APLS: A Fast Approximate Algorithm for Partial Least-Squares

Ruy Luiz Milidiú
milidiu@inf.puc-rio.br
Raúl Pierre Rentería
renteria@inf.puc-rio.br

PUC-RioInf.MCC05/01 March, 2001

Abstract: APLS, an approximate algorithm for solving the Partial Least-Squares problem is presented. When dealing with more than one dependent variable in the standard PLS modeling, one must compute eigenvectors of the mixed independent and dependent variables matrix, thus introducing a time overhead in the algorithm. APLS provides an approximate method for this calculation with competitive modeling error and improved running time. In our experiments, APLS shows a 40% faster running time when compared to the standard PLS algorithm.

Keywords: PLS, Factor Analysis, chemometric

Resumo: APLS, um algoritmo aproximado para o problema de mínimos quadrados parciais é apresentado. Quando o modelo convecional PLS inclue mais de uma variável dependente, o cálculo dos autovetores da matriz composta pelas variáveis dependentes e independentes aumenta sensívelmente a complexidade computacional do algoritmo. Por outro lado, APLS fornece um método aproximado para este cálculo possuindo uma modelagem competitiva e tempo de execução nítidadmente superior. Em nossos experimentos, APLS mostrou um tempo de execução 40 % mais rápido quando comparado com o algoritmo convencional.

Palavras-Chave: PLS, Análise Fatorial, quimiometria

1 Introduction

The PLS algorithm (Geladi and Kowalski, 1986) has been widely used as a chemometric tool for Near-Infrared spectral analysis (Haaland and Thomas, 1988a,b; Beebe and Kowalski, 1987). The simplicity of the technique and robustness of the generated model make the partial least-squares approach a powerful tool for factor analysis, being applied to many other areas such as process monitoring, marketing analysis and image processing (Morineau and Tenenhaus, 1999; Milidiú et al., 1998, 1999).

In this paper, we propose APLS, an approximate algorithm for solving the Partial Least-Squares problem. In the classical PLS modeling, the eigenvectors of the mixed independent and dependent variables matrix must be computed in the case of two or more dependent variables. This can be accomplished for example using neural techniques such as Hebbian learning (Haykin, 1999). APLS provides an approximate method for this calculation, not relying on any convergence criteria.

In order to measure the performance of APLS, we report some experiments on nine data sets mainly related to NIR spectra analysis, such as wheat data for chemometrics (Kalivas, 1997) or combustible (Wentzell et al., 1999).

The APLS model shows a competitive quality when compared to the standard PLS model. Similar prediction errors and corresponding number of factors were observed. Moreover, since it is not a convergence dependent technique, a at least 45 % faster running time has been observed.

In section 2, we present the standard PLS algorithm. In section 3, our approximate version is described. In section 4, a running time analysis of the algorithm is made. In section 5, the empirical results obtained with the selected data set are shown. Finally, in section 6, we summarize our findings.

2 PLS Algorithm

2.1 Modeling

Partial Least Squares (PLS) is a multivariate statistical method, based on the use of factors, which is aimed at prediction (Geladi and Kowalski, 1986). The goal is to predict the values of a set of variables y based on the observed values of a set of variables x. As an example, x may be formed by the values of a time series window and y be taken as the value of a single future observation, or as the values of a set of future points in the same time series.

The construction of a PLS model requires a set of observation samples (patterns) and also their corresponding values of the dependent variables. Let X be the matrix containing in its rows the patterns of observations and Y be the matrix containing in its rows the effective values to be predicted.

The PLS method is a modeling procedure that simultaneously estimates underlying factors in both X and Y. These factors are then used to define a subspace in X that is more adequate to model Y. With Principal Components Regression (PCR) (Beebe and Kowalski, 1987), the rotation defined by the eigenvectors is used to find a subspace in X that subsequently is used to model Y. The approach taken by PLS is very similar to that of Principal Component Analysis (PCA) (Lebart et al., 1997), except that factors are chosen to describe the variables in Y as well as in X. This is accomplished by using the columns of the Y matrix to estimate the factors for X. At the same time, the columns of X are used to estimate the factors of Y. The resulting models are

$$X = TP + E$$

$$Y = UQ + F$$

where the elements of T and U are called the scores of X and Y, respectively, and the elements of P and Q are called the loadings. The matrices E and F are the errors associated with modeling X and Y with the PLS model.

