

Computational performance and cross-validation error precision of five PLS algorithms using designed and real data sets[†]

João Paulo A. Martins^a, Reinaldo F. Teófilo^{a,b} and Márcia M. C. Ferreira^{a*}

An evaluation of computational performance and precision regarding the cross-validation error of five partial least squares (PLS) algorithms (NIPALS, modified NIPALS, Kernel, SIMPLS and bidiagonal PLS), available and widely used in the literature, is presented. When dealing with large data sets, computational time is an important issue, mainly in cross-validation and variable selection. In the present paper, the PLS algorithms are compared in terms of the run time and the relative error in the precision obtained when performing leave-one-out cross-validation using simulated and real data sets. The simulated data sets were investigated through factorial and Latin square experimental designs. The evaluations were based on the number of rows, the number of columns and the number of latent variables. With respect to their performance, the results for both simulated and real data sets have shown that the differences in run time are statistically different. PLS bidiagonal is the fastest algorithm, followed by Kernel and SIMPLS. Regarding cross-validation error, all algorithms showed similar results. However, in some situations as, for example, when many latent variables were in question, discrepancies were observed, especially with respect to SIMPLS. Copyright © 2010 John Wiley & Sons, Ltd.

Keywords: computational performance; partial least squares; experimental design; algorithms

1. INTRODUCTION

Multivariate calibration is used to develop a quantitative relationship between several predictor variables and a property of interest (the response or dependent variable). The regression problem, i.e. how to model one or several dependent variables, \mathbf{Y} , by means of a set of predictor variables, \mathbf{X} , is one of the most common data-analytical problems in science and technology. The dependent variables in chemistry are usually concentrations, biological activities, sensory data responses, among others, while the predictor variables are represented respectively by measured spectra, physicochemical descriptors and chromatograms. The solution of this problem is obtained by solving the equation $\mathbf{Y} = \mathbf{X}\mathbf{B}$, where \mathbf{B} is the regression matrix or vector, given by $\mathbf{B} = \mathbf{X}^+\mathbf{Y}$ where \mathbf{X}^+ is the Moore–Penrose generalized inverse [1,2].

Traditional modeling of \mathbf{Y} by means of \mathbf{X} is based on the use of MLR (Multiple Linear Regression), which works well as long as there are only a few \mathbf{X} -variables compared to the number of samples and they are poorly correlated to each other, i.e. \mathbf{X} is full rank. Data matrices can be very large in distinct applications of multivariate calibration, e.g. in QSAR/QSPR (Quantitative Structure Activity/Property Relationship) studies [3], data mining [4], near infrared spectroscopy (NIR) [5], nuclear magnetic resonance (NMR) [6], chromatography [7] and studies dealing with unfolded matrices from multiway data [8], among others [9]. In such cases, the response variables are by their nature highly correlated to each other, leading to ill-conditioned \mathbf{X} matrices. Thus, MLR cannot be used in such cases, unless a careful variable selection is carried out. To avoid the problem of ill-conditioned matrices,

projection methods such as PCR (Principal Component Regression) or PLS (Partial Least Squares) are good alternatives [10]. The central idea of both methods is to approximate \mathbf{X} by a few components and regress the dependent variables against these components. The two methods differ essentially in the way the components are obtained.

Among the multivariate calibration methods, PLS is the most popular in chemistry. This is a multivariate modeling method derived around 1975 from Herman Wold's basic concepts in the field of econometrics. It consists of calculating principal components as well as canonical correlations by means of an iterative sequence of simple ordinary least squares (OLS)

* Correspondence to: M. M. C. Ferreira, Theoretical and Applied Chemometrics Laboratory, Institute of Chemistry, University of Campinas, Campinas, SP, 13083-970, Brazil
E-mail: marcia@iqm.unicamp.br

a J. P. A. Martins, R. F. Teófilo, M. M. C. Ferreira
Theoretical and Applied Chemometrics Laboratory, Institute of Chemistry, University of Campinas, Campinas, SP, 13083-970, Brazil

b R. F. Teófilo
Instrumentation and Chemometric Laboratory, Department of Chemistry, Federal University of Viçosa, Viçosa, MG, 36571-000, Brazil

[†] Parts of this work were presented at two Conferences: Teófilo R F, Martins J P A, Ferreira M M C. Study of computational performance of PLS algorithms using experimental design. 10th Scandinavian Symposium on Chemometrics (SSC10), Lappeenranta, Finland, 2007, Book of Abstracts; 35. Teófilo R F, Martins J P A, Ferreira M M C. Computational performance of PLS algorithms: a comparison. PLS07—5th International Symposium on PLS and Related Methods, Aas, Norway, 2007, Book of Abstracts; 128–131.

regression. The chemometric version of PLS regression was originally developed by Svante Wold in 1983 as a two-block algorithm, consisting of a sequence of simple, partial models fitted by least-squares [10].

The components in PLS regression are defined in a way to keep the compromise between fitting \mathbf{X} and predicting \mathbf{Y} . In the simplest case of a single property, the \mathbf{Y} matrix is reduced to a vector \mathbf{y} and the method is designated as PLS1. In this case, each component which relates \mathbf{X} and \mathbf{y} is obtained by taking into account the information contained in \mathbf{y} by maximizing the covariance between the \mathbf{X} scores (\mathbf{t}) and \mathbf{y} , such that $\mathbf{X}\mathbf{w} = \mathbf{t}$ and $\mathbf{w} = \frac{\mathbf{X}^t\mathbf{y}}{\|\mathbf{X}^t\mathbf{y}\|}$ [10–13]. Due to its ability in handling strongly collinear (correlated), noisy and numerous X-variables, the PLS method allows investigation of more complex problems than ever before [14]. No *a priori* assumptions are made about the model's structure, but estimates of reliability may be made using the 'jack-knife' or cross-validation methods. PLS modeling has become an important tool in many diverse scientific fields, e.g. psychology [15], economics [16], chemistry [17], food science [18], medicine and the pharmaceutical sciences [19,20], among others.

For the large data sets used nowadays, computational time is a factor that cannot be neglected [21], especially in cross-validation and variable selection steps, where the PLS algorithm is run several times [22]. Therefore, a fast PLS algorithm is required for such cases, since the computational time can be radically reduced during model building. Several variants of PLS algorithms were developed in recent years in an attempt to resolve this problem. Among the most used algorithms are NIPALS [11,23], modified NIPALS [24], Kernel [24,25], SIMPLS [26] and bidiagonal PLS [21,27].

The purpose of this work is to compare these five PLS algorithms available in the literature with respect to their computational time and the precision observed in the leave-one-out cross-validation error. Matrices of different sizes were tested aiming to find out which algorithm would be the most appropriate for each situation. In these tests only PLS1 (one dependent variable) was considered.

2. NOTATION

Scalars are defined as italic lower case characters (a, b, c), vectors are in bold lower case characters ($\mathbf{a}, \mathbf{b}, \mathbf{c}$) and matrices as bold upper case characters ($\mathbf{A}, \mathbf{B}, \mathbf{C}$). Matrix elements are represented by corresponding italic lower case characters with row and column index subscripts (x_{ij} is an element of \mathbf{X}). In some cases, matrices will be written explicitly as \mathbf{X} ($I \times J$) to emphasize their dimensions (I rows and J columns). The identity matrices are represented as \mathbf{I} with their proper dimensions indicated.

Superscripts t and -1 represent transpose and inverse operations, respectively.

3. ALGORITHMS

Five algorithms were tested in order to evaluate their computational time and precision in the leave-one-out cross-validation error. It is assumed that the matrices are adequately pretreated. These algorithms are described in the following text.

