABSTRACT

A tutorial is presented on the operation and properties of the non-linear multivariate regression model kernel partial least-squares (KPLS). After the discussion of a simple non-linear univariate problem, solved by regressing a dependent variable on the projection of an independent variable onto a set of Gaussian functions, the principles of KPLS are introduced for processing non-linear multivariate data. The following aspects are covered: (1) the estimation of the model sensitivity as a function of analyte concentration from error propagation concepts, (2) the proposal of a parameter measuring the degree of non-linearity, to avoid a black-and-white description of data sets as either linear or non-linear, (3) the use of the model parameters for variable selection. The application of KPLS to both simulated and experimental data sets is discussed, in the latter case involving near infrared spectra employed for the determination of quality parameters in foodstuff samples and fluorescence spectroscopic data for the study of systems of biological relevance. Computer codes written in the popular MATLAB and R environments are also provided.

1. Introduction

Since the first applications of multivariate models to instrumental data in analytical chemistry, the classification and treatment of the data according to their degree of linearity (or non-linearity) has been a persistent and complex problem to address. It is perhaps curious to notice that the strongest tendency has always been to develop and apply either linear or non-linear models, as if they belonged to two distinct and independent worlds. They have been rarely thought as "two sides of the same coin", and this is why the link between them has not been deeply studied.

When facing with a given set of data for analytical calibration purposes, it is customary to attempt an initial approach using a linear model, and if the results are not satisfactory, a more complex non-linear model is subsequently applied. This strategy is purely empirical and based on the fact that linear models are simpler and amenable to better interpretation, since they are supported by solid mathematical and statistical foundations. A more rational strategy employs a certain suitable test to the dataset in order to gather beforehand whether the relationship between the dependent and independent variables is linear or non-linear [1,2].

Today, partial least-squares (PLS) regression is probably the most widely used and popular first-order linear multivariate models [3]. PLS is sometimes compared to principal component regression (PCR) [4], and in most of cases the results in terms of prediction are statistically equivalent [5]. PLS operates by linearly transforming the original predictor variables into a space of latent variables through a projection that maximizes the covariance between the predictor and the variable to be predicted, generating a vector of weight loadings. This is the main difference with PCR, where the projection is made considering only the direction of maximum correlation between the elements of the predictor variables, i.e., the regression is applied to the result of the principal component decomposition of the data. All in all, in both cases the underlying idea is to generate parsimonious models, that is, models based on a low number of explanatory variables that maximize the content of useful information. The generation of these variables also allows one to deal with problems of collinearity and rank deficiency, which are very common, for example, in spectroscopic analytical data, where the signals measured at many sensors are usually included in the model, exceeding the number of calibration samples.

Both PLS and PCR may allow for certain deviations from the linearity in the data, usually through the inclusion of either more latent variables

* Corresponding author.
E-mail address: olivieri@iquir-conicet.gov.ar (A.C. Olivieri).

https://doi.org/10.1016/j.talo.2023.100235
Received 23 March 2023; Received in revised form 5 May 2023; Accepted 22 May 2023
Available online 23 May 2023

1. Pre-process the spectra if needed

2. For all samples:

```
k=exp(-sum((Xcal-x*ones(1,Ical)).^2)/r^2);
```
MATLAB

```
k <- exp(-(colSums((Xcal-x%*%ones(1,Ical))^2)/r^2))
```
R

3. Build a classical PLS model with calibration $\mathbf{K}_{cal}$ and $\mathbf{y}_{cal}$

4. Predict y_{test} from $\mathbf{k}_{test}$ in the test sample

Fig. 1. Description of the main programming activities for applying KPLS to a typical calibration problem. In the red and blue boxes, 'Xcal' is the calibration matrix of spectra, 'Ical' is the number of calibration samples, 'x' is a sample spectrum (whether calibration or test), and 'r' is the radius. The kernels represented by 'k' corresponding to calibration samples are grouped in the $\mathbf{K}_{cal}$ matrix and those for the test sample in a $\mathbf{k}_{test}$ vector. The red and blue boxes contain the required programming line for projecting the spectra onto the Gaussian space in both MATLAB and R (as indicated). The nominal calibration concentrations and the estimated one in the test samples are $\mathbf{y}_{cal}$ and y_{test} respectively.

or their squared values in the regression model [6]. When the level of non-linearity is pronounced, however, alternative tools should be employed. The following three are pertinent examples: (1) partition of the non-linear space into smaller subspaces of a linear nature, as in some LOCAL-type strategies [7], (2) resort to models that allow non-linear data as input, e.g., artificial neural networks (ANNs) [8], and (3) use of pre-processing methods to increase the data linearity, by projection of the data onto a non-linear space, as will be described below in detail.

LOCAL strategies have been studied and applied as an alternative to face some types of non-linearities in the datasets. Linear models like PLS and PCR, complemented by LOCAL approaches [9], may successfully deal with certain types of deviations from linearity in data, especially when the source of non-linearity stems from a large number and diversity of samples in a database, generating numerous clusters in the score plot. The idea behind these models is to maintain the advantages of PLS, such as simplicity, robustness, predictive ability, and stability, while adapting it to deviations from linearity by training a model for each test sample based on a selected number of locally close samples.

In the case of ANNs, the most commonly used and studied ones in analytical chemistry are multilayer perceptron ANNs (MLP ANNs) [10], which operate through a back propagation fitting mechanism. Another less popular network that has gained relevance in this area is the radial basis function (RBF) ANN [11]. The latter has a much simpler structure than MLP ANNs, as it only includes a single non-linear transformation through a radial basis function in the hidden layer, usually a Gaussian function, and a linear transformation in the output layer. Consequently, the adjustment of the interconnecting weights is performed by a simple least-squares fit. This ensures that the results are always the same, unlike in MLP back propagation, where the weights vary from run to run, depending on both the random seed to initialize the weights and on the monitoring sample set used for early stopping [10].

Another less studied but highly potential alternative is to combine elements of ANNs with PLS, in order to linearize non-linear problems and then take advantage of the highly desirable characteristics of PLS. One of the first attempts to combine ANNs with PLS to solve data for non-linear chemical systems was reported by Walczak and Massart [12]. In this latter article, the difficulties that arise in RBF networks when solving overdetermined systems were considered, such as choosing the number of radial basis functions (the number of nodes in the hidden layer) and their distribution in the input data space. This led to the need of applying PLS in the predictive stage, instead of using direct least-squares, leading to a hybrid RBF-PLS model [12]. Subsequently, this approach was developed from a more theoretical point of view by Rosipal and Trejo [13], who provided a detailed description of the mathematical and statistical foundations of a model called kernel PLS (KPLS). Some years later, Kim et al. [14] described kernel PLS in detail, in combination with orthogonal signal correction (OSC) to treat several analytical multivariate data sets of different origins.

A literature search reveals a relatively small number of publications on KPLS as applied to analytical calibration. In the last five years, for example, only a few reports were specifically published on calibration with near infrared spectra [15–22] and calibration transfer [23]. Following the tendency of keeping models as simple as possible even for complex data sets, the present report is intended as a rehabilitation of KPLS, showing its main characteristics and advantages, and presenting the method in the context of analytical chemistry as a tutorial. Through the analysis of simulated data, some important theoretical aspects will be tackled. Relevant advantages will be discussed when applying this model to solve practical problems in analytical chemistry, such as predicting concentrations based on multivariate spectroscopic data from

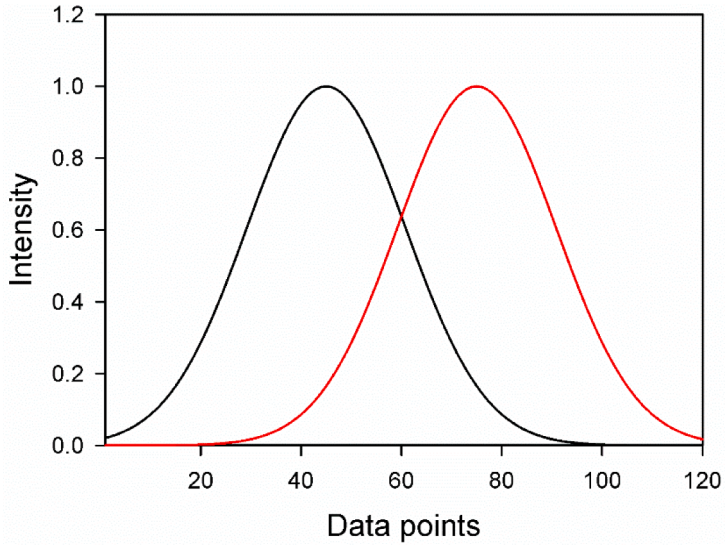


Fig. 2. Pure component profiles for building the simulated data sets: black line, analyte of interest, red line, interferent.