The T factors are not optimal for estimating the columns of X as is the case with PCA , but are rotated so as to simultaneously describe the Y matrix.

In the ideal situation, the sources of variation in X are exactly equal to the sources of variation in Y, and the factors for X and Y are identical. In real applications, X varies in ways not correlated to the variation in Y, and therefore $t \neq u$ (t and u are columns of T and U, respectively). However, when both matrices are used to estimate factors, the factors for the X and Y matrices have the following relationship:

$$u = bt + \epsilon$$

where b is termed the inner relationship between u and t and is used to calculate subsequent factors if the intrinsic dimensionality of X is greater than one. Geometrically, this states that the vectors u and t are approximately equal except for their lengths.

The main advantage of PLS is that it incorporates more information in the model-building phase.

2.2 The Algorithm

The Partial Least-Squares algorithm, as described in (Geladi and Kowalski, 1986), can be decomposed in the following two steps:

1. given a data set for training, a regression model is built. This is the calibration or training step;

2. given an independent data set, called test set, predictions are made using the model that has just been built.

In the following subsections,we examine these two steps.

2.2.1 Model Training

The standard PLS algorithm, as described in figure 1, uses as the training set both the $n \times m$ matrix X, and the $n \times l$ matrix Y. Observe that X contains the n observations of m independent variables. On the other hand, Y contains the corresponding values for the dependent variables.

At each iteration, the following factors are calculated regarding the regression model between the scores for X and Y:

1. the weights w_i;

2. the regression vector coefficients b_i for the inner relation;

3. the loadings represented by p_i.

As already stated, there is a key difference between PLS and other regression methods (Geladi and Kowalski, 1986) such as PCR. Both methods construct a regression on principal components, however the model constructed by PLS also uses information from the dependent variable Y to bias the principal components. In fact, from lines 2 and 3 in figure 1, we get that the weighting factor w_i is an eigenvector of $X^\top YY^\top X$, providing a better quality for the prediction step.

2.2.2 Prediction Step

Given a trained model, obtained as described in the previous section, one can make predictions by using an independent data set X. Figure 2 shows the algorithm for this step.

Standard PLS

for $\mathtt{i} = \mathtt{1}$ to $\mathtt{m}$

$\quad \mathtt{v} \leftarrow$ eigenvector of $\mathtt{X}^\top \mathtt{Y} \mathtt{Y}^\top \mathtt{X}$ associated with the largest eigenvalue

$\quad \mathtt{w_i} \leftarrow \mathtt{v}/\|\mathtt{v}\|$

$\quad \mathtt{t_i} \leftarrow \mathtt{X} \mathtt{w_i}$

$\quad \mathtt{b_i^\top} \leftarrow \mathtt{t_i^\top} \mathtt{Y} / \mathtt{t_i^\top} \mathtt{t_i}$

$\quad \mathtt{p_i} \leftarrow \mathtt{X}^\top \mathtt{t_i}$

$\quad \mathtt{p_i} \leftarrow \mathtt{p_i}/\|\mathtt{p_i}\|$

$\quad \mathtt{X} \leftarrow \mathtt{X} - \mathtt{t_i} \mathtt{p_i^\top}$

$\quad \mathtt{Y} \leftarrow \mathtt{Y} - \mathtt{t_i} \mathtt{b_i^\top}$

end

Figure 1: Standard PLS algorithm.

It should be noticed that the number of desired factors for the prediction is indicated by the value of k.

A common procedure, when determining the optimal number of factors k to be used in the prediction, consists in calculating a statistic for the lack of model accuracy called PRESS (Geladi and Kowalski, 1986) (Prediction Residual Sum of Squares). This kind of method, called cross-validation (Beebe and Kowalski, 1987), uses an independent data set X with an already known variable Y. PRESS is calculated for each value of k, and the one that yields the minimum PRESS indicates the recommended number of factors to be chosen.

3 APLS: Approximate PLS Algorithm

As shown in the previous section, the eigenvector of the matrix $X^\top Y Y^\top X$ associated with the largest eigenvalue must be computed for each desired factor. As an alternative to the standard PLS algorithm, one could use neural network techniques such as Hebbian learning

```
PLS Prediction
for  i = 1 to k
        t_i ← Xw_i
        y ← y + t_i b_i^⊤
        X ← X − t_i p_i^⊤
end
```

Figure 2: Standard PLS algorithm for the prediction of y.

(Haykin, 1999), based on a learning rate and thus providing an iterative algorithm with a convergence criterion. APLS substitute this step by an approximate non-iterating one.