3.1. The classical NIPALS algorithm

The first algorithm used in PLS regression was NIPALS (nonlinear iterative partial least squares), presented in detail elsewhere [11,23]. It can be summarized as follows:

- (1) Call the \mathbf{X} matrix and \mathbf{y} vector \mathbf{X}_0 and \mathbf{y}_0 , respectively;
- (2) Compute the quantities \mathbf{w} (PLS weights for \mathbf{X}), \mathbf{t} (PLS scores for \mathbf{X}), \mathbf{p} (PLS loadings for \mathbf{y}) and \mathbf{q} (PLS loadings for \mathbf{X}):

$$\begin{aligned} \mathbf{w}_{a+1} &= \mathbf{X}_a^t \mathbf{y}_a \\ \mathbf{w}_{a+1} &= \frac{\mathbf{w}_{a+1}}{\|\mathbf{w}_{a+1}\|} \\ \mathbf{t}_{a+1} &= \mathbf{X}_a \mathbf{w}_{a+1} \\ \mathbf{p}_{a+1} &= \frac{\mathbf{X}_a^t \mathbf{t}_{a+1}}{\mathbf{t}_{a+1}^t \mathbf{t}_{a+1}} \\ q_{a+1} &= \frac{\mathbf{y}_a^t \mathbf{t}_{a+1}}{\mathbf{t}_{a+1}^t \mathbf{t}_{a+1}} \end{aligned}$$

- (3) Deflate \mathbf{X} and \mathbf{y} by subtracting the computed latent vectors from them:

$$\begin{aligned} \mathbf{X}_{a+1} &= \mathbf{X}_a - \mathbf{t}_{a+1} \mathbf{p}_{a+1}^t \\ \mathbf{y}_{a+1} &= \mathbf{y}_a - \mathbf{t}_{a+1} q_{a+1} \end{aligned}$$

- (4) Go to step (2) to compute the next latent vector, until reaching A latent vectors ($a = A$)
- (5) Store \mathbf{w} , \mathbf{t} , \mathbf{p} and q in \mathbf{W} , \mathbf{T} , \mathbf{P} , and \mathbf{q} respectively.
- (6) Calculate the final regression coefficients: $\mathbf{b} = \mathbf{W}^t (\mathbf{P}\mathbf{W}^t)^{-1} \mathbf{q}$ [28], where \mathbf{W} ($J \times A$) and \mathbf{P} ($J \times A$) are matrices whose columns are the vectors \mathbf{w} and \mathbf{p} , respectively.

3.2. Modified NIPALS algorithm (NIPALSy)

Dayal and Macgregor [24] have shown that only one of either the \mathbf{X} or the \mathbf{Y} matrix needs to be deflated. Since only the ($I \times 1$) \mathbf{y} vector is deflated after each latent vector computation, the speed of the NIPALS algorithm is improved.

3.3. Kernel algorithm

The kernel algorithm presented by Lindgren *et al.* [25] was developed for matrices with a large number of objects and relatively few predictor variables. A complete PLS solution can be obtained by handling the condensed kernel matrix $\mathbf{X}^t \mathbf{y} \mathbf{y}^t \mathbf{X}$, usually computed using the cross product of $\mathbf{X}^t \mathbf{y}$, that is $(\mathbf{X}^t \mathbf{y})(\mathbf{X}^t \mathbf{y})^t$. This procedure avoids the need to deflate the kernel matrix, and the two covariance matrices, $\mathbf{X}^t \mathbf{X}$ and $\mathbf{X}^t \mathbf{y}$, are of a considerably smaller size than the original matrices \mathbf{X} and \mathbf{y} .

The Kernel algorithm is given below:

- (1) Compute the covariance matrices $\mathbf{X}^t \mathbf{X}$ and $\mathbf{X}^t \mathbf{y}$, and the kernel matrix $\mathbf{X}^t \mathbf{y} \mathbf{y}^t \mathbf{X}$.
- (2) The PLS weight vector \mathbf{w}_a is computed as the eigenvector corresponding to the largest eigenvalue of $(\mathbf{X}^t \mathbf{y} \mathbf{y}^t \mathbf{X})_a$.
- (3) The PLS loading vectors \mathbf{p}_a and q_a are computed as

$$\begin{aligned} \mathbf{p}_a^t &= \frac{\mathbf{w}_a^t (\mathbf{X}^t \mathbf{X})_a}{\mathbf{w}_a^t (\mathbf{X}^t \mathbf{X})_a \mathbf{w}_a} \\ q_a &= \frac{\mathbf{w}_a^t (\mathbf{X}^t \mathbf{y})_a}{\mathbf{w}_a^t (\mathbf{X}^t \mathbf{X})_a \mathbf{w}_a} \end{aligned}$$

- (4) After each latent vector computation, the covariance matrices $\mathbf{X}^t\mathbf{X}$ and $\mathbf{X}^t\mathbf{y}$ can be updated as

$$(\mathbf{X}^t\mathbf{X})_{a+1} = (\mathbf{I} - \mathbf{w}_a\mathbf{p}_a^t)(\mathbf{X}^t\mathbf{X})_a(\mathbf{I} - \mathbf{w}_a\mathbf{p}_a^t)$$

$$(\mathbf{X}^t\mathbf{y})_{a+1} = (\mathbf{I} - \mathbf{w}_a\mathbf{p}_a^t)(\mathbf{X}^t\mathbf{y})_a$$

- (5) Calculate the regression vector as in the NIPALS algorithm.

Based on the fact that only the deflation of \mathbf{y} in $\mathbf{X}^t\mathbf{y}$ is required, Dayal and MacGregor [24] proposed a modification that improved the original kernel algorithm and that is the version tested in this work.

3.4. SIMPLS algorithm

The SIMPLS algorithm, proposed by De Jong [26], derives the PLS factors directly as linear combinations of the original (centered) \mathbf{X} variables. One advantage of this method is that it is not necessary to deflate \mathbf{X} or \mathbf{y} , which may result in faster computation and less memory requirements.

When applied to a single-dependent variable \mathbf{y} , the results obtained by the SIMPLS algorithm turn out to be essentially the same as those obtained by the NIPALS algorithm. The SIMPLS algorithm for PLS1 can be summarized as follows:

- (1) Compute \mathbf{s} as

$$\mathbf{s} = \mathbf{X}^t\mathbf{y}$$

- (2) Compute the quantities \mathbf{r} (PLS weights for \mathbf{X}), \mathbf{t} (PLS scores for \mathbf{X}), q (PLS loading for \mathbf{y}) and \mathbf{p} (PLS loadings for \mathbf{X}) as follows:

$$\mathbf{r}_a = \mathbf{s}$$

$$\mathbf{t}_a = \mathbf{X}\mathbf{r}_a$$

$$\mathbf{t}_a = \frac{\mathbf{t}_a}{\|\mathbf{t}_a\|}$$

$$\mathbf{r}_a = \frac{\mathbf{r}_a}{\|\mathbf{r}_a\|}$$

$$\mathbf{p}_a = \mathbf{X}^t\mathbf{t}_a$$

$$q_a = \mathbf{y}^t\mathbf{t}_a$$

- (3) Store \mathbf{r} , \mathbf{t} , q and \mathbf{p} in \mathbf{R} , \mathbf{T} , \mathbf{q} and \mathbf{P} , respectively.

- (4) Project \mathbf{s} on a subspace orthogonal to \mathbf{P}_a

$$\mathbf{s} = \mathbf{s} - \mathbf{P}(\mathbf{P}^t\mathbf{P})^{-1}\mathbf{P}^t\mathbf{s}$$

- (5) Go to step (2) to compute the next latent vector until reaching A latent vectors

- (6) Calculate the regression vector as

$$\mathbf{b} = \mathbf{R}\mathbf{q}$$

3.5. The bidiagonalization algorithm (PLSBI)

Manne [27] has shown that PLS1 is equivalent to an algorithm developed by Golub and Kahan [2] for matrix bidiagonalization. Matrix bidiagonalization is a useful decomposition often employed as a fast initialization in algorithms for singular value decomposition [1].

This method considers that any matrix $\mathbf{X}(I \times J)$ can be written as $\mathbf{X} = \mathbf{U}\mathbf{R}\mathbf{V}^t$, where $\mathbf{U}(I \times J)$ and $\mathbf{V}(I \times J)$ are matrices with orthonormal columns, i.e. they satisfy $\mathbf{U}^t\mathbf{U} = \mathbf{V}^t\mathbf{V} = \mathbf{I}$, and $\mathbf{R}(J \times J)$ is a bidiagonal matrix.

Several papers in the literature describe the relation between PLS1 and bidiagonal decomposition [27,29–32]. The PLSBI algorithm can be summarized as follows [29,31]:

- (1) Initialize the algorithm for the first component, $\mathbf{v}_1 = \mathbf{X}^t\mathbf{y}/\|\mathbf{X}^t\mathbf{y}\|$; $\alpha_1\mathbf{u}_1 = \mathbf{X}\mathbf{v}_1$

- (2) Compute the following values for $a = 2, \dots, A$ latent variables

$$2.1. \gamma_{a-1}\mathbf{v}_a = \mathbf{X}^t\mathbf{u}_{a-1} - \alpha_{a-1}\mathbf{v}_{a-1}$$

$$2.2. \alpha_a\mathbf{u}_a = \mathbf{X}\mathbf{v}_a - \gamma_{a-1}\mathbf{u}_{a-1}$$

with

$$\mathbf{V}_A = (\mathbf{v}_1, \dots, \mathbf{v}_A), \mathbf{U}_A = (\mathbf{u}_1, \dots, \mathbf{u}_A) \text{ and}$$

$$\mathbf{R}_A = \begin{pmatrix} \alpha_1 & \gamma_1 & & & & \\ & \alpha_2 & \gamma_2 & & & \\ & & \ddots & \ddots & & \\ & & & \alpha_{A-1} & \gamma_{A-1} & \\ & & & & \alpha_A & \end{pmatrix}$$

It can be proved that $\mathbf{X}\mathbf{V}_A = \mathbf{U}_A\mathbf{R}_A$ and, therefore, $\mathbf{R}_A = \mathbf{U}_A^t\mathbf{X}\mathbf{V}_A$.