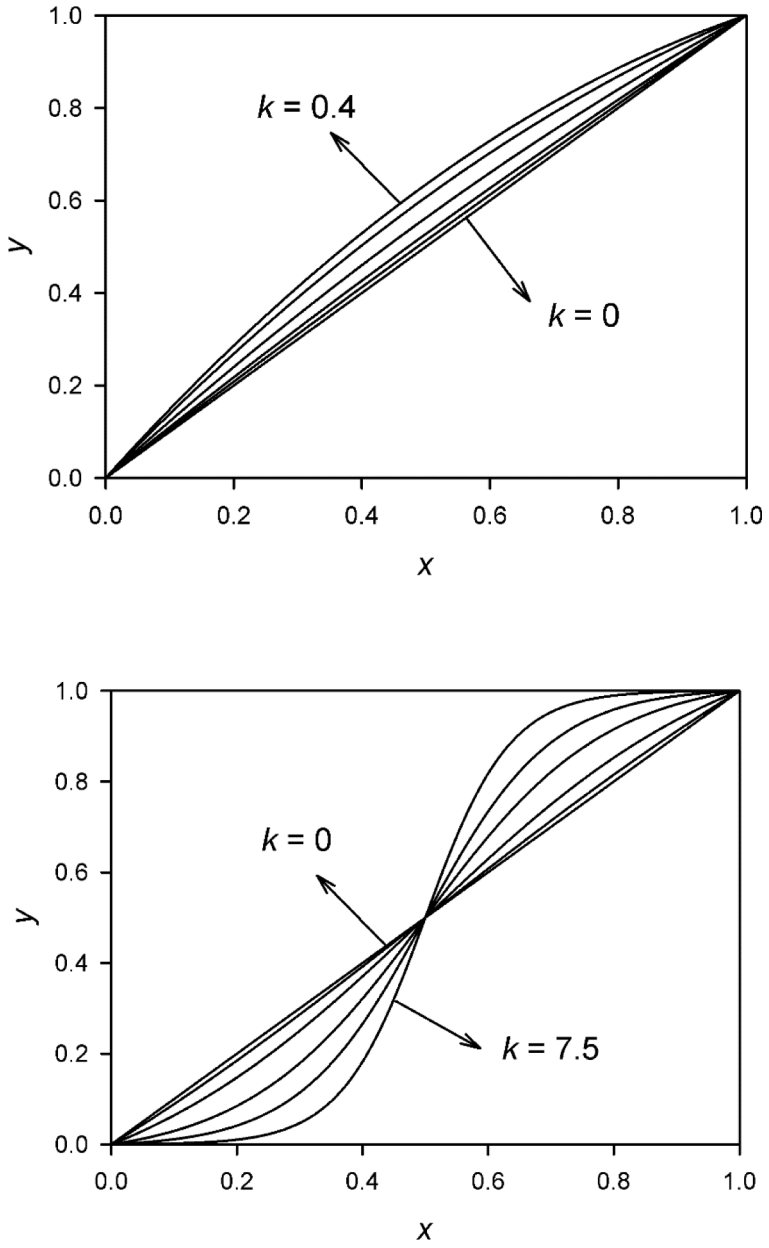


Fig. 3. Functions used in the simulations for different values of *k*. Top, quadratic system (*k* in the range 0–0.4), bottom, sigmoidal system (*k* in the range 0–7.5).

Table 1
Summary of the experimental data sets.

System (analyte) / Spectral technique	Number of calibration samples	Number of test samples	Wavelength range / nm	Step / nm	Spectral data points
Meat (fat and moisture) / NIR	170	69	850–1050	2	100
Yerba (aromatic herbs) / NIR	109	47	1100–2500	0.5	2800
Wheat (protein) / NIR	171	74	820–1100	0.5	562
Corn (oil) / NIR	50	30	1100–2500	2	700
Serum (tetracycline) / Fluorescence	50	57	400–600	1	61
Bioprocess (Protein) / Fluorescence	26	9	[a]	[a]	9625[a]

[a] These data consists of unfolded excitation-emission matrices, in the ranges 250–598 nm every 2 nm (emission) and 225–495 nm every 5 nm (excitation).

near infrared spectroscopy (NIRS), or studying samples of biological relevance from fluorescence spectra. A novel study about the sensitivity in kernel PLS is another important issue to be developed in the present article. Finally, the focus will be directed to a very interesting property derived from the calculated sensitivity coefficients in the context of kernel PLS: the estimation of the non-linearity degree (NLD) in the data under analysis. We notice that KPLS has also been applied to second-order (matrix) data, combining the KPLS model applied to the unfolded data with a residual bilinearization (RBL) procedure [24]. Inasmuch as the present tutorial is concerned with first-order (vector) data, the specifics of KPLS/RBL are beyond the scope of this report, and can be found in ref. [24].

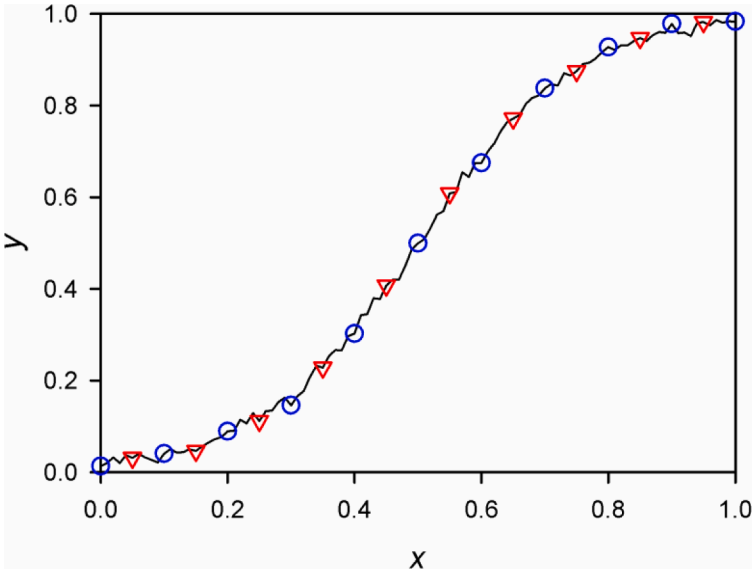


Fig. 4. Black line, non-linear relationship between *y* and *x* (including noise), blue circles, 11 points selected for model training, red triangles, 10 points selected for model validation.

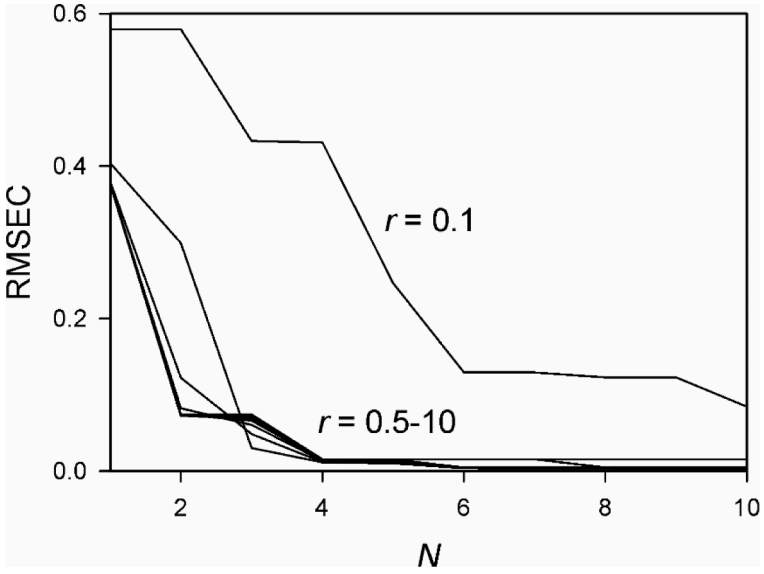


Fig. 5. RMSEC for the 10 validation points as a function of the number of Gaussian terms (*N*) for various radii (*r*).

2. Theory

2.1. Kernel PLS

In multivariate regression problems, each sample is characterized by a spectrum **x** with a number of elements, e.g., the number of wavelengths of a spectrum. The projection of the multivariate spectrum **x** on the space spanned by the *n*th. Gaussian function of a set *N* functions is performed using the following expression [13]:

$$k_n = \exp\left(- \| \mathbf{x} - \mathbf{c}_n \|^2 \Big/ r^2\right) \quad (1)$$

where $\mathbf{c}_n$ is a vector of centers and || || stands for the 2-norm of a vector. Once the radius *r* and a suitable number of $\mathbf{c}_n$ vectors (*N*) are selected, each sample will be represented by a **k** vector with *N* elements.

There are different manners in which *r, N* and the associated **c** vectors can be selected. In kernel PLS, *N* is equal to the number of calibration samples, and the **c** vectors are the calibration spectra. The optimum radius can be found by scanning a range of values within a certain range or using a method to minimize an error indicator parameter, e.g., the 'fminsearch' function in MATLAB.

Specifically, kernel PLS involves a calibration phase based on a kernel matrix $\mathbf{K}_{\text{cal}}$, whose elements are defined as follows:

$$K_{\text{cal},i1i2} = \exp\left(- \|\mathbf{x}_{\text{cal},i1} - \mathbf{x}_{\text{cal},i2}\|^2 \Big/ r^2\right) \quad (2)$$

where $\mathbf{x}_{\text{cal},i1}$ and $\mathbf{x}_{\text{cal},i2}$ are the spectra for two calibration samples with indexes *i*1 and *i*2. The $\mathbf{K}_{\text{cal}}$ matrix is of size $I_{\text{cal}} \times I_{\text{cal}}$, where I_{cal} is the number of calibration samples. Notice that some reports use the PCA scores instead of raw (or pre-processed) spectral signals; inasmuch as the chosen scores capture a significant portion of the spectral variance, the performance of KPLS models built with raw signals and with PCA scores should not be significantly different [22].

It has been shown that non-linear problems can be linearized by projection of the original data onto a feature space, in the present case represented by Gaussian functions [13]. Therefore, the next step in KPLS

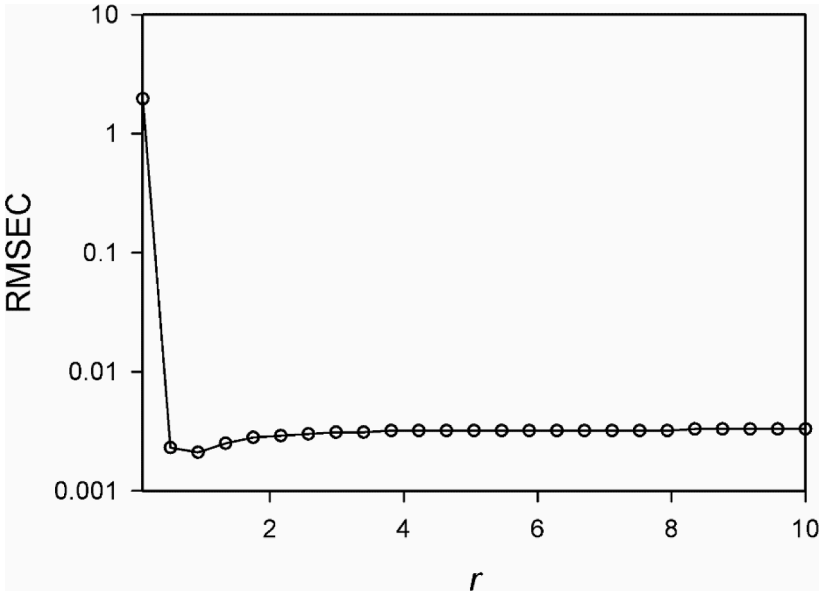


Fig. 6. RMSEC for the 10 validation points as a function of radius when $N = 4$.