3.1 The Algorithm

Let G denote the matrix $X^\top YY^\top X$. For each factor, we must find the eigenvector w of G associated with the largest eigenvalue.

When Y has only one dependent variable ($l = 1$), then G has rank 1 and the corresponding eigenvector is simply $X^\top Y$. This eigenvector corresponds to the unique non-zero eigenvalue $(X^\top Y)^\top (X^\top Y)$. On the other hand, when $l \geq 2$ that is not true anymore. Nevertheless, one can decompose G as follows:

$$G = X^\top (Y_1 Y_1^\top + Y_2 Y_2^\top + \cdots + Y_l Y_l^\top) X$$

and then,

$$G = X^\top Y_1 Y_1^\top X + \cdots + X^\top Y_l Y_l^\top X \tag{1}$$

where Y_i $(1 \leq i \leq l)$ corresponds to the ith column of Y. For a simpler notation, we write (1) as

$$G = G_1 + G_2 + \cdots + G_l$$

where $G_i = X^\top Y_i Y_i^\top X$ for $i = 1, \ldots, l$.

As we can see, the eigenvector w of G associated with the largest eigenvalue is also an eigenvector of the sum of the l already known matrices G_i of rank 1. It is acceptable that the eigenvector w_i of matrix G_i with the largest eigenvalue λ_i will have a greater influence on w, and that eigenvectors having little correlation with w_i (towards orthogonality) will have little contribution to the sum described in (1). It follows that a simple approximation for w can be obtained through the following scheme:

1. For each $i = 1, \ldots, l$, calculate the eigenvector w_i and corresponding eigenvalue λ_i of G_i given by the following relations

$$w_i = X^\top Y_i$$

$$\lambda_i = w_i^\top w_i$$

2. Find the eigenvector $w_{(1)}$ corresponding to the largest eigenvalue $\lambda_{(1)}$, defined by

$$\lambda_{(1)} = \max_{1 \le i \le l} \{\lambda_i\}$$

3. Calculate w by

$$w = \sum_{i=1}^{l} \lambda_i w_i (w_i^\top w_{(1)}) \tag{2}$$

and apply normalization $w \leftarrow w/\|w\|$.

Equation (2) offers an explicit form for the computation of w, not requiring any kind of convergence criteria such as presented by Hebbian learning or the standard PLS algorithm.

It turns out that the APLS algorithm provides a competitive prediction quality and an improved running time, as shown in the following sections.

4 Running Time Analysis

Let X and Y be $(n \times m)$ and $(n \times l)$ matrices respectively. For both PLS and APLS, considering the code presented in Figure 1, we can describe each factor calculation by the two major steps that follows:

1. Compute the eigenvector w associated with the largest eigenvalue. This corresponds to the first two assignments in Figure 1;

2. Compute the regression vector coefficients b, loadings p and residuals matrices. This corresponds to the remaining assignments in Figure 1.

Hence, the time complexity of PLS and APLS differ only in the first major step just described.

For the standard PLS algorithm, we use the NIPALS algorithm presented in (Geladi and Kowalski, 1986), which is a power method for the largest eigenvector extraction (Golub and Loan, 1996). For each step towards convergence,$O(nml)$ time is required.

On the other hand, APLS has the same $O(nml)$ time complexity for its unique approximate eigenvector step.

We can conclude that in the case of NIPALS requiring only one iteration, the time complexity of both PLS and APLS are the same. However, this does not occur very often. Usually, NIPALS requires more than 10 iterations to converge at usual error levels. This makes the APLS approach faster since it requires just one iteration.

In fact, as shown in (Golub and Loan, 1996), the error obtained at the kth iteration of NIPALS for the eigenvector estimate $w^{(k)}$ is bound by

$$|\sin(\theta_k)| \leq \tan(\theta_0)\left|\frac{\lambda_{(2)}}{\lambda_{(1)}}\right|^k$$

where

$$cos(\theta_k) = \left|w_{(1)}^\top w^{(k)}\right|$$

and $\lambda_{(1)}$ and $\lambda_{(2)}$ are the largest and second largest eigenvalues. As we can see, depending of the rate $|\lambda_{(2)}|/|\lambda_{(1)}|$ the convergence rate can be very slow, making the APLS approach relatively faster.