Once the matrices \mathbf{U} , \mathbf{V} and \mathbf{R} are computed with A components truncated in \mathbf{R} , one can estimate the Moore–Penrose pseudo-inverse of \mathbf{X} and solve the least squares problem as

$$\mathbf{y} = \mathbf{X}\mathbf{b} \rightarrow \mathbf{y} = \mathbf{U}_A\mathbf{R}_A\mathbf{V}_A^t\mathbf{b} \rightarrow \mathbf{b} = \mathbf{V}_A\mathbf{R}_A^{-1}\mathbf{U}_A^t\mathbf{y}$$

4. EXPERIMENTAL

This section is divided into two main parts: the first one deals with simulated data sets especially designed to cover a wide range of data sizes, with the aid of factorial and Latin square designs and in the second, real data sets of different sizes and nature were investigated.

For the sake of clarity, the columns of \mathbf{X} matrices are referred to as *variables* and the variables studied in the experimental designs are designated as *factors*.

4.1. Simulated data sets


4.1.1. Factorial designs

Two full factorial designs [33,34], 2^3 , with triplicate in the central point were proposed to investigate two sizes of data sets: small (SX) and large (LX) \mathbf{X} matrices. A total of 11 experiments were performed for each design, eight at the factorial levels and three at the central point level. Each PLS algorithm was run for both designs and the experiments at the central point were performed for error estimation. The predictor (\mathbf{X}) and dependent (\mathbf{y}) variables were generated using a pseudo-random number generator. The response investigated in the experimental design was the running time of the algorithms during leave-one-out cross-validations and designated from here on as *time*. Three factors were investigated: the number of rows, R , and the number of columns, C , from \mathbf{X} , and the number of PLS latent variables, n_{LV} . Table I summarizes the variables and the explored domain. The matrix dimensions are described by levels of row and column factors. All data were mean centered as a standard preprocessing procedure.

Assuming that there is a functional relation between the experimental variables and the observed running time in the

Table I. Factors, coded levels and investigated domain for the 2^3 full factorial designs

Factors	SX levels			LX levels		
	-1	0	1	-1	0	1
Rows (<i>R</i>)	20	60	100	100	550	1000
Columns (<i>C</i>)	50	275	500	500	2750	5000
Latent variables (<i>nLV</i>)	8	12	16	10	15	20



described domain, the following response surface model with linear and interaction terms was determined.

$$\text{time} = \beta_0 + \beta_1 R + \beta_2 C + \beta_3 nLV + \beta_{12} R \times C + \beta_{13} R \times nLV + \beta_{23} C \times nLV + e \quad (1)$$

The estimated $\hat{\beta}_0$ is the average of all the running time values of the design. The main and interaction effects are the estimated model parameters multiplied by 2. The effects can also be calculated by the following equations:

$$\text{Mean} = \frac{\sum_{i=1}^n \text{time}_i}{n} \quad (2)$$

$$ef = \frac{\sum_{i=1}^{n/2} \text{time}_{i(+)} - \sum_{i=1}^{n/2} \text{time}_{i(-)}}{n/2} \quad (3)$$

where n is the number of assays and time_i is an individual observation given by the PLS run time during leave-one-out cross-validation.

Equation (2) describes the mean effect of all observations, while Equation (3) stands for the effects for factors and interactions using the difference between the mean of observations in the high level ($\text{time}_{i(+)}$) and the mean of observations in the low level ($\text{time}_{i(-)}$).

In this work, the standard errors for the effects were obtained by the mean square residual (MS residual), according to Equation (4), because the pure error presented a very low value due to the high precision of replicates.

$$\text{MS residual} = \frac{\sum_{i=1}^m \sum_{j=1}^r (\text{time}_{ij} - \widehat{\text{time}}_i)^2}{n - q} \quad (4)$$

In this equation, m is the total level number (experimental design points); r is the total replicate number; $n - q$ is the number of degrees of freedom of MS residual; n is the number of assays, q is the number of calculated parameters (coefficients or effects) and $\widehat{\text{time}}_i$ is the estimated running time of the model. The error due to the factorial design was obtained as described in Equation (5).

$$\text{Err} = \sqrt{\frac{\text{MS residual}}{n}} \quad (5)$$

4.1.2. Latin square designs

Latin square designs are adequate when the factors of interest have more than two levels and it is previously known that there are no (or only negligible) interactions between them. The aim is to estimate the main effects by investigating several levels for each factor.

A Latin square of order n is an $n \times n$ array in which each cell contains a set of n symbols, in such a way that each symbol occurs only once in each row and once in each column.

In this work, a 5×5 Latin square design with two replications was used to investigate the influence of several levels of variables over the run time for the five PLS algorithms. Five levels for each factor (number of rows, columns and nLV) were studied and a total of 50 experiments were carried out for each PLS algorithm (Table II). All data were mean centered as standard preprocessing.

Table II shows the large number of rows, columns and latent variables investigated. In this study sample dominant and variable dominant matrices were considered, covering a wide number of possibilities that could be found in the real world.

The statistical evaluation was performed using the analysis of variance (ANOVA) as well others described in the literature [33,34].

4.2. Real data sets

Six data sets from real applications were explored. They were obtained from different sources, i.e. Near-infrared (NIR) spectroscopy, Raman spectroscopy, fluorescence spectroscopy, gas chromatography (GC), voltammetry and finally, one UV-like spectra data set simulated using a Gaussian distribution

Table II. Levels studied for each factor in the Latin square design

Levels		
<i>R</i>	<i>C</i>	<i>nLV</i>
50	200	3
100	500	5
200	1000	10
500	5000	15
1000	10000	20

R: number of rows; *C*: number of columns; and *nLV*: number of latent variables.

generator. All data sets were investigated using three levels of latent variables ($nLV = 3, 5$ and 10) for each algorithm.

NIR data set: This data set was measured at the Southwest Research Institute (SWRI) in a project sponsored by the US Army. It is formed by 231 rows and 401 columns, as acquired from the Eigenvector Research homepage at <http://www.eigenvector.com>. Freeze—the freezing temperature of the fuel ($^{\circ}\text{C}$) is the modeled physical property.

Raman data set: This data set is available at <http://www.models.kvl.dk/research/data/> as it was presented previously by Dyrby *et al.* [35]. It consists of Raman scattering for 120 samples and 3401 wavenumbers in the range of $200\text{--}3600\text{ cm}^{-1}$ (interval, 1 cm^{-1}). The dependent variable refers to the relative amount of active substance in Escitalopram[®] tablets in %w/w units.

Fluorescence data set: This data set was designed by Bro *et al.* [36] for the study of several topics in fluorescence spectroscopy and can be found at <http://www.models.kvl.dk/research/data/>. The dependent variable in this case is the hydroquinone concentration. An unfolding was performed prior to PLS regression yielding a matrix with 405 rows and 2548 columns.

Voltammetry data set: This data set was obtained from Teófilo *et al.* [37] and consists of 62 baseline corrected voltamograms. The predictors (variables) are the oxidation current from mixtures of guaiacol and chloroguaiacol with potential varying from 0.5 to 1.2 mV (353 variables) while the analyte investigated was guaiacol.

UV-like data set: Spectra with Gaussian distribution from four different analytes were used to generate 1000 mixtures with concentrations given by pseudo-random numbers. The matrix used in this case was formed by 1000 rows and 150 columns.

Chromatogram data set: This data set was presented by Ribeiro *et al.* [7] and contains the pretreated chromatograms of 62 Brazilian Arabica roasted coffee samples with retention times varying from 1.8 to 19 s by 0.00085 s steps (20 640 variables). The dependent variable was the sensory attribute flavor of the roasted coffee samples.

