MKPLS: A Multi-kernel based Partial Least-Squares regression

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

PUC-RioInf.MCC07/03 February, 2003

Abstract: MKPLS, a non-linear version of the Partial Least-Squares regression is presented. The non-linearity is introduced in the classical algorithm through the use of multiple kernel functions, thus providing an straightforward non-linear adaptation. MKPLS provides a multi-kernel based version for the PLS algorithm with a competitive modeling error. Experimental results show that the use of different kernels for the regression model enhances the predictive power when compared to a PLS regression based on only one function kernel.

Keywords: PLS, MKPLS, Kernel Functions, Multi-Kernel, Factor Analysis, chemometric

Resumo: MKPLS, uma versão não linear para a regressão por mínimos quadrados parciais é apresentado. A não linearidade é introduzida no algoritmo clássico com o uso de múltiplas funções núcleo, fornecendo desta forma uma eficiente adaptação não linear. MKPLS fornece uma versão baseada em múltiplos núcleos para o algoritmo PLS possuindo um erro de modelagem competitivo. Resultados experimentais indicam que o uso de diferentes núcleos para o modelo de regressão aumenta o poder de predição quando comparado com a regressão PLS baseada em apenas uma função núcleo.

Palavras-Chave: PLS, MKPLS, Funções de Núcleo, Multi-Kernel, Análise Fatorial, quimiometria

1 Introduction

Partial Least Squares regression (Wold, 1966; Wold et al., 1983) has been widely used as a chemometric tool for Near-Infrared spectral analysis (Geladi and Kowalski, 1986; Haaland and Thomas, 1988a,b; Beebe and Kowalski, 1987) for the robustness of the generated model when the number of variables is large when compared to the number of samples. This led to its application 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 MKPLS, a multi-kernel based algorithm for Partial Least-Squares regression. A kernel PLS2 algorithm based on only one kernel has already been proposed in Rosipal and Trejo (2001) showing that the use of non-linear modeling can improve predictive power. With MKPLS we show that using different kernels at the training phase provides a better adaptation to the input data resulting in not only a more compact model but also better prediction quality.

In order to measure the performance of MKPLS, we report some experiments on data sets mainly related to NIR spectra analysis, such as wheat data for chemometrics (Kalivas, 1997) or combustible (Wentzell et al., 1999). For the kernel based regression, LPLS is used, a kernel PLS formulation for the case of only one dependent variable (PLS1) that shows better numerical stability when compared to the PLS kernel algorithm in Rosipal and Trejo (2001).

In section 2, our multi-kernel approach is described. In section 3, we present LPLS, the kernel based PLS1 algorithm. In section 4, the empirical results obtained with the selected data set are shown. Finally, in section 5, we summarize our findings.

2 MKPLS: Multi-Kernel PLS regression algorithm

The main motivation for MKPLS was the PRESS curve obtained with one kernel PLS when compared to the standard linear PLS. For example, if both curves are plotted (figure 1) for the *Meat* data set described in section 4.1.5, we see that one kernel PLS outperforms PLS if one uses more than 13 factors. However the performance of one kernel PLS is really poor for the first factors. The polynomial kernel defined as $K_{i,j} = (x_i \cdot x_j + 1)^2$ can barely model the predicted variable y for the first 10 factors. It would be interesting to have one regression model that would be as sharp as PLS on the first factors and as one kernel PLS on the remaining ones. MKPLS generalizes the one kernel PLS by using a kernel matrix K_1 for the first f_1 factors and then switching to a different kernel K_2 for the remaining