5 Experimental results

We have tested the APLS algorithm on nine data sets: Wheat, Light gas oil, Combustible, Metal ions, Corn, Wet grass, Dry grass, Meat and Polymer test plant. For all of them we compared the PRESS obtained using PLS and APLS. For each case, we have chosen the corresponding number of factors for a simultaneous best prediction and model synthetis. Unless otherwise stated, all sets were mean centered based on the training samples. For a better understanding of the results, a short description of each data set follows.

5.1 Data Set descriptions

5.1.1 Wheat

The first one was taken from Kalivas (Kalivas, 1997). We used the data set containing the NIR spectra of 100 wheat samples along with specified protein and moisture content. Samples were measured using diffuse reflectance as $\log(1/R)$ from 1100 to 2500 nm in 2nm intervals. Of the 100 spectra, 70 were utilized for training (calibration) and the 30 remaining for testing (validation) the constructed model. Spectra were reduced to contain only 141 response by using every fifth response.

5.1.2 Light gas oil

As the second data set, we used the light gas oil data available at Dalhousie University (Wentzell et al., 1999). This set is for the calibration of light gas oil (and diesel) fuels for hydrocarbon content and consists of 115 samples from three subsets for which the spectra over 572 channels have been obtained. For the calibration and validation matrices we used the first 70 and remaining 44 samples respectively, along with the concentrations of the four components in each sample. Being an outlier, the last sample (115) was not used.

5.1.3 Combustible

As the third data set, we used a set of 30 combustible samples for which the NIR spectra over 3632 channels have been measured. Samples were reduced to contain only 363 measures by using every tenth response. 21 samples were utilized for calibrating (70% of the set) and the remaining 9 for validating. As the dependent variables, concentrations of three components were used for each sample.

5.1.4 Metal ions

The next data set, also available from Dalhousie University, was obtained through a carefully designed experiment involving three-component mixtures of metal ions (Co(II), Cr(III), Ni(II)), a system suggested from the work of Osten and Kowalski. Spectra were recorded over the range of 300-650 nm on a HP 8452 diode array spectrophotometer using a standard 1 cm quartz cuvette. Measurements were made at 2nm intervals with a 1s integration time. For training, 92 samples were used, and the remaining 38 for testing.

5.1.5 Corn

As the fifth data set, the NIR spectra of corn samples was used. This data set consists of 80 samples of corn measured on 3 different NIR spectrometers. The wavelength range is 1100-2498nm at 2 nm intervals (700 channels). As the dependent variables, the moisture, oil, protein and starch values for each of the samples were used.

Table 1: Data Sets used for testing

Data Set	N. Samples	Indep. Variables	Dep. Variables
Wheat	100	200	2
Light gas oil	114	572	4
Combustible	30	363	3
Metal ions	130	176	3
Corn	80	700	4
Wet grass	282	1050	3
Dry grass	141	1050	3
Meat	215	100	3
Polymer test plant	61	10	4

5.1.6 Wet and Dry grass

The next two data set were taken from shootout at the IDRC98, a conference held at the Chambersburg, Pennsylavania. The data sets consisted of scans and chemistry obtained from fescue grass grown in a soil medium where the moisture was wicked from liquid tanks containing four levels of fertilization (0, 50, 250 and 500 ppm of nitrogen). The chemistry was determined on a LECO CNS-2000 Carbon, Nitrogen and Sulphur Analyzer. This instrument is a non-dispersive, infrared, microcomputer based instrument designed to measure the carbon, nitrogen and sulphur content in a wide variety of organic compounds. Carbon and sulphur are measured by infrared radiation detection; nitrogen is determined by conductivity. The first data set corresponds to wet green samples scanned directly after harvesting, while the second to dry samples grounded through a 1 mm screen in a Wiley mill. For each training we used 196 samples from the wet grass set and 96 from the powdered set. Regarding testing, 86 and 45 samples were used respectively.

5.1.7 Meat

The Tecator data set was used next, where the task was to predict the fat content of a meat sample on the basis of its near infrared absorbance spectrum. The data were recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850 - 1050 nm by the Near Infrared Transmission (NIT) principle. For each meat sample the data consists of a 100 channel spectrum of absorbances and the contents of moisture (water), fat and protein. The absorbance is -log10 of the transmittance measured by the spectrometer. The three contents, measured in percent, are determined by analytic chemistry. As sugested by the author, the first 172 samples were used for training while the following 43 for testing purposes.