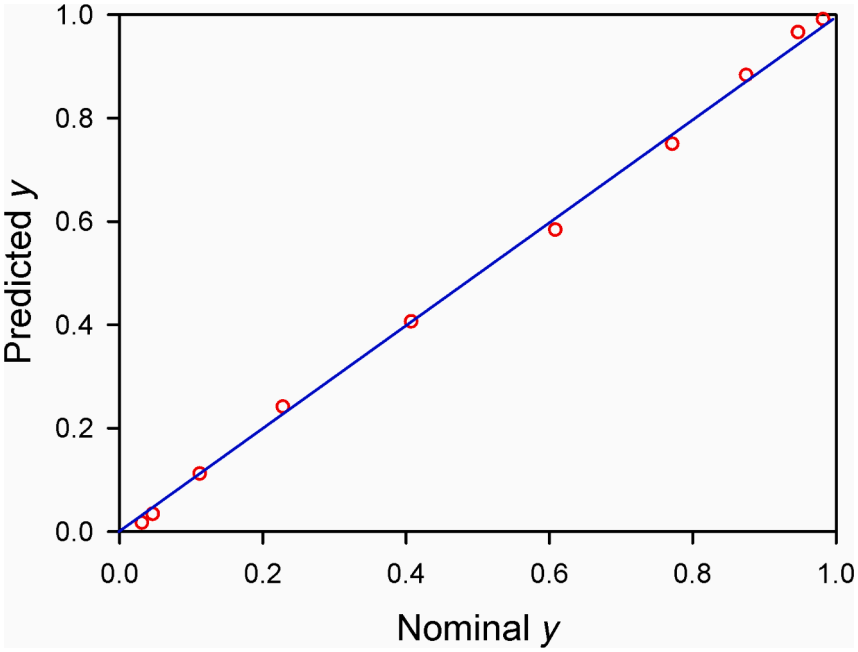


Fig. 7. Predicted vs. nominal y values for the 10 validation points for $N = 4$ and $r = 0.45$ (red circles). The blue line indicates the perfect fit.

is to build a classical PLS model between the matrix $\mathbf{K}_{cal}$ and the vector of calibration concentrations or target sample properties, using a certain number of latent variables, a procedure which provides a vector of regression coefficients $\boldsymbol{\beta}$ to be applied for prediction in future test samples. The optimum number of latent variables for estimating $\boldsymbol{\beta}$ can be set by leave-one-out cross-validation, while simultaneously optimizing the parameter r, usually by scanning a set of discrete values in the range 0.5–10 to yield its optimum value for model building. During the latter procedure, the spectra (either raw or pre-processed) are previously scaled in the range 0–1 before mean centering, to make the scale compatible with the range of r values. A better approach is to estimate both the radius and the number of latent variables by non-linear optimization techniques to avoid missing a good r value if a discrete trial set is search.

In the prediction phase, the spectrum for the unknown test sample ($\mathbf{x}_{test}$) is transformed in a kernel vector whose generic element is given by:

$$k_{test,i} = \exp\left(- \| \mathbf{x}_{cal,i} - \mathbf{x}_{test} \|^2 \Big/ r^2\right) \quad (3)$$

where the optimum value of r is considered.

The vector $\mathbf{k}_{test}$ for the unknown sample then renders the predicted analyte concentration y_{pred} from the standard PLS prediction expression:

$$y_{pred} = \mathbf{k}_{test}^{T} \beta \quad (4)$$

The algorithm for applying KPLS is very simple. It includes just one additional line in comparison with classical PLS, as indicated in Fig. 1 with the red and blue squares for the key MATLAB and R lines respectively. For specific algorithms written in both MATLAB and R for applying KPLS see the Appendix. The following are the details for

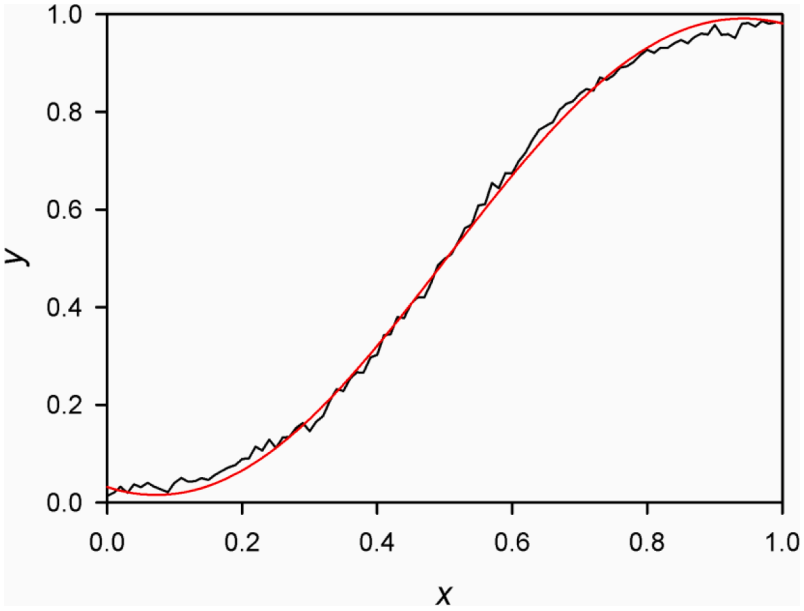


Fig. 8. Comparison of the non-linear relationship between y and x (black line) and that estimated using $N = 4$ and $r = 0.45$ (red line).

implementing the MATLAB code (practitioners of R will find the analogous R programming lines in the Appendix). Specifically, line 2 estimates the mean calibration spectrum, which is then subtracted from all calibration (line 3) and test (line 4) spectra. Lines 6 and 7 scale the data matrices so that the maximum intensity is 1. Lines 9–13 describe a double loop for estimating the elements of the calibration kernel matrix, with the corresponding activity for the test data matrix in lines 14–18. Both are mean-centered in lines 20–24, along with the calibration and test concentration vectors. The subsequent lines (26–37) contain the classical PLS iterative algorithm where the scores T, the loadings P, the weight loadings W and the latent regression vector v are estimated, as described in detail in ref. [25]. At each step of the loop, the X and Y blocks are deflated (lines 35 and 36). Line 38 estimates the vector of regression coefficients in real space, and line 40 predicts the analyte concentration as the scalar product of the test kernel matrix and the latter vector (after uncentering the result). The algorithm given in the Appendix can be easily adapted to a previous cross-validation phase to estimate the optimum radius and the number of latent variables, as provided in the Research Data accompanying the present work.

2.2. Simulations

A series of simulated data sets were prepared with two components showing partially overlapped Gaussian lines (Fig. 2) and different degrees of non-linearity in the signal-concentration relationship. One hundred calibration samples were built with random component concentrations in the range 0–1. Gaussian noise was added at 1% level with respect to the maximum calibration signal. The calibration data set was then submitted to an algorithm estimating the optimum value of r and the number of latent variables using the error in leave-one-out cross-validation as error indicator.

In the quadratic systems, the relation between the variables is governed by the following expression:

$$\mathbf{x} = \mathbf{s}\,\frac{y - ky^2}{1 - k} \tag{5}$$

where $\mathbf{x}$ is the spectrum of a given analyte in a sample, $\mathbf{s}$ is the pure-component spectrum, k is a parameter measuring the departure of linearity and y is the analyte concentration. On the other hand, in the sigmoidal systems the relationship is as follow:

$$\mathbf{x} = \mathbf{s}\frac{\tanh\left[k\left(y - \frac{1}{2}\right)\right] - \tanh\left(-\frac{k}{2}\right)}{\tanh\left(\frac{k}{2}\right) - \tanh\left(-\frac{k}{2}\right)} \tag{6}$$

where the symbols have the same meaning as in Eq. (5). Fig. 3 shows the various plots of signal at a given sensor as a function of concentration for all simulated systems. Notice that in the latter two equations, $k = 0$ leads to the linear problem $\boldsymbol{x} = \mathbf{s}\,y$.

2.3. Sensitivity

The material of this section has already been published as the Supplementary Material of ref. [22]. However, it is briefly presented here for completeness. The sensitivity (SEN) can be estimated by error propagation concepts, following the expression [22]:

$$\mathrm{SEN} = \frac{s_x}{s_y} \tag{7}$$

where s_x and s_y are the uncertainties in signal and concentration respectively.

On the other hand, the uncertainty in analyte concentration s_y can be computed from the KPLS expression for estimating the analyte concentration:

$$y = \beta^{\mathrm{T}}\mathbf{k} \tag{8}$$

From Eq. (8), and assuming a precise calibration phase, small changes dy and $d\mathbf{k}$ are related by:

$$dy = \beta^{\mathrm{T}} d\mathbf{k} = \beta^{\mathrm{T}}\,\mathbf{J}\,d\mathbf{x} \tag{9}$$

where $\mathbf{J}$ is the Jacobian matrix containing the first derivatives. The Jacobian can be calculated from the test sample kernel and the spectrum, leading to the following generic element $J(i,j)$:

$$J_{ij} = \frac{2}{r^2}\left[x_j - x_{\mathrm{cal},ji}\right]\exp\left(-\,\|\,\mathbf{x} - \mathbf{x}_{\mathrm{cal},i}\|^2 / r^2\right) \tag{10}$$

where j indicates the wavelength index, x_j is an element of the spectrum, $x_{\mathrm{cal},ji}$ an element of the matrix of calibration spectra, and $\mathbf{x}_{\mathrm{cal},i}$ the spectrum for the ith. calibration sample.