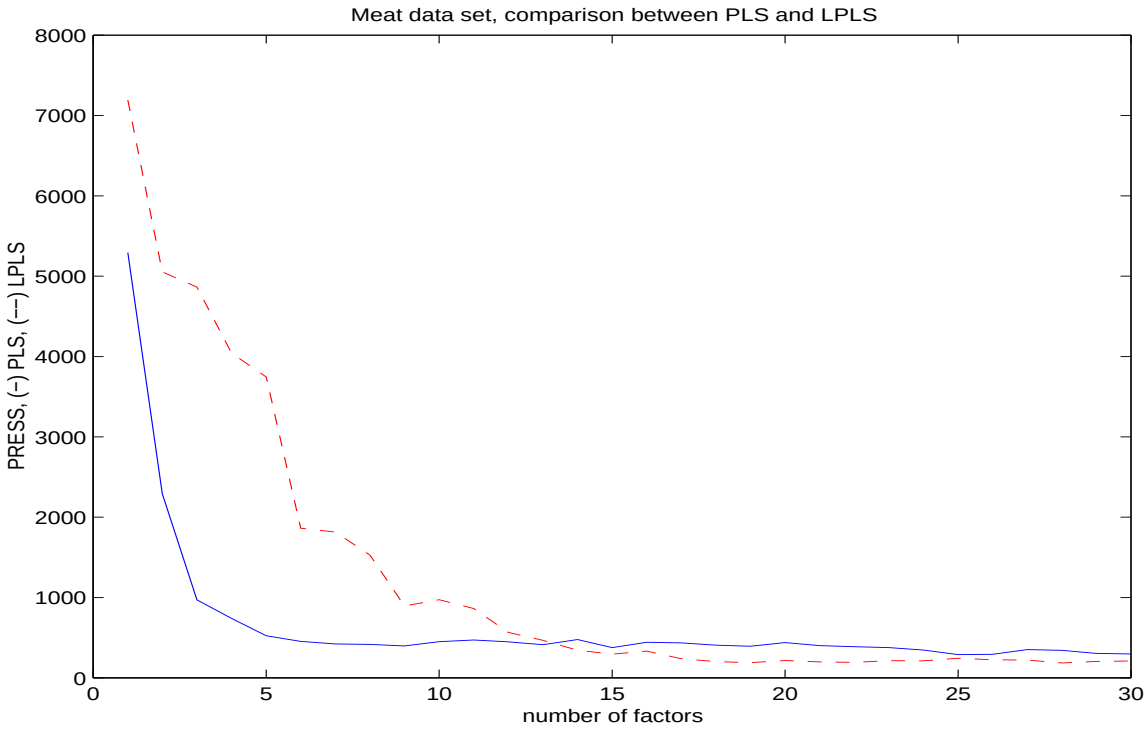

Figure 1: PRESS values for PLS and LPLS, Meat data set.

Multi-Kernel PLS regression approach
Apply one kernel PLS regression to first kernel $\mathtt{K_1}$
Deflate second kernel $\mathtt{K_2}$ using model obtained in step 1
Apply one kernel PLS regression to the deflated kernel $\mathtt{K_2}$

Figure 2: MKPLS main steps.

factors, as indicated by the high level algorithm on figure 2. For the correctness of this operation it is important that the mapping done with K_2 includes the mapping done by K_1. Since the addition of two kernels is still a kernel this can be simply done by defining $K_2 = K_1 + K$ where K denotes the kernel for the additional non-linearity, Gaussian or polynomial for instance.

This is a requirement in MKPLS since the switching of the kernels is done by deflating the factors t_i found with K_1. This would make no sense if the mapping of K_1 was not also done with K_2.

2.1 Training step

Given two kernel functions with the corresponding kernel matrices K_1 and K_2 for some training data set, the MKPLS model can be constructed through the following procedure:

1. obtain the first f_1 factors $\{t_i, b_i\}$ applying the one kernel PLS algorithm using the first kernel matrix K_1;

2. deflate the second kernel matrix K_2 and the dependent variable Y by applying the deflating algorithm described in figure 3;

3. apply again the one kernel PLS algorithm to the deflated kernel obtaining the remaining f_2 model factors.

At the end of this procedure, the set of $f_1 + f_2$ factors corresponding to the non-linear multi-kernel model will be available.

	$\mathtt{K_2}$ and $\mathtt{Y}$ deflation for the training step	
1	for	$\mathtt{i} = 1$ to $\mathtt{f_1}$
2		$\mathtt{g'_i} \leftarrow \mathtt{K_2 t_i}$
3		$\mathtt{a_i} \leftarrow \mathtt{t_i^\top t_i}$
4		$\mathtt{U} \leftarrow \mathtt{g'_i t_i^\top / a_i}$
5		$\mathtt{K_2} \leftarrow \mathtt{K_2} - \mathtt{U} - \mathtt{U}^\top + (\mathtt{t_i^\top g_i})\mathtt{t_i t_i^\top}/\mathtt{a_i^2}$
6		$\mathtt{Y} \leftarrow \mathtt{Y} - \mathtt{t_i b_i^\top}$
7	end	

Figure 3: MKPLS training deflating algorithm.

2.2 Prediction step

Since the one kernel PLS prediction algorithm uses a specific kernel matrix related to the test data set, it will be also necessary to switch the kernel matrices K'_1 and K'_2 during the prediction phase.

1. apply the same prediction algorithm starting with K'_1. However the set of f_1 score t'_i should be retained for deflation along with the predicted y;

2. deflate K'_2 using the algorithm in figure 4. Note the use of g'_i and a_i obtained during training deflation;

3. apply the prediction algorithm starting with the deflated K'_2 and the predicted y obtained in step 1.