5.1.8 Polymer

The last data set employed was taken from a polymer test plant. It has been disguised by linearly transforming the data, so that all variables lie between 0.1 and 0.9. Due to the proprietary nature of the data, the ten independent variables are measurements of controlled variables in a polymer processing plant (e.g. temperatures, feed rates ...) and the four dependent variables are measures of the output of that plant. Of the 61 samples available, 42 were used for training and 19 for testing.

5.2 Experiment results

When comparing the PRESS of the models produced by standard PLS and APLS we obtain similar results. To illustrate the overall behaviour of the APLS performance, the PRESS values are ploted for some data sets. Figures 3 and 4 show the modeling quality for the first data set. Despite of having a better modeling with the standard PLS, APLS still shows a competitive result. For some data sets, the PRESS curve of both models shows a very similar shape, as shown in figure 5 for the Dalhousie metal experiment. In some cases, the APLS shows not only a better PRESS but also needs fewer factors. This is the case of the wet grass data set (figures 6 and 7).

As we can see, the APLS algorithm is very competitive when compared to the standard PLS. In Table 2, we summarize our empirical findings. First, the minimum PRESS for both methods are nearly the same. The observed relative difference runs in the range of -xx% to yy%. Second, the chosen number of factors are similar. It is also important to note that during the experiments the APLS modeling shows at least a 45% faster runing time. In fact, the NIPALS convergence in the standard PLS required an average of 40 iterations.

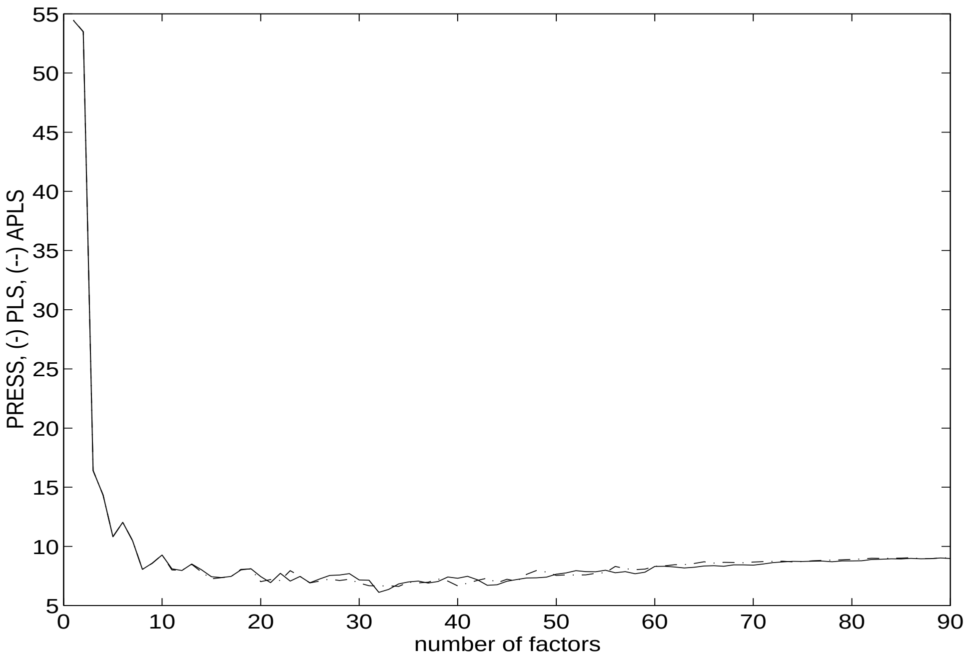

Figure 3: PRESS values for PLS and APLS, Wheat data set.

Table 2: PRESS and Number of Chosen Factors for PLS and APLS

Data Set	Number of factors		PRESS		PLS
	PLS	APLS	PLS	APLS	Iteration
Wheat	32	34	6.12	6.62	0
Light gas oil	8	8	61.77	61.60	0
Combustible	5	5	184.79	171.09	0
Metal ions	8	8	1.5096×10^{-4}	1.5007×10^{-4}	47.57
Corn	11	10	4.146	4.159	60.40
Wet grass	28	22	203.62	199.19	45.92
Dry grass	14	12	38.97	35.76	57.34
Meat	33	32	259.55	273.65	34.65
Polymer test plant	5	5	0.4288	0.4257	15.10