All calculations using PLS algorithms were carried out on MATLAB 7.0 (MathWorks, Natick, USA) in double precision, installed on a PC with windows XP operating system, 1.86 GHz Intel core 2 duo processor, 2 GB RAM memory. The experimental design calculations were carried out using Excel spreadsheets according to Teófilo and Ferreira [33].

The precision of the algorithms, regarding the cross-validation error, was defined by the difference between the values of root mean square error of cross-validation (RMSECV) obtained from each assay, according to Equation (6),

The comparison of cross-validation results between algorithms i and j was quantified by

$$\text{diff}_{ij} = |\text{RMSECV}_i - \text{RMSECV}_j| \quad (6)$$

where RMSECV is given by Equation (7)

$$\text{RMSECV}_k = \sqrt{\frac{\sum_{i=1}^l (y_i - \hat{y}_i)^2}{l}} \quad (7)$$

In Equation (7), y_i is the measured response of the i th sample, \hat{y}_i is the predicted response from calibration equation obtained for the data without the i th sample and l is the number of samples in the calibration set. Cross-validation was performed using a lab-built algorithm written for MATLAB 7.0.

5. RESULTS AND DISCUSSION

5.1. Simulated data sets

5.1.1. Factorial design

The effects obtained for the five algorithms considering both SX and LX data sets from full factorial design models are shown in Table III.

According to Equation (3), the effect is the difference between the mean running times obtained for the levels of each factor and, thus, its value must be related to run time. The effect indicates the influence of a factor or interaction between two factors over the run time. The error (Err) is obtained from Equation (5) and t is the ratio Effect/Err, the parameter of Student distribution. The t value with specified degrees of freedom and significance level, α , obtained from t distribution available in statistical Tables [34] or from the p value [33,34], is used to judge whether the effect is statistically significant.

As the calculations were performed under the same conditions for different algorithms, it is possible to compare the effects and responses between the algorithms and between the data sets. Thus, observing the SX data set from Table III, it can be noted that both factors (R and C) are significant. However, C is approximately 62% ((Effect $C \times 100$)/Effect R) more important than R . The interaction $R \times C$ was also important in all calculations, being even more important than nLV which had only a minor importance.

When analyzing the results for the LX data set in Table III, it can be seen that only the main factors R and C are significant. Unlike SX, the factor R is approximately 86% ((Effect $R \times 100$)/Effect C) more important than C . This inversion with respect to R and C significance is related to the cross-validation procedure. The increase in the number of rows increases the number of steps in leave-one-out cross-validation and, consequently, the run time. Unlike for SX, nLV is not significant for LX within the studied levels. In this case, the effect of nLV over run time could be minimized by its smaller range relative to R and C . Otherwise, the interaction $R \times C$ is very important in all calculations.

After discussing the significance of each factor and their interactions for the models, it is necessary to address the difference in the running times for the various algorithms. Since the same data set was tested by various algorithms, the paired t -test is the choice to test if running time for algorithms i and j are statistically different.

The null hypothesis in this case is that the mean difference between running time for algorithms i and j is zero, which means that there is no statistical evidence that the computation times are different for the two algorithms. The alternative hypothesis is that the running times for the two algorithms are different. For the paired t -test, the difference between the running times is calculated for each pair i, j and the mean and standard deviation of these differences are calculated. Dividing the mean by the standard deviation of the mean yields a t value that is t -distributed with $n - 1$ degrees of freedom. The null hypothesis was rejected at a significance level of 0.05 when t calculated $> t$ critical or $p \leq 0.050$, where the p -value is the smallest level of significance that would lead to rejection of the null hypothesis H_0 with the given data [34].

Table IV presents the results obtained. Note that for the SX data set, the algorithm SIMPLS was statistically equal to Kernel and for the LX data set, the pairs PLSBi-SIMPLS and SIMPLS-Kernel

Table III. Full factorial models for the five algorithms using SX and LX data sets

	PLSbi				SIMPLS				Kernel				NIPALSy				NIPALS			
	Effect	Err	t	t	Effect	Err	t	t	Effect	Err	t	t	Effect	Err	t	Effect	Err	t	t	
	SX																			
Mean	0.21	0.01	15.96	16.58	0.23	0.01	16.58	0.24	0.02	13.90	0.27	0.02	13.43	0.47	0.05	9.40				
R	0.22	0.03	7.37	7.26	0.24	0.03	7.26	0.28	0.04	6.84	0.29	0.05	6.26	0.68	0.12	5.76				
C	0.35	0.03	11.50	11.33	0.37	0.03	11.33	0.39	0.04	9.74	0.46	0.05	9.74	0.83	0.12	7.09				
nLV	0.12	0.03	4.01	3.91	0.13	0.03	3.91	0.17	0.04	4.14	0.19	0.05	4.09	0.36	0.12	3.03				
R × C	0.18	0.03	6.06	6.08	0.20	0.03	6.08	0.23	0.04	5.70	0.25	0.05	5.24	0.60	0.12	5.09				
R × nLV	0.05	0.03	1.68	2.02	0.07	0.03	2.02	0.08	0.04	2.02	0.09	0.05	1.92	0.23	0.12	1.97				
C × nLV	0.10	0.03	3.23	2.74	0.09	0.03	2.74	0.12	0.04	2.99	0.14	0.05	3.08	0.29	0.12	2.50				
	LX																			
Mean	200.48	29.35	6.83	6.95	206.14	29.66	6.95	210.33	30.72	6.85	284.95	45.05	6.32	584.93	93.84	6.23				
R	443.52	68.83	6.44	6.54	454.97	69.55	6.54	461.39	72.05	6.40	630.97	105.66	5.97	1294.97	220.07	5.88				
C	384.65	68.83	5.59	5.74	399.50	69.55	5.74	407.76	72.05	5.66	561.66	105.66	5.32	1110.20	220.07	5.04				
nLV	108.90	68.83	1.58	1.55	107.71	69.55	1.55	117.10	72.05	1.63	175.77	105.66	1.66	396.07	220.07	1.80				
R × C	375.12	68.83	5.45	5.59	388.89	69.55	5.59	395.26	72.05	5.49	547.01	105.66	5.18	1083.66	220.07	4.92				
R × nLV	105.51	68.83	1.53	1.49	103.78	69.55	1.49	111.77	72.05	1.55	169.75	105.66	1.61	385.43	220.07	1.75				
C × nLV	94.34	68.83	1.37	1.36	94.82	69.55	1.36	103.36	72.05	1.43	155.64	105.66	1.47	333.45	220.07	1.52				

Err: from mean square residual; t: ratio Effect/Err; the parameter of Student distribution. R: number of rows; C: number of columns; and nLV: number of latent variables. Bold and italic effects with four degrees of freedom are statistically significant, $\alpha = 0.05$.

Table IV. Comparison for run time differences between algorithms using paired t-test for SX and LX data sets

SX										
	PLSBI	SIMPLS	PLSBI	Nipalsy	PLSBI	Nipals	PLSBI	Kernel	SIMPLS	Nipalsy
Mean	0.21	0.23	0.21	0.27	0.21	0.47	0.21	0.24	0.23	0.27
Variance	0.05	0.05	0.05	0.09	0.05	0.37	0.05	0.07	0.05	0.09
Correlation	1.00		1.00		0.99		1.00		1.00	
t_0	4.91		2.65		2.24		2.34		1.87	
p	0.0003		0.012		0.025		0.021		0.045	
LX										
	PLSBI	SIMPLS	PLSBI	Nipalsy	PLSBI	Nipals	PLSBI	Kernel	SIMPLS	Nipalsy
Mean	200.48	206.14	200.48	284.95	200.48	584.93	200.48	210.33	206.14	284.95
Variance $\times 10^5$	1.07	1.14	1.07	2.28	1.07	9.39	1.07	1.19	1.14	2.28
Correlation	1.00		1.00		1.00		1.00		1.00	
t_0	1.79		1.86		1.98		1.88		1.85	
p	0.052		0.047		0.038		0.045		0.047	
	SIMPLS	Nipals	SIMPLS	Kernel	Nipalsy	Nipals	Nipalsy	Kernel	Nipals	Kernel
Mean	206.14	584.93	206.14	210.33	284.95	584.93	284.95	210.33	584.93	210.33
Variance $\times 10^5$	1.14	9.39	1.14	1.19	2.28	9.39	2.28	1.19	9.39	1.19
Correlation	1.00		1.00		1.00		1.00		1.00	
t_0	1.98		-1.50		2.02		1.85		1.99	
p	0.038		0.082		0.035		0.047		0.038	

Degrees of freedom: 10; significance level: 0.05; and t -critical: 1.81.
The bold and italic numbers indicate the null hypothesis was accepted.

were statistically equal. In the other comparisons the times were statistically different indicating the necessity to evaluate which algorithm should be used.