		$\mathtt{K}'_2$ deflation for the prediction step
1	for	$\mathtt{i} = 1$ to $\mathtt{f}_1$
		// Residual calculation
2		$\mathtt{U1} \leftarrow \mathtt{K}'_2\mathtt{t}_\mathtt{i}\mathtt{t}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i}$
3		$\mathtt{U2} \leftarrow \mathtt{t}'_\mathtt{i}\mathtt{g}_\mathtt{i}'^\top/\mathtt{a}_\mathtt{i}$
4		$\mathtt{K}'_2 \leftarrow \mathtt{K}'_2 - \mathtt{K}'_2\mathtt{t}_\mathtt{i}\mathtt{t}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i} - \mathtt{t}'\mathtt{g}_\mathtt{i}'^\top/\mathtt{a}_\mathtt{i} + (\mathtt{t}_\mathtt{i}^\top\mathtt{g}'_\mathtt{i})\mathtt{t}'_\mathtt{i}\mathtt{t}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i}^2$
5	end	

Figure 4: MKPLS prediction deflating algorithm.

The MKPLS training and prediction procedures are very close to the one kernel version of PLS. The main difference being the deflation algorithms for the kernel function switching.

3 LPLS regression algorithm

In this section we describe LPLS, the kernel based PLS1 used with MKPLS for the experiments described in section 4. We first introduce the classical linear PLS1 algorithm and then the kernel based approach is presented.

3.1 PLS algorithm

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. The Partial Least-Squares algorithm, as described in (Geladi and Kowalski, 1986), can be decomposed in a training step and a prediction step, as follows:

1. given a data set for training, a regression model is built;

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

3.1.1 Training step

The PLS1 algorithm, as described in figure 5, uses as the training set both the $n \times m$ matrix X, and the $n \times 1$ 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 variable. At each iteration, the following items are calculated:

1. the weights w_i;
2. the factors t_i;
3. the regression coefficients b_i for the inner relation between t_i and Y;
4. the loadings represented by p_i.

PLS1 regression algorithm		
1	for	$\mathtt{i} = \mathtt{1}$ to $\mathtt{m}$
2		$\mathtt{v} \leftarrow \mathtt{X}^\top \mathtt{Y}$
3		$\mathtt{w_i} \leftarrow \mathtt{v}/\|\mathtt{v}\|$
4		$\mathtt{t_i} \leftarrow \mathtt{X w_i}$
5		$\mathtt{b_i} \leftarrow \mathtt{Y}^\top \mathtt{t_i}/\mathtt{t_i}^\top \mathtt{t_i}$
6		$\mathtt{p_i} \leftarrow \mathtt{X}^\top \mathtt{t_i}/\mathtt{t_i}^\top \mathtt{t_i}$
7		$\mathtt{X} \leftarrow \mathtt{X} - \mathtt{t_i p_i}^\top$
8		$\mathtt{Y} \leftarrow \mathtt{Y} - \mathtt{t_i b_i}$
9	end	

Figure 5: PLS1 algorithm.

There is a key difference between PLS and other regression methods (Geladi and Kowalski, 1986) such as PCR (Principal Component Regression). 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 5, we get that the weighting factor w_i is an eigenvector of $X^\top Y Y^\top X$, providing a better quality for the prediction step.

3.1.2 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 6 shows the algorithm for this step. It should be noticed that the number of desired factors for the prediction is indicated by the value of k.

PLS1 Prediction		
1	for	$\mathtt{i} = \mathtt{1}$ to $\mathtt{k}$
2		$\mathtt{t_i} \leftarrow \mathtt{Xw_i}$
3		$\mathtt{y} \leftarrow \mathtt{y} + \mathtt{b_i t_i}$
4		$\mathtt{X} \leftarrow \mathtt{X} - \mathtt{t_i p_i}^{\top}$
5	end	

Figure 6: PLS1 algorithm for the prediction of y.

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.

3.2 Lifted PLS

LPLS is a PLS1 based regression algorithm. The main idea behind the use of kernel functions with PLS is to embed the original data from the input space into a higher dimensional feature space and then apply the desired linear algorithm. This is why we call it *lifted*. This becomes computational feasible since the data is not explicitly represented in the feature space but instead the dot product in this space is expressed in terms of kernel functions in the input space. The difficulty in devising a kernel based algorithm is to formulate it only in terms of dot products, not explicitly using the additional variables themselves. The strength of the technique is that the learning algorithm is still the same and through the choice of an appropriate kernel function the characteristics of a given data set can be learned. In that way, if a kernel function that maps the data to the same input space is used, the algorithm will behave exactly as the original linear one.

3.2.1 Training step

As already explained, the algorithm on figure 5 must be reformulated so that variables in the form of X are not explicit stated but rather the dot products between samples through the kernel matrix $XX^\top$. This has been done for PLS2 by Rosipal and Trejo (2001) to introduce non-linearity in the modeling. In fact the kernel matrix is replaced by the kernel Gram matrix $K = \mathbf{\Phi}\mathbf{\Phi}^\top$, $\mathbf{\Phi}$ being the matrix of mapped input training data where the input samples x_i are transformed into a feature space $\mathcal{F}$ through the mapping $\Phi : x_i \in \mathtt{R}^\mathtt{m} \mapsto \Phi(x_i) \in \mathcal{F}$. In that way $\mathbf{\Phi}$ is an $(n \times m')$ matrix where row i is the vector $\Phi(x_i)$. Depending on the non-linear mapping $\Phi()$ used, the feature space can be high-dimensional, even infinite as in the case of the Gaussian kernel.