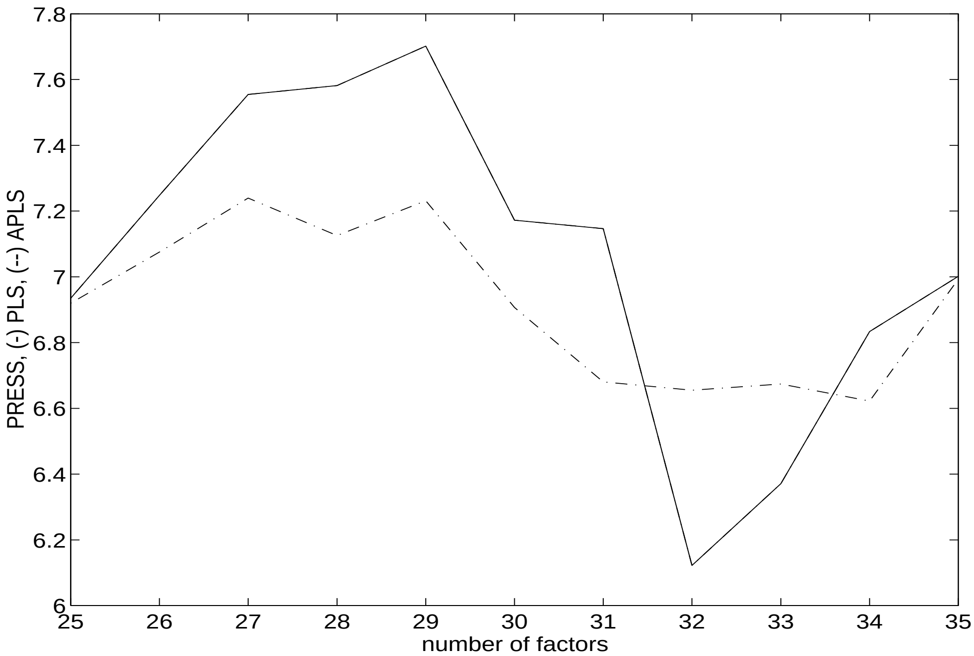

Figure 4: PRESS values of PLS and APLS for Wheat data set, around the selected factors.

6 Conclusions

We introduce APLS, an approximate algorithm for the Partial Least-Squares problem. Instead of computing the eigenvector of the matrix $X^\top YY^\top X$ in the case of two or more dependent variables, APLS uses an approximate scheme based on a special decomposition of the given matrix.

We have made experiments with nine chemometric data sets. It turns out that the main characteristics of APLS are:

1. minimum PRESS found is similar to the one found by PLS;

2. similar number of factors chosen when compared to standard PLS;

3. improved running time, since we obtained a 40% better time in our experiments.

As we can see, APLS can be considered an alternative approach to PLS. Furthermore, it is easy to see that the parallelization of APLS is straightforward (Milidiú and Rentería, 2000), making it also suitable for processing large data sets at low cost computer architectures.