The performance of the five algorithms given in terms of run time can be observed in Figure 1(A) where the effects give a measure of the run time. Note that PLSBI, SIMPLS and Kernel show equivalent performance; NIPALSy is slightly worse and

NIPALS has the poorest performance. It is clear that by using the deflation only in y , the NIPALS algorithm is significantly improved with respect to the run time.

The effects for the algorithms are shown in Figure 1(B), where a similar trend to that for SX can be observed.

Table V shows the relative precision, regarding the cross-validation results, calculated as described in Equation (6) for the

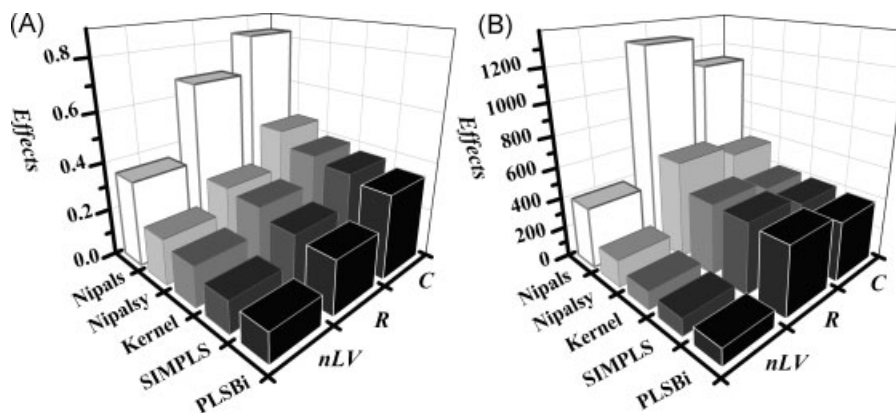


Figure 1. Effects obtained from the full factorial design models for SX (A) and LX (B) data sets.

Table V. Difference in RMSECV values (Equation (6)) among assays for SX and LX data sets^a

R	C	nLV	Bi-Si	Bi-Niy	Bi-Ni	Bi-K	Si-Niy	Si-Ni	Si-K
Factorial design SX									
20	50	8	1.91×10^{-14}	0	0	0	1.90×10^{-14}	1.90×10^{-14}	1.90×10^{-14}
100	50	8	7.52×10^{-13}	0	0	0	7.51×10^{-13}	7.51×10^{-13}	7.51×10^{-13}
20	500	8	2.33×10^{-15}	0	0	0	2.50×10^{-15}	2.55×10^{-15}	2.55×10^{-15}
100	500	8	5.27×10^{-15}	0	0	0	5.27×10^{-15}	5.27×10^{-15}	5.27×10^{-15}
20	50	16	6.20×10^{-12}	0	0	0	6.20×10^{-12}	6.20×10^{-12}	6.20×10^{-12}
100	50	16	1.84×10^{-4}	0	0	0	1.84×10^{-4}	1.84×10^{-4}	1.84×10^{-4}
20	500	16	1.24×10^{-14}	5.00×10^{-16}	6.11×10^{-16}	5.55×10^{-16}	1.19×10^{-14}	1.18×10^{-14}	1.18×10^{-14}
100	500	16	1.77×10^{-11}	0	0	0	1.77×10^{-11}	1.77×10^{-11}	1.77×10^{-11}
60	275	12	7.32×10^{-14}	0	0	0	7.32×10^{-14}	7.32×10^{-14}	7.32×10^{-14}
60	275	12	3.61×10^{-13}	0	0	0	3.61×10^{-13}	3.61×10^{-13}	3.61×10^{-13}
60	275	12	1.08×10^{-13}	0	0	0	1.08×10^{-13}	1.08×10^{-13}	1.08×10^{-13}
Factorial design LX									
100	500	10	1.81×10^{-14}	0	0	0	1.82×10^{-14}	1.82×10^{-14}	1.83×10^{-14}
1000	500	10	1.17×10^{-15}	4.44×10^{-16}	0	0	7.22×10^{-16}	8.88×10^{-16}	8.33×10^{-16}
100	5000	10	1.59×10^{-09}	0	0	0	1.59×10^{-09}	1.59×10^{-09}	1.59×10^{-09}
1000	5000	10	4.33×10^{-15}	0	0	0	4.50×10^{-15}	4.55×10^{-15}	4.66×10^{-15}
100	500	20	4.17×10^{-10}	0	0	0	4.17×10^{-10}	4.17×10^{-10}	4.17×10^{-10}
1000	500	20	3.00×10^{-15}	9.99×10^{-16}	1.05×10^{-15}	9.99×10^{-16}	4.00×10^{-15}	4.05×10^{-15}	4.00×10^{-15}
100	5000	20	0.10	7.22×10^{-16}	7.22×10^{-16}	1.44×10^{-15}	0.10	0.10	0.10
1000	5000	20	1.25×10^{-10}	0	0	0	1.25×10^{-10}	1.25×10^{-10}	1.25×10^{-10}
550	2750	15	3.27×10^{-12}	0	0	0	3.27×10^{-12}	3.27×10^{-12}	3.27×10^{-12}
550	2750	15	4.52×10^{-13}	0	0	0	4.52×10^{-13}	4.52×10^{-13}	4.52×10^{-13}
550	2750	15	3.87×10^{-12}	0	0	0	3.87×10^{-12}	3.87×10^{-12}	3.87×10^{-12}

R: number of rows; C: number of columns; and nLV: number of latent variables. The values to Niy-K, Ni-K and Niy-Ni are equal to zero.

^a PLSBi (Bi), SIMPLS(Si), Kernel (K), NIPALS (Ni), NIPALSy (Niy).

Table VI. ANOVA results using Latin square design for the five algorithms

	SS	df	MS	F
PLSBI				
R	1824949	4	456237.3	7.48
C	1250554	4	312638.5	5.13
nLV	866337	4	216584.3	3.55
Residual	2255381	37	60956.3	
SIMPLS				
R	2001938	4	500484.6	7.74
C	1336032	4	334008	5.16
nLV	919261	4	229815.1	3.55
Residual	2393615	37	64692.3	
Kernel				
R	1982460	4	495615	7.43
C	1375964	4	343991	5.15
nLV	947643	4	236910.8	3.55
Residual	2469374	37	66739.8	
NIPALSy				
R	3618584	4	904646	7.08
C	2550944	4	637735.9	4.99
nLV	1796580	4	449145	3.52
Residual	4725097	37	127705.3	
NIPALS				
R	14190920	4	3547730	7.09
C	9703320	4	2425830	4.84
nLV	6977808	4	1744452	3.48
Residual	18525600	37	500692	

SS: Sums of Squares; df: degrees of freedom; MS: mean square residual; F: statistics ratio; α : 0.05; R: number of rows; C: number of columns; and nLV: number of latent variables. Bold and italic values are statistically significant.

algorithms tested using the SX and LX data sets. Significant differences between the algorithms are mostly observed at large nLV values. The SIMPLS algorithm showed results noticeably different from those of other algorithms for some specific matrix dimensions and large values of nLV. Other results indicate negligible differences among the algorithms, i.e. equal results for RMSECV.

5.1.2. Latin square design

Table VI shows the ANOVA results for the five algorithms. The sums of squares (SS) in Table VI are related to the variance in each factor. The higher the variance, the higher is the influence of a factor on the run time. The mean square (MS) is given by the ratio of sums of squares and the number of degrees of freedom (df) and better explains the results. F is the parameter of F-distribution for variance tests, and is obtained as the ratio of MS and MS Residual. Using the F-value for specific degrees of freedom and the significance level, α , it is possible to determine whether SS is statistically significant.

By analyzing the values of SS and MS from Table VI, the similarities among the PLSBi, SIMPLS and Kernel algorithms, and the high values for NIPALSy and especially NIPALS, can be

observed as before. In this case, the number of rows is approximately 68% more important than the number of columns, and nLV is slightly less important when compared to R and C.

When using MS residual to represent the run time, it is possible to observe in Figure 2, the behavior of the algorithms and the influence of the factors R, C and nLV. The large run time for the NIPALS algorithm as well as the better performance of PLSBi, SIMPLS and Kernel can also be seen.