The next step is to estimate the variance in concentration as the ensemble average of $(dy)^2$:

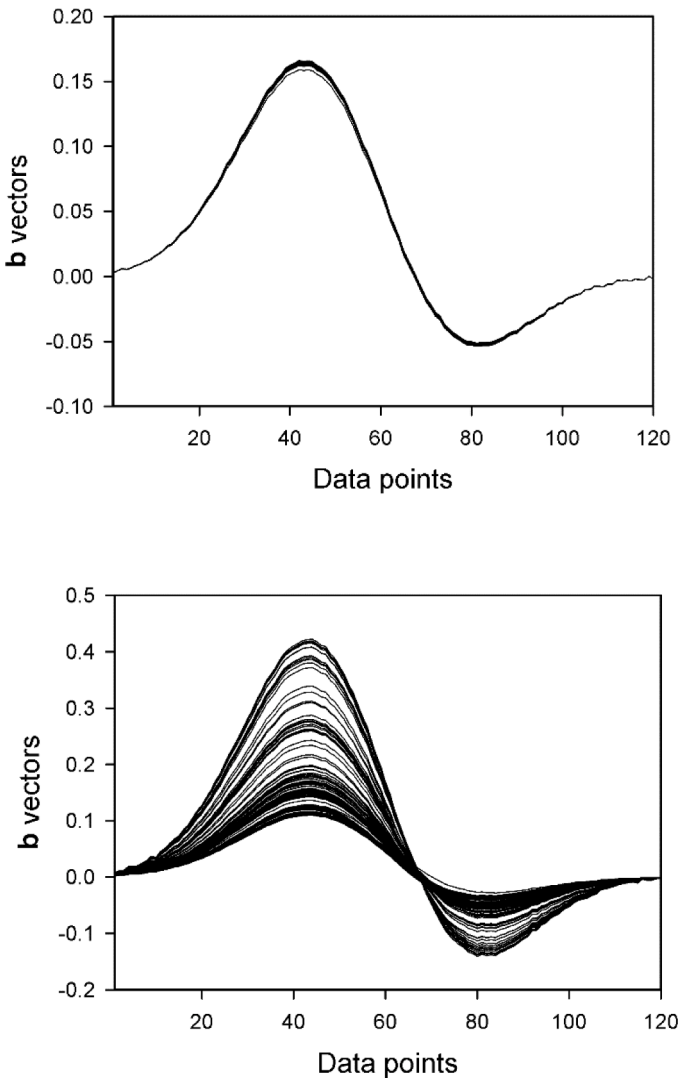


Fig. 9. Top, plot of the 100 calibration **b** vectors in the simulated quadratic system ($k = 0.01$). Bottom, the same plot for $k = 0.4$.

$$s_y^2 = \langle (dy)^2 \rangle = \beta^T \mathbf{J} \langle (d\mathbf{x}\, d\mathbf{x}^T) \rangle \mathbf{J}^T \beta \quad (11)$$

Since the ensemble average $<(d\mathbf{x}\ d\mathbf{x}^T)>$ is equal to the spectral variance $(s_x)^2$ multiplied by an identity matrix:

$$s_y^2 = s_x^2\, \beta^T\, \mathbf{J}\, \mathbf{J}^T \beta \quad (12)$$

From Eq. (12), the square sensitivity follows:

$$\left(\frac{s_y}{s_x}\right)^2 = \|\mathbf{J}^T \beta\|^2 \quad (13)$$

Recalling Eq. (7), the sensitivity is given by:

$$\mathrm{SEN} = 1 / \ \| \mathbf{J}^T \beta \| \quad (14)$$

In the latter expression, the elements of **J** are sample dependent, making the sensitivity to be sample dependent, as expected for a non-linear regression model.

When spectra are mathematically pre-processed before model building, **x** is pre-multiplied by a filtering matrix **F** containing the pre-processing coefficients. This leads to a modified spectrum $\mathbf{x}_p$:

$$\mathbf{x}_p = \mathbf{F}\, \mathbf{x} \quad (15)$$

and the uncertainty in $\mathbf{x}_p$ can be directly related to the uncertainty in the raw spectrum:

$$d\mathbf{x}_p = \mathbf{F}\, d\mathbf{x} \quad (16)$$

Therefore, the sensitivity is given by the following equation:

$$\mathrm{SEN} = 1 / \ \| \mathbf{F}\, \mathbf{J}^T \beta \| \quad (17)$$

2.4. Degree of non-linearity

A new aspect of KPLS is the possibility of classifying the data sets in terms of a certain degree of departure from linearity, and not in a black and white fashion as just 'linear' and 'non-linear', as is customary in the literature [2]. Eq. (17) can be interpreted by giving the sensitivity as the inverse of the 2-norm of a vector:

$$\mathbf{b} = \mathbf{F}\, \mathbf{J}^T \beta \quad (18)$$

In classical PLS, **b** would be the vector of model regression coefficients, but in KPLS the **b** vectors are sample dependent and provide a means of estimating the sensitivity. However, it is possible to employ them to define a non-linearity degree (NLD) parameter, as the ratio of

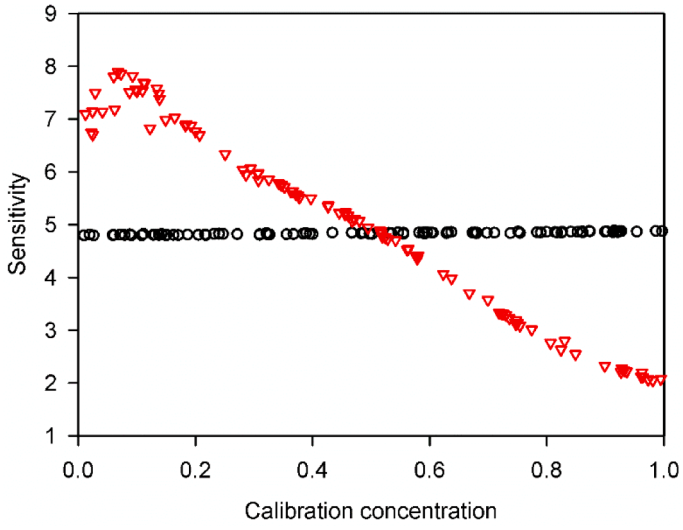


Fig. 10. Sensitivity for the 100 calibration samples of the simulated quadratic system as a function of nominal analyte concentration. Black circles, $k = 0.01$, red triangles, $k = 0.4$.

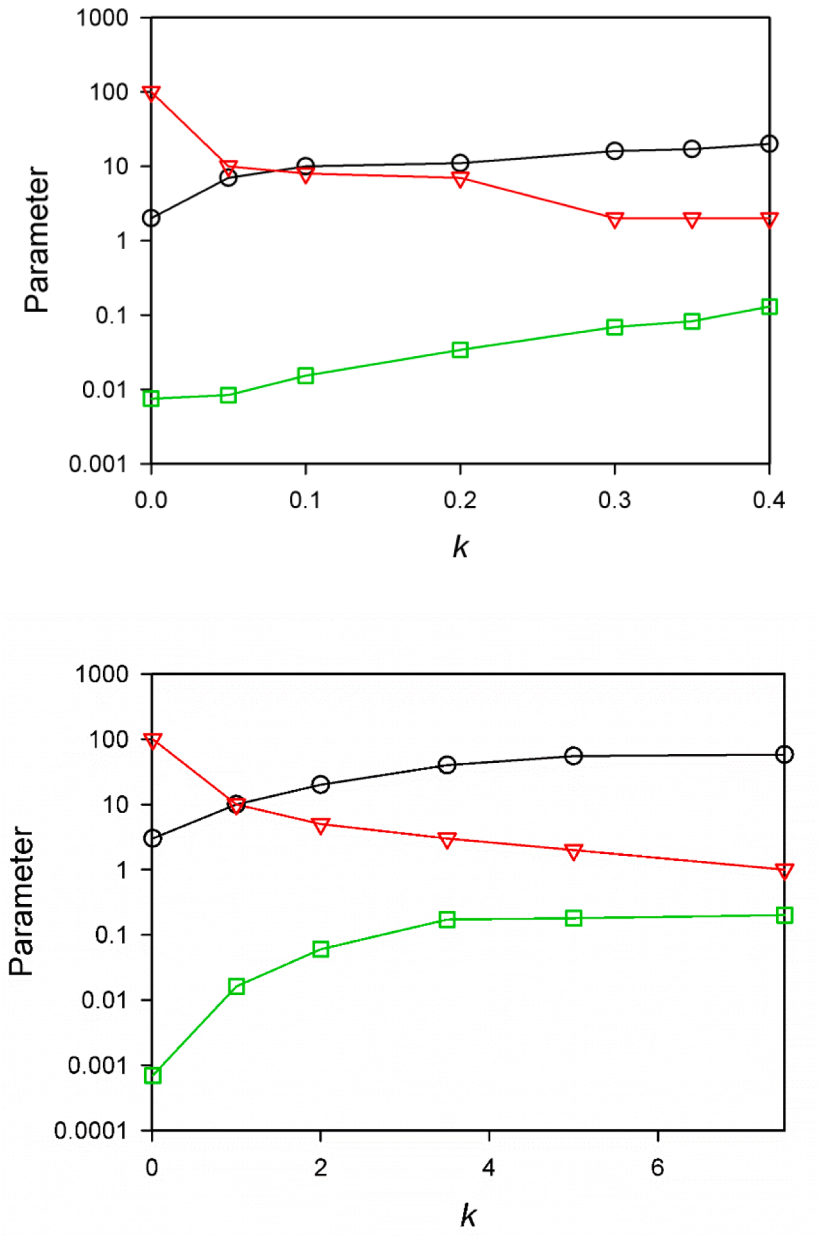


Fig. 11. Plot of the number of LVs (black line and circles), radius (red line and triangles) and NLD parameter (green line and squares) in the simulations as a function of the non-linearity constant (k). Top, quadratic system, bottom, sigmoidal system.

the amplitude of the set of **b** vectors to the mean, i.e.:

$$\mathrm{NLD} = \frac{\sum \Delta b_p}{P \parallel \overline{\mathbf{b}} \parallel} \tag{19}$$

where Δb_p is the difference between the maximum and minimum value of the **b** vector across the calibration samples at wavelength p, P is the total number of wavelengths and $\overline{\mathbf{b}}$ is the average **b** vector. The application and meaning of this parameter will be discussed below in connection with both the simulated and experimental data sets.