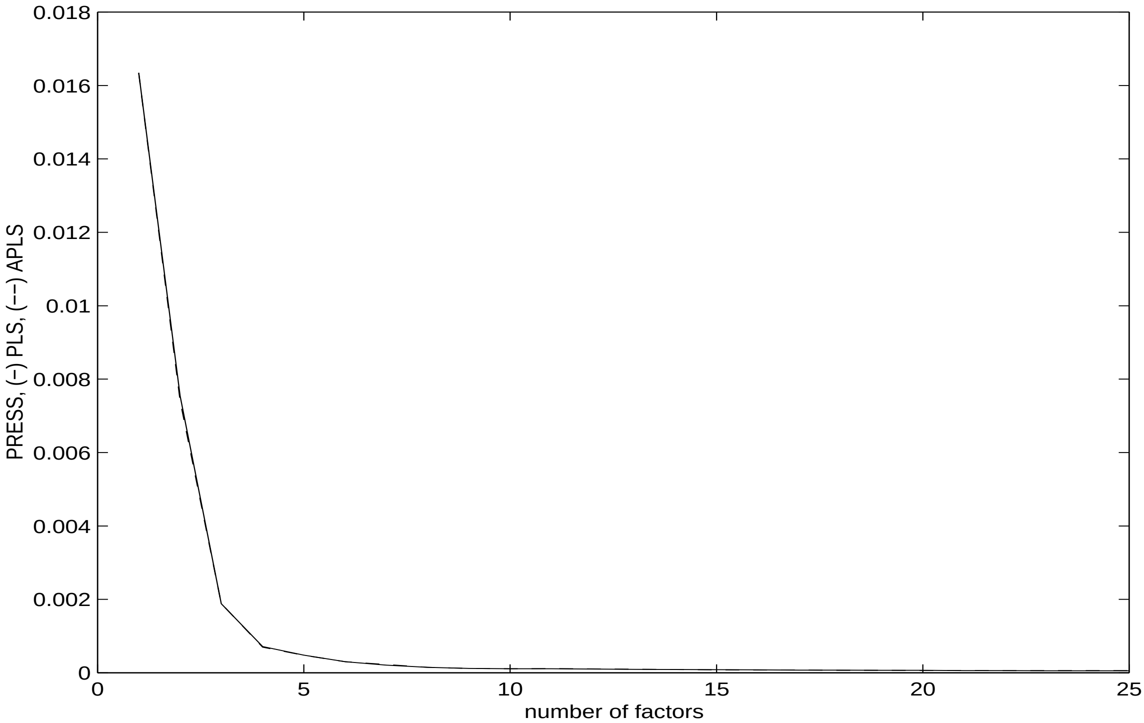

Figure 5: PRESS values of PLS and APLS, Metal ions data set.

References

Beebe, K. R., Kowalski, B. R., September 1987. An introduction to multivariate calibration and analysis. Analytical Chemistry 59 (17), 1007–1017.

Geladi, P., Kowalski, B. R., 1986. Partial least squares regression: A tutorial. Analytica Chimica Acta 185, 1–17.

Golub, G. H., Loan, C. F. V., 1996. Matrix Computations. John Hopkins University Press.

Haaland, D. M., Thomas, E. V., June 1988a. Partial least-squares methods for spectral analysis. 1. relation to other quantitative calibration methods and the extraction of qualitative information. Analytical Chemistry 60 (11), 1193–1202.

Haaland, D. M., Thomas, E. V., June 1988b. Partial least-squares methods for spectral analysis. 2. application to simulated and glass spectral data. Analytical Chemistry 60 (11), 1202–1208.

Haykin, S., 1999. Neural Networks, a Comprehensive Foundation. Prentice-Hall.

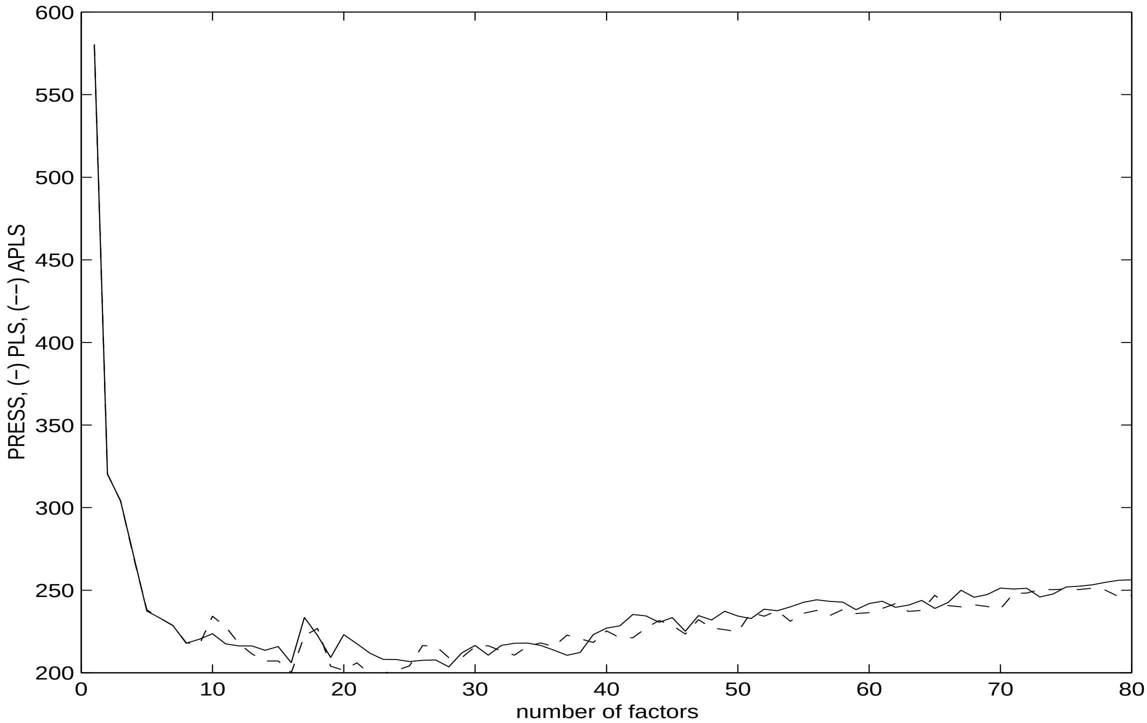

Figure 6: PRESS values of PLS and APLS, Wet grass data set.