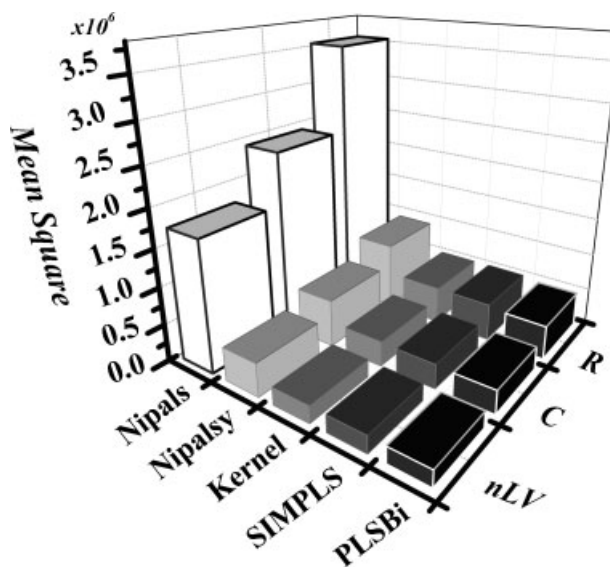
The plot for the main effects is shown in Figure 3, indicating the influence of the level of each factor on the run time. An exponential growth is observed in all cases due to the increase of the number of rows and columns. However, a drop in the run time is observed for the maximum nLV studied. This trend is due to the absence of investigation for the maximum level of nLV with the maximum level of R and C. The maximum level of nLV was studied only for the lower levels of R and C. As the nLV has little influence on the run time, the result obtained for the maximum level of nLV, as noted in the Figure 3, is not real. The real influence of nLV over time can be observed in Figure 4 for a fixed dimension of matrices, where a linear increase is observed.

The differences between run time algorithms for results using Latin square were calculated and statistically evaluated using the paired t-test as before.

Table VII presents the results obtained. Note that the algorithm SIMPLS was statistically equal to Kernel, in accordance to results obtained previously. In other comparisons, the times were statistically different indicating the necessity to evaluate which algorithm should be used.

Table VIII shows the precision analysis, regarding the cross-validation results, for the Latin square design. Three assays indicate a large difference in the RMSECV values. Observing these values it can be concluded that the number of latent variables is critical for matrices where the number of samples is approximately 2% (or less) of the number of variables. With a large number of latent variables (>10), great differences among the results obtained with PLS algorithms, mainly for the SIMPLS algorithm, can be observed.

However, the RMSECV differences for the other assays are very small ($<10^{-9}$), indicating that the algorithms are quite similar regarding the precision in most of the cases.

**Figure 2.** Mean square values from Latin square design.

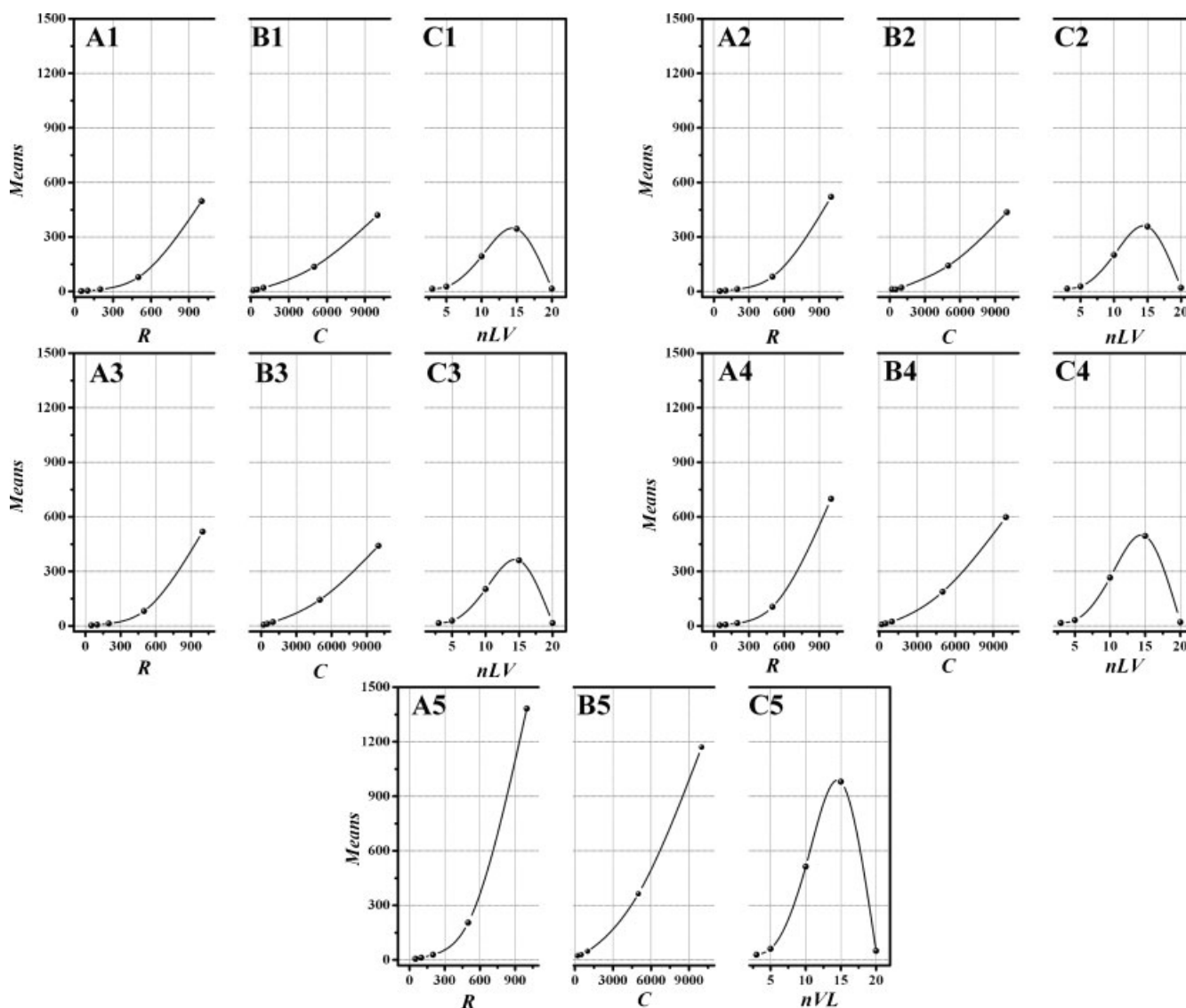


Figure 3. Main effect plots for Latin square design. PLSBi, A1, B1, C1; SIMPLS, A2, B2, C2; Kernel, A3, B3, C3; NIPALSy, A4, B4, C4; and NIPALS, A5, B5, C5.

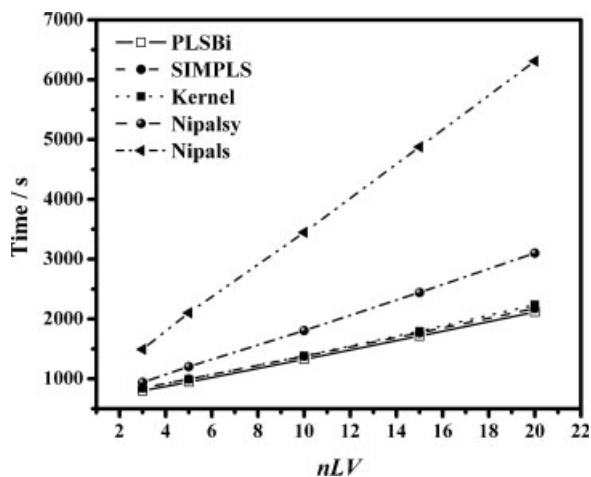


Figure 4. Run time versus nLV for a matrix 1000×10000 .

5.2. Real data sets

The spectra, voltammograms and chromatograms are shown in Figure 5. Note that each data set shows a different behavior and structure. These data were studied varying the number of latent variables for each algorithm applied.

Table IX contains the run time obtained for each real data set and algorithm. Note that the run time increases linearly with nLV for all algorithms and data sets. The best performance was obtained for the PLSBi algorithm and the worst performance was obtained for NIPALS in most of the cases. The Kernel algorithm was slightly better than the SIMPLS algorithm for all assays.

It was observed for most of the cases that the data type did not affect the behavior of the run time differences among the algorithms with respect to the random data. However, the UV-like data set presented an unexpected result because SIMPLS had the poorest performance (Table IX). This abnormal behavior can be justified by the matrix dimensions used in this data set. The number of rows (1000) is much higher than the number of columns (150). Thus, SIMPLS is not recommended to be used in matrices with such dimensions.