3.5. Software

MATLAB was used for simulations and for processing the data with MCR-ALS using an in-house code [26]. The Appendix provides the codes in both MATALB and R for model training and prediction on new samples. The R code was written in RStudio [27]. The set of MATLAB codes for cross-validation, calibration and prediction, along with the studied data sets, are given as Research Data at https://data.mendeley.com/datasets/vm2b5mfhbw .

4. Experimental

Four experimental data sets discussed below involve the calibration of food quality parameters from NIR spectra. They have already been described in the literature. Please see the corresponding references, which discuss the determination of: fat and moisture in meat samples [9, 28], aromatic herbs in yerba mate [22], protein in wheat [9,29], and oil in corn seeds [30,31].

The remaining two examples involve the determination of an antibiotic in human serum samples [32] and the monitoring of a protein in a bioprocess, in both cases from fluorescence spectral data [33]. Table 1 collects the relevant information regarding number of samples, spectroscopic technique, wavelength range and data points for each set.

Table 2
Analytical results and KPLS parameters for the experimental data sets.

System (analyte)	RMSEP PLS (LVs)[b]	RMSEP KPLS (LVs)[b]	r	NLD	Type
Meat (fat)[a]	2.6(8)	0.68 (52)	2	0.14	High
Meat (moisture)[a]	3.3(8)	0.54 (30)	3	0.13	High
Yerba (aromatic herbs)[a]	4.6(6)	2.9 (28)	6	0.025	Medium
Wheat (protein)[a,c]	0.22(13)	0.14 (41)	45	0.010	Medium-low
Corn (oil)[a,d]	0.052(8)	0.056 (16)	>1000	0.0011	Very low
Serum (tetracycline)[c]	0.10(4)	0.11(9)	330	0.0008	Very low
Bioprocess (Protein)	15(5)	14(9)	13	0.009	Medium-low

[a] First-derivative spectra were used for model building (SG filter, 5th. order polynomial, 9-points window).
[b] Latent variables (LVs) for model building in parenthesis. For PLS, they were found by leave-one-out cross-validation. For KPLS, the number of latent variables and the radius were found using MATLAB 'fminsearch' toolbox with leave-one-out RMSECV as the objective function to be minimized.
[c] After variable selection considering the shape of the **b** vectors.
[d] In this a fairly linear system the search for the best radius leads to very large values.

5. Discussion

5.1. A simple univariate example

In this section a simple example involving the modeling of a non-linear univariate data set will be discussed [34]. Fig. 4 (black line) shows a non-linear relationship between the y and x variables, where noise is apparently affecting the data. We note that the underlying equation relating y to x is not known. For developing a mathematical model to map the x-y relationship, the y values at eleven equally spaced points in the x axis are considered (blue circles in Fig. 4) and ten additional data points, independent from those used for training the model (red triangles in Fig. 4) are selected for validation. This procedure intends to mimic the characteristics of a multivariate regression problem, where spectra from a set of samples is used to build a model, and independent sample spectra are employed for validation.

One possible model consists of a linear combination of N Gaussian functions [33]:

$$y = \sum_{n=1}^{N} w_n \exp\left[-(x - c_n)^2 / r_n^2\right] \tag{20}$$

where c_n and r_n are the centers and radii of each Gaussian function and w_n the weights of the linear combination. The radii are proportional to the widths of the Gaussian functions, and are usually considered equal to a single r value for simplicity [35]. Thus, training the model demands the estimation of the number of terms (N), the centers (c_n), the common radius (r) and the weights (w_n). Given the values of N and r, and selecting N centers at random from the set of eleven x values considered for building the model, the weights can be estimated by least-squares regression, because the expression (1) represents an overdetermined system of linear equations. The procedure first involves the building of a so-called design matrix **D** of size ($11 \times N$) whose elements are given by:

$$D_{in} = \exp\left[-\left(x_{\text{cal},i} - c_n\right)^2 / r^2\right] \tag{21}$$

where $x_{\text{cal},i}$ represents each of the eleven x values chosen for calibration.

The vector of weights can then be estimated as follows, provided the matrix ($\mathbf{D}^{\mathrm{T}}\mathbf{D}$) can be safely inverted [33]:

$$\mathbf{w} = \left(\mathbf{D}^{\mathrm{T}}\mathbf{D}\right)^{-1} \mathbf{D}^{\mathrm{T}}\, \mathbf{y}_{\text{cal}} \tag{22}$$

where $\mathbf{y}_{\text{cal}}$ is the vector of y values at the calibration points. Eq. (22) can be applied if ($\mathbf{D}^{\mathrm{T}}\,\mathbf{D}$) can be safely inverted.

The estimation of N and r is usually done by trial and error. This activity includes the following steps: (1) r is scanned at a number of discrete values in a suitable range, in the present case 0.1–10, (2) in each case the weights are estimated according to Eq. (22) for increasing values of N using randomly chosen centers from the list of training x values, (3) the model is used to predict y for the eleven calibration points, and (4) the predictions are compared with the known values, calculating the root mean square error for calibration (RMSEC) as:

$$\text{RMSEC} = \sqrt{\frac{\left(y_{\text{pred}} - y_{\text{cal}}\right)^2}{I_{\text{cal}}}} \tag{23}$$

where y_{cal} and y_{pred} are the nominal and predicted calibration y values respectively, and $I_{\text{cal}} = 11$ is the number of calibration data points.

Fig. 5 shows the RMSEC as a function of N, where it is clear that $N = 4$ is a good choice, because the RMSEC stabilizes at this value, and using more than four Gaussian terms does not appear to be reasonable. The figure also shows that some values of r are not likely to produce good results (i.e., 0.1), and that the optimum r may be found in the range 0.5–10. It is also seen in Fig. 5 that after $N = 4$ the behavior of the RMSEC is similar to multivariate regression situations, when including too many latent variables in the predictive models.

Using this trial-and-error method, the best r can be searched by fixing N at 4, and then estimating RMSEC for a range of r values. A plot of RMSEC as a function of r (Fig. 6) shows that the optimum r is ca. 0.5. Since there is a clear minimum in Fig. 6, it may be risky to scan a discrete range of values to find the best radius. Instead, a method can be used to find the optimum r by minimizing the RMSEC, as implemented in the MATLAB 'fminsbnd' toolbox. This leads to $r = 0.45$ in the present case.

Having set $N = 4$ and $r = 0.45$ as the optimum calibration parameters, the next phase is the study of the independent validation set of 10 y values. Fig. 7 plots predicted vs. known y for the independent test set. The apparent good agreement is reflected in the root mean square error for prediction (RMSEP) which can be estimated using an equation similar to (23), replacing I_{cal} by $I_{\text{test}} = 10$, leading to RMSEP = 0.015 units, i.e., 3% with respect to the mean calibration y. This implies that the linear combination of Gaussian functions has been able to adequately train a non-linear model of a data set with an unknown underlying relationship between the two variables. Finally, it is interesting to see how well the 4-Gaussian linear combination maps the entire range of x from 0 to 1. This is shown in Fig. 8.

It is also worth mentioning that additional flexibility could be achieved by letting the r values to be different for each Gaussian function. However, when the four radii were simultaneously optimized, their estimated values were all very similar. It is therefore simpler to use the same radius for all terms.

The whole process can be described as first projecting the non-linear data onto the space spanned by Gaussian functions to linearize the data, followed by a linear model between the latter projections and the independent variable [33].

It is interesting to explore the situation if the above univariate system were linear. Repeating the modeling process as above, only two kernels are needed to model the y variable. The optimized r using the 'fminsearch' MATLAB toolbox is in this case rather large, i.e., on the order of 10^3 for any random choice of two centers from the eleven training x values. In addition, the least-squares estimation of the two weights renders $w_1 \approx -w_2$.

When r is large enough so that all ratios $(x - c)/r^2$ are small, the Gaussian functions can be approximated by a Taylor expansion. If only the linear terms of the latter are retained, the expression for y as a function of x can be written as:

$$y = w_1 \exp\left[-(x - c_1)^2 / r^2\right] + w_2 \exp\left[-(x - c_2)^2 / r^2\right] \approx$$

$$\approx w_1\left[1 - (x - c_1)^2 / r^2\right] + w_2\left[1 - (x - c_2)^2 / r^2\right] \tag{24}$$