Kalivas, J. H., 1997. Two data sets of near infrared spectra. Chemometrics and Intelligent Laboratory Systems 37, 255–259.

Lebart, L., Morineau, A., Piron, M., 1997. Statistique Exploratoire Multidimensionnelle. Dunod, Paris.

Milidiú, R., Machado, R., Rentería, R., September 1998. Time series forecasting throught wavelets and a mixture of experts models. In: NEURAP'98. Marseilles, France.

Milidiú, R., Machado, R., Rentería, R., 1999. Time series forecasting throught wavelets and a mixture of experts models. Neurocomputing 28, 145–156.

Milidiú, R. L., Rentería, R., October 2000. Ppls: An efficent parallel algorithm for partial least-squares. In: 12th Symposium on Computer Architecture and High Performance Computing. São Pedro, SP, Brazil.

Morineau, A., Tenenhaus, M. (Eds.), October 1999. Les Méthodes PLS, Symposium International PLS'99. Cisia-Ceresta.

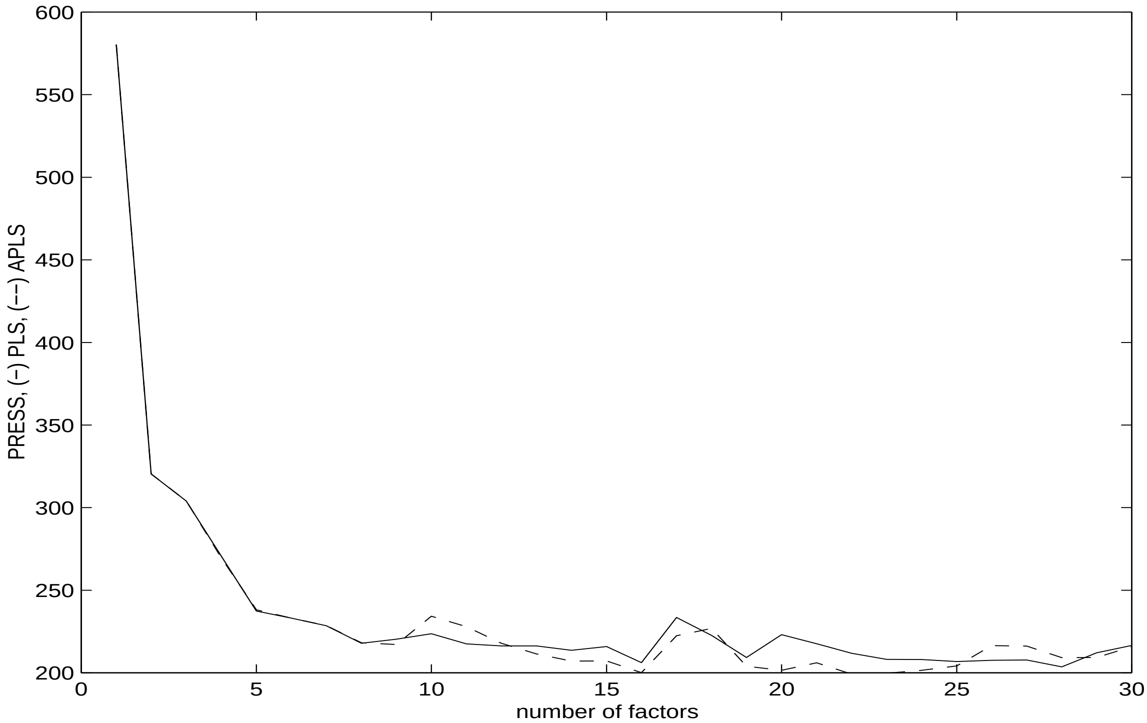

Figure 7: PRESS values of PLS and APLS for data set 6, around the selected factors.

Wentzell, P., Andrews, D., Walsh, J., Cooley, J., Spencer, P., 1999. Estimation of hydrocarbon types in light gas oils and diesel fuels by ultraviolet absorption spectroscopy and multivariate calibration. Can. J. Chem. 77.