Table VII. Comparison for run time differences between algorithms using paired t-test for Latin square data set

	PLSBI	SIMPLS	PLSBI	NIPALSy	PLSBI	NIPALS	PLSBI	Kernel	SIMPLS	NIPALSy	Kernel	NIPALS	PLSBI	Kernel	SIMPLS	NIPALSy	Kernel
Mean	119.114	124.767	119.114	166.189	119.114	326.695	119.114	124.749	119.114	166.189	124.749	326.695	119.114	124.749	124.767	166.189	124.749
Variance $\times 10^5$	1.26	1.36	1.26	2.59	1.26	10.08	1.26	1.38	1.26	2.59	1.38	10.08	1.26	1.38	1.36	1.36	1.38
Correlation	0.9999	0.9998	0.9998	0.9998	0.9997	0.9998	0.9997	0.9998	0.9997	0.9999	0.9998	0.9998	0.9997	0.9998	0.9996	0.9998	0.9996
t_0	-2.898	-2.169	-2.169	-2.263	-2.263	-2.442	-2.263	-2.292	-2.263	-2.292	-2.133	-2.258	-2.442	-2.133	2.258	-2.079	2.258
p	0.0028	0.0175	0.0175	0.4929	0.0140	0.0091	0.0140	0.0131	0.0140	0.0131	0.0190	0.0142	0.0091	0.0142	0.0214	0.0214	0.0142
	SIMPLS	NIPALS	SIMPLS	Kernel	NIPALSy	NIPALS	NIPALSy	Kernel	NIPALSy	NIPALS	Kernel	NIPALS	NIPALSy	Kernel	NIPALS	Kernel	Kernel
Mean	124.767	326.695	124.767	124.749	166.189	326.695	166.189	124.749	166.189	326.695	124.749	326.695	166.189	124.749	326.695	124.749	124.749
Variance $\times 10^5$	1.36	10.08	1.36	1.38	2.59	10.08	2.59	1.38	2.59	10.08	1.38	10.08	2.59	1.38	10.08	1.38	1.38
Correlation	0.9996	0.9998	0.9998	0.9998	0.9999	0.9998	0.9999	0.9998	0.9999	0.9998	0.9998	0.9996	0.9998	0.9996	0.9996	0.9996	0.9996
t_0	-2.245	0.018	0.018	0.018	-2.292	0.018	-2.292	0.018	-2.292	2.133	2.258	2.258	2.133	2.258	2.258	2.258	2.258
p	0.0146	0.4929	0.4929	0.4929	0.0131	0.0131	0.0131	0.0131	0.0131	0.0190	0.0190	0.0142	0.0190	0.0142	0.0142	0.0142	0.0142

Degree of freedom: 49; significance level: 0.05; t-critical: 1.68.

The bold and italic number indicates the null hypothesis was accepted ($p > 0.05$).

Table VIII. Difference in RMSECV values (Equation (6)) among assays for Latin square design data sets^a

Latin square design		RMSECV difference									
R	C	nLV	Bi-Si	Bi-Niy	Bi-K	Bi-Ni	Bi-Ni	Si-Niy	Si-K	Niy-K	Ni-K
50	5000	15	6.85×10^{-3}	0	0	0	6.85×10^{-3}	6.85×10^{-3}	6.85×10^{-3}	0	0
50	10000	20	0.45	3.44×10^{-3}	2.28×10^{-3}	3.44×10^{-3}	3.44×10^{-3}	0.45	0.45	1.16×10^{-3}	1.16×10^{-3}
100	5000	20	0.53	4.33×10^{-15}	8.94×10^{-15}	4.27×10^{-15}	4.27×10^{-15}	0.53	0.53	4.61×10^{-15}	4.66×10^{-15}

R: number of rows; C: number of columns; and nLV: number of latent variables. The values of Niy-Ni are equal to zero. Other experiments were less than 1×10^{-9} .

^a PLSBI (Bi), SIMPLS (Si), Kernel (K), NIPALS (Ni), NIPALSy (Niy).

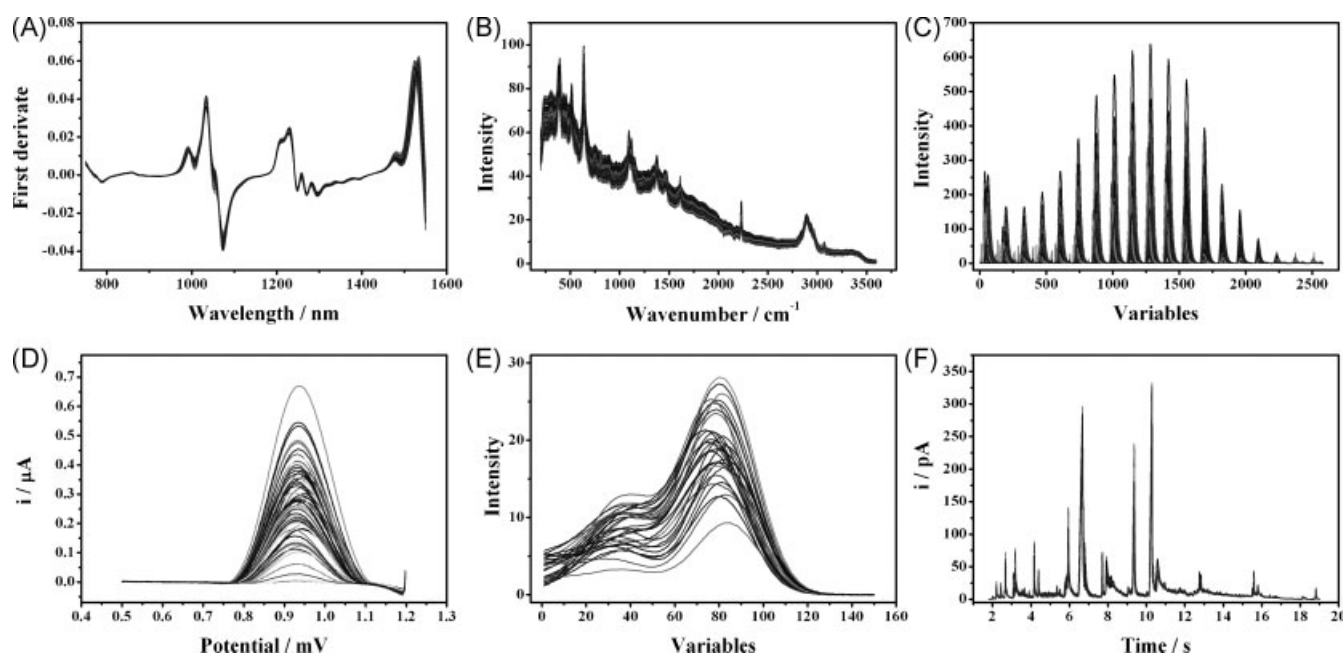


Figure 5. Data set tests used. (A) Near-infrared spectra (NIR); (B) Raman spectra (Raman); (C) unfolded fluorescence spectra (Fluor); (D) voltammograms (Volt); (E) UV-like data set (UV-like); and (F) gas chromatography (GC).

Table IX. Time (in seconds) of each algorithm varying the data set type, dimension and number of latent variables

Data set	Dimension	nLV^a	PLSbi	SIMPLS	Kernel	NIPALSy	NIPALS
NIR	231×401	3	1.16	1.24	1.17	1.27	2.33
		5	1.39	1.48	1.41	1.56	3.17
		10	1.86	1.94	1.91	2.23	5.25
Raman	120×3401	3	3.30	3.61	3.58	3.98	6.78
		5	4.17	4.64	4.53	5.34	9.72
		10	6.59	7.20	7.66	9.03	17.49
Fluor	405×2584	3	27.98	29.58	29.50	32.38	55.77
		5	33.52	35.17	35.14	40.80	77.38
		10	47.88	49.81	50.63	63.33	132.88
Volt	62×353	3	0.11	0.11	0.11	0.11	0.19
		5	0.14	0.14	0.13	0.14	0.28
		10	0.20	0.23	0.22	0.25	0.50
UV-like	1000×150	3	8.55	27.38	8.39	9.36	17.27
		5	9.88	28.25	9.41	10.92	23.38
		10	13.41	30.72	12.38	15.63	38.84
GC	58×20640	3	67.24	73.95	67.36	78.17	110.06
		5	74.72	78.89	78.92	91.09	144.00
		10	106.63	113.23	118.39	144.00	249.64

^a Number of latent variables.