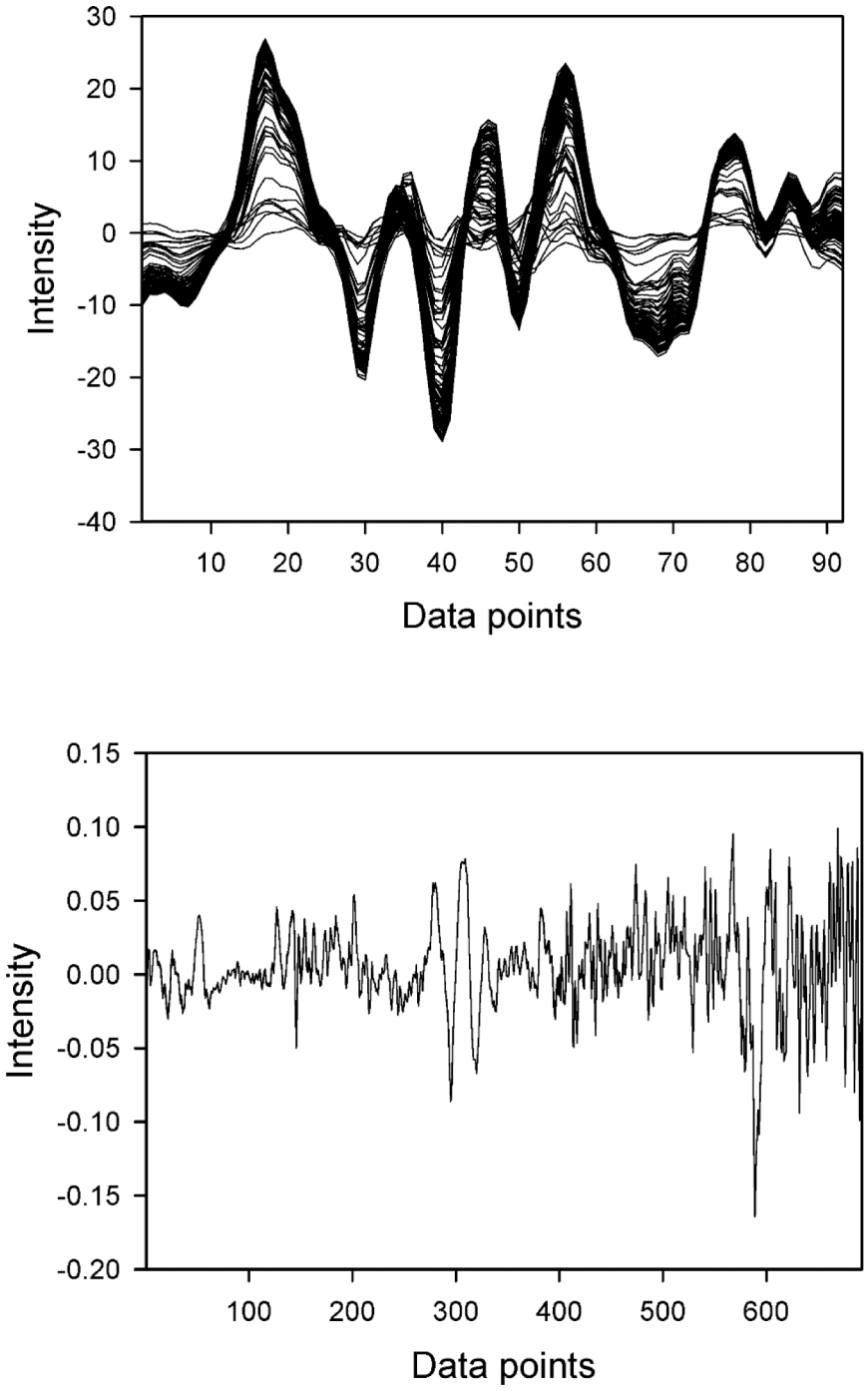


Fig. 12. Plot of all **b** vectors for the calibration samples in two extreme situations. Top, meat data set for fat (high NLD), bottom, corn data set for moisture (very low NLD).

Furthermore, if $w_1 = -w_2$, the latter equation yields:

$$y = \frac{w_2}{r^2}\left(c_1^2 - c_2^2\right) + \frac{2w_2}{r^2}(c_2 - c_1)x \tag{25}$$

Therefore, the projection onto the Gaussian space is nothing more than a linear transformation. As a consequence, there is no need of using a Gaussian expansion, and a linear model for mapping y as a function of x would be better. These results suggests that if a large value of r is found when optimizing a Gaussian projection model, a better choice would be a linear regression model.

5.1. Simulated multivariate data

From the simulations for the various two-component multivariate systems, Fig. 9 shows a typical set of **b** vectors for a nearly linear quadratic system ($k = 0.01$) and for a clearly non-linear system ($k = 0.4$). Notice that for both systems, the **b** vectors have a maximum at the sensor for which the analyte of interest is most responding (sensor No. 40) and a minimum at the sensor No. 80, the position of the maximum for the interferent (Fig. 9). In this respect, the **b** vectors indicate regions of interest for the analyst which are analogous to those provided the classical PLS regression coefficients. Moreover, the (almost identical) **b** vectors for the nearly linear case are very similar to the vector of PLS regression

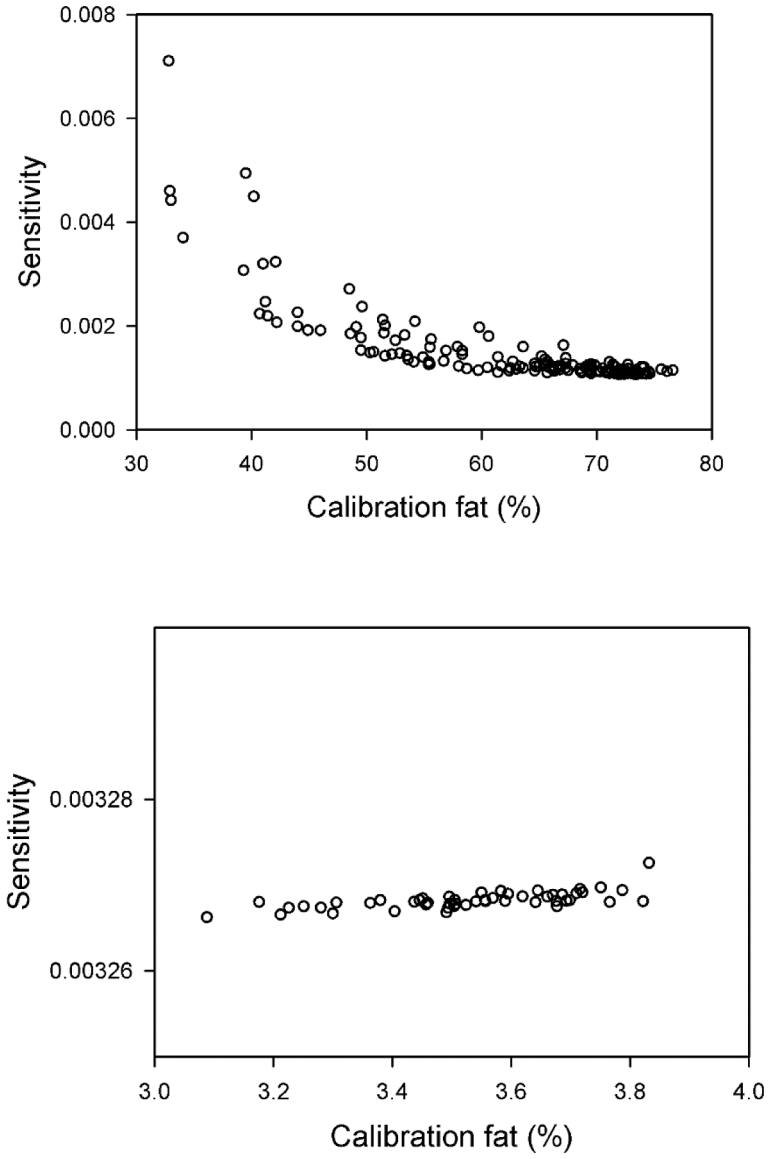


Fig. 13. Top, sensitivity as a function of calibration concentration for the training samples in the meat data set (analyte = fat). Bottom, same for the corn data set (analyte = oil).

coefficients which would be computed in the absence of Gaussian projections.

On the other hand, the sensitivities as a function of calibration concentrations for these two systems are shown in Fig. 10. For the nearly linear one the values are almost constant, as expected, whereas for the non-linear one they decrease with increasing concentration. This latter behavior is also the expected one from the function having a negative sign in the quadratic term, i.e., Eq. (5).

Increasing the degree of non-linearity of these data sets lead to the following conclusions: (1) the number of latent variables for building the PLS model after the Gaussian projection phase increases (in the linear case this number is close to the ideal one, i.e., the number of responsive constituents), (2) the estimated radius for the Gaussian functions continually decreases from a large value for the linear case and (3) the parameter NLD increases.

These observations are confirmed in Fig. 11 for both the quadratic and sigmoidal systems. It may be established the following limits for NLD for classifying the data sets according to the degree of non-linearity: for NLD > 0.05, high, for NLD in the range 0.01–0.05, medium, and for NLD < 0.01, low.

5.2. Experimental data

The experimental data sets discussed in this section have already been processed in the literature using various multivariate models, including variants of PLS and artificial neural networks. We only wish to discuss here two aspects: (1) the analytical improvement in going from the classical linear PLS model to the non-linear KPLS model, and (2) the classification of the various systems according to the degree of non-linearity. The former issue has already been discussed in some cases; however, we would like to call the attention to the fact that KPLS can be applied to the non-linear data sets with a success which is comparable to seemingly more complex methodologies. This fact is shown in the first two columns of Table 2. Notice that first-derivative was employed as mathematical pre-processing of the NIR spectra. Regarding the latter activity, a combination of KPLS with a previous convolutional phase driven by a stochastic search was presented for the automatic selection of the mathematical pre-processing method to be applied to NIR spectra [22]. Explicit details on the operation and application of convolutional KPLS (CKPLS), including the required software, can be found in ref. [22] and will not be reported here.

As regards the non-linearity degree NLD, Table 2 shows the

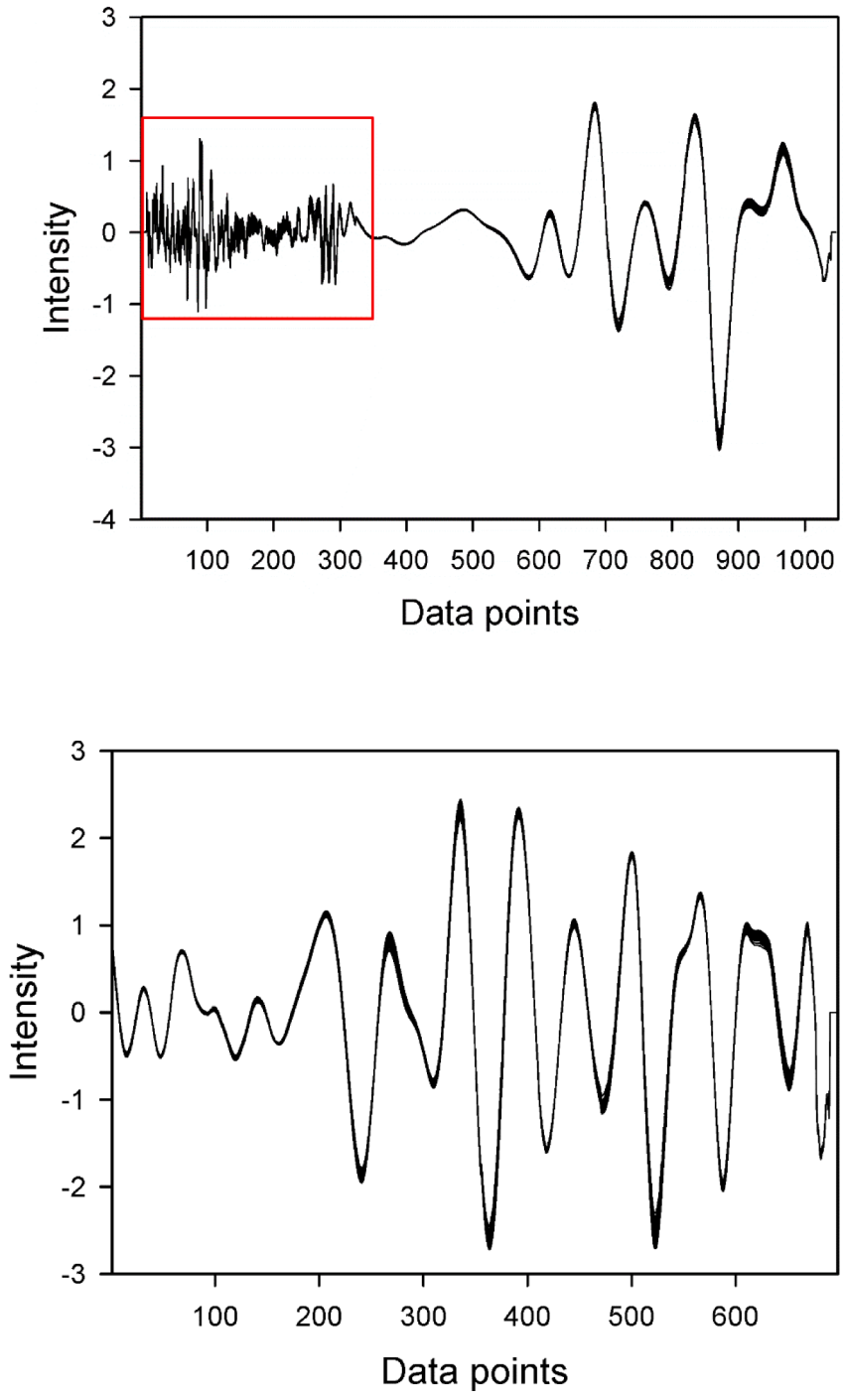


Fig. 14. Plots of all **b** vectors for the calibration samples in the determination of protein in wheat. Top, full spectra (the square shows the noisy region to be discarded for model building), bottom, discarding the first 350 data points. In both cases, first-derivative spectra were used for calibration.

optimized value of *r* and the NLD parameter for the various data sets. Two of them (the determination of fat and moisture in meat samples) are characterized as highly non-linear, in accordance with the literature. Three other systems have a medium-type non-linearity (detection of aromatic herbs in yerba, protein in wheat and protein in a bioprocess). The remaining two (detection of oil in corn seeds and tetracycline in serum) shows a very small value of NLD and is a typical example of a linear case. A visual comparison of the degree of non-linearity is presented in Fig. 12, where the **b** vectors are plotted for the two extreme cases (detection of fat in meat and of oil in corn seeds).

The significantly different distribution of **b** vectors in the latter two systems lead, as can be expected, to widely different sensitivity values as a function of analyte concentration in the calibration samples. In the case of the determination of fat in meat samples (Fig. 13, top) the sensitivity clearly decreases on increasing fat content, as is expected for a highly non-linear system. In contrast, for the detection of oil in corn seeds (Fig. 13, bottom) the sensitivity is fairly constant, corresponding to a nearly linear system.

As a summary of the application of KPLS to the experimental data sets, one may notice the following relevant issues: (1) the KPLS model is able to handle systems where the degree of non-linearity varies from very low to high with comparable success to the classical PLS model in

the linear systems and to advanced artificial neural network approaches in the non-linear ones, (2) the degree of non-linearity can be assessed on a continuous scale, avoiding the black-and-white classification in either linear or non-linear, (3) analytical figures of merit are available as in the classical PLS version, and (4) variable selection can be performed by calculating the vector of kernel regression coefficients in the space of the real spectral variables. All these advantages should be considered when selecting kernel PLS as a flexible multivariate calibration model for processing spectral data sets.

5.3. Variable selection

The selection of relevant variables for building a successful multivariate regression model is an important aspect in the analytical calibration scenario, and many different methods have been developed in this regard [36]. This section refers specifically to the variable selection procedure relying on the consideration of the properties of the vector of regression coefficients [37]. Although the kernel PLS vector **β** of Eq. (4) is not suitable in this regard, because it is not defined in the real wavelength (or sensor) space, the KPLS **b** vectors defined by Eq. (18) can be employed for variable selection.

When heavily overlapped spectra occur in certain regions, or when the signal is too high and saturates the detecting system, the **b** vector shows a highly noisy spectral region which should be discarded from the calibration range. This behavior is analogous to that of the vector of regression coefficients in the linear PLS model [36]. As an example, Fig. 14 shows the set of **b** vectors for all calibration samples in the KPLS model for the determination of protein content in wheat seeds using the full spectral data, and the analogous results when the noisy region is discarded. When processing these data in previous reports, the same spectral region was removed before model building, leading to significantly improved analytical results [9,28].

6. Conclusion

A tutorial has been presented on the theory and application of kernel partial least-squares regression, a flexible model applicable to multivariate calibration problems when the relationship between instrumental signals and target concentrations or properties is not linear. The advantages of the model can be summarized as follows: (1) the analytical figures of merit can be estimated by closed-form expressions, (2) sensitivity spectral vectors can be calculated which are useful for variable selection, (3) a parameter measuring the non-linearity degree (NLD) can be computed and used to size the departure of linearity in a continuous fashion, and (4) the analytical systems can be classified according to the non-linearity degree in high, medium, low and very low; in the latter case the classical partial least-squares regression model is recommended.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data, MATLAB codes and R codes for "Kernel partial least-squares (KPLS) for multivariate calibration. A tutorial." (Original data) (Mendeley Data)

Acknowledgments

Financial support from the following institutions is acknowledged: Universidad Nacional de Rosario (UNR, Project 80020190100037UR), CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas) and ANPCyT (Agencia Nacional de Promoción Científica y Tecnológica, Project PICT 2020-00179).

Appendix

MATLAB algorithm for KPLS training and prediction

The Xcal and Xtest matrices contain the calibration and test spectra respectively, ycal is the vector of calibration concentrations, Ical and Itest are the numbers of calibration and test samples, r is a previously optimized Gaussian radius, A is the optimized number of KPLS latent variables. Details on how to optimize r and A are given in the MATLAB codes provided as Research Data accompanying this report.

Line number	Code
1	% Mean centering and scaling
2	mXcal=mean(Xcal,2);
3	Xcal=Xcal-mXcal*ones(1,Ical);
4	Xtest=Xtest-mXcal*ones(1,Itest);
5	mmXcal0=max(abs(Xcal(:)));
6	Xcal=Xcal/mmXcal0;
7	Xtest=Xtest/mmXcal0;
8	% Estimation of kernel matrices
9	for i_s = 1:Ical
10	for j = 1:Ical
11	Kcal(i_s,j)=-exp(-(norm(Xcal(:,i_s)-Xcal(:,j)))^2/r^2);
12	end
13	end
14	for i_s = 1:Itest
15	for j = 1:Ical
16	Ktest(i_s,j)=-exp(-(norm(Xtest(:,i_s)-Xcal(:,j)))^2/r^2);
17	end
18	end
19	% Mean centering
20	mKcal=mean(Kcal,2);

(continued on next page)

(*continued*)

Line number	Code
21	Kcal=Kcal-mKcal*ones(1,Ical);
22	mycal=mean(ycal);
23	ycal=ycal-mycal;
24	Ktest=Ktest'-mKcal*ones(1,Itest);
25	% Kernel PLS
26	Xpls=Kcal';
27	Ypls=ycal;
28	*T*=[];*P*=[];*W*=[];Wn=[];*v*=[];
29	for *i* = 1:A
30	*W*=Xpls'*Ypls/(Ypls'*Ypls);
31	Wn(:,i)=*W*/sqrt(W'*W);
32	T(:,i)=Xpls*Wn(:,i);
33	v(:,i)=*T*(:,i)'*Ypls/(T(:,i)'*T(:,i));
34	P(:,i)=Xpls'*T(:,i)/(T(:,i)'*T(:,i));
35	Xpls=Xpls-T(:,i)*P(:,i)';
36	Ypls=Ypls-v(:,i)*T(:,i);
37	end
38	b1=Wn*inv(P'*Wn)*v';
39	% Estimates test concentrations
40	ypred=Ktest'*b1+mycal;

R algorithm for KPLS training and prediction

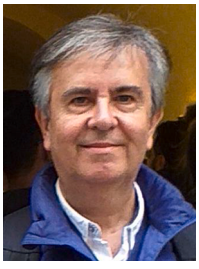
The symbols are the same as for the above MATLAB algorithm.

Line number	Code
1	library(pracma)
2	# Centering and scaling
3	meanX<-rowMeans(Xcal, dims = 1)
4	Xm<-Xcal-meanX%*%matrix(data = 1, nrow = 1, ncol = Ical)
5	Xtm<-Xtest-meanX%*%matrix(data = 1, nrow = 1, ncol = Itest)
6	meanY=mean(ycal)
7	Ym<-ycal-meanY
8	mmaX <- max(abs(Xm))
9	Xm <- Xm/mmaX
10	Xtm <- Xtm/mmaX
11	% Kernel matrices
12	Wn <- matrix(data = NA, nrow = Ical, ncol = A*)*
13	Ts <- matrix(data = NA, nrow = Ical, ncol = A*)*
14	*P* <- matrix(data = NA, nrow = Ical, ncol = A*)*
15	*v* <- matrix(data = NA, nrow = 1, ncol = A*)*
16	Kcal <- matrix(data = NA, nrow = Ical, ncol = Ical)
17	Ktest <- matrix(data = NA, nrow = Ical, ncol = Itest)
18	for (i in 1:Ical) {
19	#dXm <- Xm[,*i*]%*%ones(1,Ical)-Xm
20	Kcal[i,] <- exp(-(colSums((Xm[,*i*]%*%ones(1,Ical)-Xm)^2)/r^2))
21	}
22	for (i in 1:Ical) {
23	#dXtm <- Xm[,*i*]%*%ones(1,Itest)-Xtm
24	Ktest[i,] <- exp(-(colSums((Xm[,*i*]%*%ones(1,Itest)-Xtm)^2)/r^2))
25	}
26	# Centering
27	meanKcal<-rowMeans(Kcal, dims = 1)
28	Kcal<-Kcal-meanKcal%*%matrix(data = 1, nrow = 1, ncol = Ical)
29	Ktest<-Ktest-meanKcal%*%matrix(data = 1, nrow = 1, ncol = Itest)
30	# KPLS
31	Xpls=Kcal
32	Ypls<-Ym
33	for (i in 1:A) {
34	*W* <- Xpls%*%Ypls%*%solve(t(Ypls)%*%Ypls)
35	Wn[,i] <- W/as.numeric(sqrt(t(W)%*%W))
36	Ts[,i] <- t(Xpls)%*%Wn[,i]
37	v[i] <- t(Ts[,*i*])%*%Ypls%*%solve(t(Ts[,*i*])%*%Ts[,*i*])
38	P[,i] <- Xpls%*%Ts[,*i*]/as.numeric((t(Ts[,*i*])%*%Ts[,*i*]))
39	Xpls <- Xpls - P[,*i*]%*%t(Ts[,*i*])
40	Ypls <- Ypls-v[*i*]*Ts[,i]
41	}
42	b1 <- Wn%*%solve(t(P)%*%Wn)%*%t(v)
43	# Estimates test concentrations
44	ypred <- t(b1)%*%Ktest+meanY