The RMSECV difference (relative cross-validation error precision) for the six real data sets lies between 1.45×10^{-5} and 7.37×10^{-18} , indicating that there are no significant differences among the algorithms with respect to the cross-validation error precision.

6. CONCLUSIONS

The choice of the PLS algorithm for multivariate regression is an important issue when dealing with large data sets, due to

significant differences in running time of algorithms presented in the literature and, in some cases, because of important differences in RMSECV values.

For the matrices analyzed in this work, it is shown that the matrix dimension is the major factor responsible for computational time, while the number of latent variables has a lesser influence. In addition, in most of the cases the number of rows has greater influence than the number of columns for all algorithms. The number of latent variables exhibits a linear influence with increasing time, but it is less

important than the influence of the numbers of rows and columns.

Among the five algorithms analyzed in this work, PLSBi is the best with respect to computational time followed by Kernel and SIMPLS and the differences in speed although relatively small are statistically different. Comparing NIPALS to NIPALS_y, NIPALS_y was statistically faster because only **y** values need to be deflated. The values of RMSECV for all the algorithms tested were essentially the same in most of the cases. However, pronounced differences between RMSECV values were observed in some specific assays (with a high number of latent variables), especially for the SIMPLS algorithm. Further investigation is required for a theoretical explanation of such behavior.

Acknowledgements

The authors acknowledge CNPq and FAPESP for financial support and Dr Scott Ramos for providing the PLS the bidiagonal algorithm.

REFERENCES

1. Golub GH, Van Loan CF. *Matrix Computation*. John Hopkins University Press: Baltimore, 1996.
2. Golub GH, Kahan W. Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Num. Anal. Ser. B* 1965; **2**: 205–224.
3. Wold S, Johansson E, Cocchi M. *3D QSAR*. North Holland: Leiden, 1993.
4. Givehchi A, Dietrich A, Wrede P, Schneider G, ChemSpaceShuttle: a tool for data mining in drug discovery by classification, projection, and 3D visualization. *QSAR Comb. Sci.* 2003; **22**: 549–559.
5. Bao JS, Cai YZ, Corke H. Prediction of rice starch quality parameters by near-infrared reflectance spectroscopy. *J. Food Sci.* 2001; **66**: 936–939.
6. Alam TM, Alam MK. Chemometric analysis of NMR spectroscopy data: a review. *Annu. Rep. NMR Spectrosc.* 2005; **54**: 41–80.
7. Ribeiro JS, Augusto F, Salva TJG, Thomaziello RA, Ferreira MMC. Prediction of sensory properties of Brazilian Arabica roasted coffees by headspace solid phase microextraction-gas chromatography and partial least squares. *Anal. Chim. Acta* 2009; **634**: 172–179.
8. Gil DB, de la Pena AM, Arancibia JA, Escandar GM, Olivieri AC. Second-order advantage achieved by unfolded-partial least-squares/residual bilinearization modeling of excitation-emission fluorescence data presenting inner filter effects. *Anal. Chem.* 2006; **78**: 8051–8058.
9. Cruz SC, Aarnoutse PJ, Rothenberg G, Westerhuis JA, Smilde AK, Blik A. Kinetic and mechanistic studies on the Heck reaction using real-time near infrared spectroscopy. *Phys. Chem. Chem. Phys.* 2003; **5**: 4455–4460.
10. Martens H, Naes T. *Multivariate Calibration*. Wiley: New York, 1989.
11. Geladi P, Kowalski BR. Partial least-squares regression—a tutorial. *Anal. Chim. Acta* 1986; **185**: 1–17.
12. Hoskuldsson A. PLS regression methods. *J. Chemom.* 1988; **2**: 211–228.
13. Helland IS. Some theoretical aspects of partial least squares regression. *Chemom. Intell. Lab. Syst.* 2001; **58**: 97–107.
14. Wold S, Sjostrom M, Eriksson L. PLS-regression: a basic tool of chemometrics. *Chemom. Intell. Lab. Syst.* 2001; **58**: 109–130.
15. Rajah MN, McIntosh AR. Overlap in the functional neural systems involved in semantic and episodic memory retrieval. *J. Cogn. Neurosci.* 2005; **17**: 470–482.
16. Li XM, Zhou JX, Yuan SH, Zhou XP, Fu Q. The effects of tolerance for ambiguity and uncertainty on the appropriateness of accounting performance measures. *Biomed. Environ. Sci.* 2008; **41**: 45–52.
17. Pedro AMK, Ferreira MMC. Simultaneously calibrating solids, sugars and acidity of tomato products using PLS2 and NIR spectroscopy. *Anal. Chim. Acta.* 2007; **595**: 221–227.
18. Henrique CM, Teófilo RF, Sabino L, Ferreira MMC, Cereda MP. Classification of cassava starch films by physicochemical properties and water vapor permeability quantification by FTIR and PLS. *J. Food Sci.* 2007; **72**: E184–E189.
19. Ferreira MMC, Multivariate QSAR. *J. Braz. Chem. Soc.* 2002; **13**: 742–753.
20. Kiralj R, Ferreira MMC. Comparative chemometric and QSAR/SAR study of structurally unrelated substrates of a MATE efflux pump VmrA from *V. parahaemolyticus*: prediction of multidrug resistance. *QSAR Comb. Sci.* 2008; **27**: 314–329.
21. Wu W, Manne R. Fast regression methods in a Lanczos (or PLS-1) basis. Theory and applications. *Chemom. Intell. Lab. Syst.* 2000; **51**: 145–161.
22. Teófilo RF, Martins JPA, Ferreira MMC. Sorting variables by using informative vectors as a strategy for feature selection in multivariate regression. *J. Chemom.* 2009; **23**: 32–48.
23. Haaland DM, Thomas EV. Partial least-squares methods for spectral analyses. 1. Relation to other quantitative calibration methods and the extraction of qualitative information. *Anal. Chem.* 1988; **60**: 1193–1202.
24. Dayal BS, MacGregor JF. Improved PLS algorithms. *J. Chemom.* 1997; **11**: 73–85.
25. Lindgren F, Geladi P, Wold S. The kernel algorithm for PLS. *J. Chemom.* 1993; **7**: 45–59.
26. De Jong S. SIMPLS—an alternative approach to partial least-squares regression. *Chemom. Intell. Lab. Syst.* 1993; **18**: 251–263.
27. Manne R. Analysis of 2 partial-least-squares algorithms for multivariate calibration. *Chemom. Intell. Lab. Syst.* 1987; **2**: 187–197.
28. Helland IS. On the structure of partial least squares regression. *Commun. Stat-Simul. C.* 1988; **17**: 581–607.
29. Lorber A, Kowalski BR. A note on the use of the partial least-squares method for multivariate calibration. *Appl. Spectrosc.* 1988; **42**: 1572–1574.
30. Phatak A, de Hoog F. Exploiting the connection between PLS, Lanczos methods and conjugate gradients: alternative proofs of some properties of PLS. *J. Chemom.* 2002; **16**: 361–367.
31. Elden L. Partial least-squares vs. Lanczos bidiagonalization—I: analysis of a projection method for multiple regression. *Comput. Stat. Data Anal.* 2004; **46**: 11–31.
32. Pell RJ, Ramos LS, Manne R. The model space in partial least square regression. *J. Chemom.* 2007; **21**: 165–172.
33. Teófilo RF, Ferreira MMC. Chemometrics II: spreadsheets for experimental design calculations, a tutorial. *Quim. Nova* 2006; **29**: 338–350.
34. Montgomery DC, Runger GC. *Applied Statistics and Probability for Engineers*. Wiley: New York, 2003.
35. Dyrby M, Engelsen SB, Norgaard L, Bruhn M, Lundsberg-Nielsen L. Chemometric quantitation of the active substance (containing C=N) in a pharmaceutical tablet using near-infrared (NIR) transmittance and NIR FT-Raman spectra. *Appl. Spectrosc.* 2002; **56**: 579–585.
36. Bro R, Rinnan A, Faber NM. Standard error of prediction for multilinear PLS—2. Practical implementation in fluorescence spectroscopy. *Chemom. Intell. Lab. Syst.* 2005; **75**: 69–76.
37. Teófilo RF, Ceragioli HJ, Peterlevitz AC, Baranauskas V, Ferreira MMC, Kubota LT. Simultaneous determination of guaiaicol and chloroguaiaicol by SWV using boron-doped diamond electrode and PLS algorithms. In *10th International Conference on Chemometrics in Analytical Chemistry*, Águas de Lindóia, 2006, P009.