References

[1] S.V.C. de Souza, R.G. Junqueira, A procedure to assess linearity by ordinary least squares method, Anal. Chim. Acta 552 (2005) 25–35.
[2] V. Centner, O.E. de Noord, D.L. Massart, Detection of nonlinearity in multivariate calibration, Anal. Chim. Acta 376 (1998) 153–168.
[3] S. Wold, M. Sjöström, L. Eriksson, PLS-regression: a basic tool of chemometrics, Chemom. Intell. Lab. Syst. 58 (2001) 109–130.
[4] H. Martens, T. Næs, Multivariate Calibration, John Wiley and Sons, Chichester, UK, 1989.
[5] P.D. Wentzell, L. Vega Montoto, Comparison of principal components regression and partial least squares regression through generic simulations of complex mixtures, Chemom. Intell. Lab. Syst. 65 (2003) 257–279.
[6] A. Höskuldsson, Quadratic PLS regression, J. Chemom. 6 (1992) 307–334.
[7] F. Allegrini, J.A. Fernandez Pierna, W.D. Fragoso, A.C. Olivieri, V. Baeten, P. Dardenne, Regression models based on new local strategies for near infrared spectroscopic data, Anal. Chim. Acta 993 (2016) 50–58.
[8] Z.B. Alfassi, Z. Boger, Y. Ronen, Statistical Treatment of Analytical Data. Artificial Neural Networks – Unlikely but Effective Tools in Analytical Chemistry, Blackwell Science, Oxford, 2004, pp. 172–262.
[9] F. Allegrini, A.C. Olivieri, Linear or non-linear multivariate calibration models? That is the question, Anal. Chim. Acta 1226 (2022), 340248.
[10] F. Despagne, D.L. Massart, Neural Networks in multivariate calibration, Analyst 123 (1998) 157R–178R.
[11] J. Mark, L. Orr, Introduction to Radial Basis Function Networks: Recent Advaces in Radial Basis Function Networks; Centre for Cognitive Science, University of Edinburgh, Scotland, 1996, pp. 1–67.
[12] B. Walczak, D.L. Massart, The Radial Basis Functions – Partial Least Squares approach as a flexible non-linear regression technique, Anal. Chim. Acta 331 (1996) 177–185.
[13] R. Rosipal, L.J. Trejo, Kernel partial least squares regression in reproducing kernel Hilbert space, J. Mach. Learn. Res. 2 (2001) 97–123.
[14] K. Kim, J.M. Lee, I. Lee, A novel multivariate regression approach based on kernel partial least squares with ortogonal signal correction, Chemom. Intell. Lab. Syst. 79 (2005) 22–30.
[15] P. Shan, Y. Bi, Z. Li, Q. Wang, Z. He, Y. Zhao, S. Peng, Unsupervised model adaptation for multivariate calibration by domain adaptation-regularization based kernel partial least square, Spectrochim. Acta A 292 (2023), 122418.
[16] D.D. Silalahi, H. Midi, J. Arasan, M.S. Mustafa, J.-.P. Caliman, Kernel partial least square regression with high resistance to multiple outliers and bad leverage points on near-infrared spectral data analysis, Symmetry (Basel) 13 (2021) 547.
[17] Y. Bao, J. Liu, Y. Zhong, Y. Chen, D. Zhai, Q. Wang, C.S. Brennan, H. Liu, Kernel partial least squares model for pectin content in peach using near-infrared spectroscopy, Int. J. Food Sci. Technol. 56 (2021) 1877–1885.
[18] H. Chen, B. Lin, K. Cai, A. Chen, S. Hong, Quantitative analysis of organic acids in pomelo fruit using FT-NIR spectroscopy coupled with network kernel PLS regression, Infrared Phys. Technol. 112 (2021), 103582.
[19] J. Liu, X. Luan, F. Liu, Adaptive JIT-Lasso modeling for online application of near infrared spectroscopy, Chemom. Intell. Lab. Syst. 183 (2018) 90–95.
[20] V.E. de Almeida, A. de Araújo Gomes, D.D. de Sousa Fernandes, H.C. Goicoechea, R. Kawakami Harrop Galvão, M.C. Ugulino Araújo, Vis-NIR spectrometric determination of Brix and sucrose in sugar production samples using kernel partial least squares with interval selection based on the successive projections algorithm, Talanta 181 (2018) 38–43.
[21] X. Huang, L. Xia, Improved kernel PLS combined with wavelength variable importance for near infrared spectral analysis, Chemom. Intell. Lab. Syst. 168 (2017) 107–113.
[22] G.B. Rossi, V.A. Lozano, A.C. Olivieri, Spectral pre-processing and non-linear calibration with convolutional kernel partial least-squares. Teaching new tricks to an old dog, Chemom. Intell. Lab. Syst. 233 (2023), 104736.
[23] P. Shan, Y. Zhao, Q. Wang, S. Wang, Y. Ying, S. Peng, A nonlinear calibration transfer method based on joint kernel subspace, Chemom. Intell. Lab. Syst. 210 (2021), 104247.
[24] A. García-Reiriz, P.C. Damiani, A.C. Olivieri, Residual bilinearization combined with kernel-unfolded partial least-squares: a new technique for processing non-linear second-order data achieving the second-order advantage, Chemom. Intell. Lab. Syst. 100 (2010) 127–135.
[25] D.M. Haaland, E.V. Thomas, Partial least-squares methods for spectral analyses. 1. Relation to other quantitative calibration methods and the extraction of qualitative information, Anal. Chem. 60 (1988) 1193–1202.
[26] MATLAB version R2012a, The Mathworks, Natick, Massachusetts, USA, 2012.
[27] RStudio Team, RStudio: Integrated Development For R. RStudio, PBC, Boston, MA, 2020. URL, http://www.rstudio.com/.
[28] T. Naes, T. Isaksson, Locally weighted regression in diffuse near-infrared transmittance spectroscopy, Appl. Spectrosc. 46 (1992) 34–43.
[29] P.B. Harrington, Feature expansion by a continuous restricted Boltzmann machine for near-infrared spectrometric calibration, Anal. Chim. Acta 1010 (2018) 20–28.
[30] NIR of corn samples http://www.eigenvector.com/data/Corn (accessed Feb 12, 2023).
[31] T.M. Antonelli, A.C. Olivieri, Developing and implementing an R Shiny application to introduce multivariate calibration to advanced undergraduate students, J. Chem. Educ. 97 (2020) 1176–1180.
[32] H.C. Goicoechea, A.C. Olivieri, Enhanced synchronous spectrofluorometric determination of tetracycline in blood serum by chemometric analysis. Comparison of partial least-squares and hybrid linear analysis calibrations, Anal. Chem. 19 (1999) 4361–4368.
[33] F. Chiappini, C.M. Teglia, G. Forno, H.C. Goicoechea, Modelling of bioprocess nonlinear fluorescence data for at-line prediction of etanercept based on artificial neural networks optimized by response surface methodology, Talanta 210 (2020), 120664.
[34] A.C. Olivieri, Introduction to Multivariate calibration. A practical Approach, Springer-Nature, Berlin, 2018. Chapter 12.
[35] M.J.L. Orr, Regularization in the selection of radial basis function centers, Neur. Comp. 7 (1995) 606–623.
[36] T. Mehmood, K.H. Liland, L. Snipen, S. Sæbø, A review of variable selection methods in Partial Least Squares Regression, Chemom. Intell. Lab. Syst. 118 (2012) 62–69.
[37] Ref. [30], Chapter 8.

Franco Allegrini was born in Venado Tuerto, Santa Fe, Argentina (10/30/1986). He obtained his B. Sc. (2011) and his Ph. D (2015) from the National University of Rosario. He is currently doing a Postdoc at the Rosario Institute of Chemistry (IQUIR) with a fellowship from the National Research Council (CONICET), under the supervision of Prof. Olivieri. He has published 18 papers in international journals and a book chapter, including colaborations with different research groups around the world. His-main research topic is chemometrics oriented to multivariate calibration in Analytical Chemistry. He has recently joined the staff of a laboratory for quality control of foodstuff at a private institution.



Alejandro C. Olivieri is a member of the Institute of Chemistry of Rosario, and professor of the Department of Analytical Chemistry of the National University of Rosario. He was born in Rosario, Argentina, on July 28, 1958. Degree in Industrial Chemistry (Catholic Faculty of Chemistry and Engineering, 1982), Doctor (Faculty of Biochemical and Pharmaceutical Sciences, University of Rosario, 1986), fellow of the National Council for Scientific and Technical Research (CONICET). About 250 publications, books and book chapters. John Simon Guggenheim Memorial Foundation Fellow (2001–2002). Konex Platinum Award (Konex Foundation, Argentina, 2013) for his contributions to analytical chemistry. Current interest: chemometrics in analytical chemistry.