The algorithm for LPLS is shown on figure 7. Its main characteristics are:

1. the input for the independent variables is in the form of the kernel matrix K calculated using the appropriated kernel function;

2. h_i in line 3 is the squared norm of the eigenvector w_i which can't be explicitly expressed since we are in feature space $\mathcal{F}$;

3. the only required information for the model are the b_i, t_i and h_i. The others, a_i, g_i and Yr_i are retained for computational performance purposes during the prediction step;

4. in theory, the number of calculated factors m' in line 1 could be as high as the dimension of the feature space. In practice this should be limited to a number not higher than m.

3.2.2 Prediction Step

Figure 8 shows the prediction algorithm for LPLS. Note the presence of K' which is $\mathbf{\Phi}'\mathbf{\Phi}^\top$ where $\mathbf{\Phi}'$ is the matrix of mapped input testing data where the input test samples are transformed into the feature space $\mathcal{F}$ through the same mapping $\Phi()$ used for the training data.

LPLS regression algorithm		
1	for	$\mathtt{i} = 1$ to $\mathtt{m}'$
2		$\mathtt{Yr}_\mathtt{i} \leftarrow \mathtt{Y}$
3		$\mathtt{h}_\mathtt{i} \leftarrow \mathtt{Y}^\top \mathtt{KY}$
4		$\mathtt{t}_\mathtt{i} \leftarrow \mathtt{KY}/\sqrt{\mathtt{h}_\mathtt{i}}$
5		$\mathtt{a}_\mathtt{i} \leftarrow \mathtt{t}_\mathtt{i}^\top \mathtt{t}_\mathtt{i}$
6		$\mathtt{b}_\mathtt{i} \leftarrow \mathtt{Y}^\top \mathtt{t}_\mathtt{i}/\mathtt{a}_\mathtt{i}$
		// Residual calculation
8		$\mathtt{g}_\mathtt{i} \leftarrow \mathtt{Kt}_\mathtt{i}$
7		$\mathtt{U} \leftarrow \mathtt{g}_\mathtt{i}\mathtt{t}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i}$
9		$\mathtt{K} \leftarrow \mathtt{K} - \mathtt{U} - \mathtt{U}^\top + (\mathtt{t}_\mathtt{i}^\top \mathtt{g}_\mathtt{i})\mathtt{t}_\mathtt{i}\mathtt{t}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i}^2$
10		$\mathtt{Y} \leftarrow \mathtt{Y} - \mathtt{t}_\mathtt{i}\mathtt{b}_\mathtt{i}$
11	end	

Figure 7: LPLS algorithm.

LPLS Prediction		
1	for	$\mathtt{i} = 1$ to $\mathtt{k}$
2		$\mathtt{t}' \leftarrow \mathtt{K}'\mathtt{Yr}_\mathtt{i}/\sqrt{\mathtt{h}_\mathtt{i}}$
3		$\mathtt{y} \leftarrow \mathtt{y} + \mathtt{b}_\mathtt{i}\mathtt{t}'$
		// Residual calculation
4		$\mathtt{K}' \leftarrow \mathtt{K}' - \mathtt{K}'\mathtt{t}_\mathtt{i}\mathtt{t}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i} - \mathtt{t}'\mathtt{g}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i} + (\mathtt{t}_\mathtt{i}^\top \mathtt{g}_\mathtt{i})\mathtt{t}'\mathtt{t}_\mathtt{i}^\top/\mathtt{a}_\mathtt{i}^2$
5	end	

Figure 8: LPLS algorithm for the prediction of y.

4 Experimental results

4.1 Data Set descriptions

4.1.1 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.

4.1.2 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.

4.1.3 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.

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
Corn	80	700	4
Meat	215	100	3

4.1.4 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.

4.1.5 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 suggested by the author, the first 172 samples were used for training while the following 43 for testing purposes.

4.2 Experiment results

To compare the PRESS of the model produced by MKPLS with PLS and LPLS, two key characteristics are observed:

1. model complexity;
2. prediction quality.

The number of required factors to achieve a sufficiently small prediction error is our modeling complexity measure. This is obtained by comparing the PRESS curves for either the first 10 or 15 factors. The minimum PRESS value is our prediction quality measure.
For each region just described the minimum of each curve is compared. Also the percentage of times that MKPLS performed equally and better is calculated since re-sampling is done 20 times. For each data set, the following parameters are used:

1. the first kernel function resulting in matrix K_1;
2. the number f_1 of factors calculated with K_1;

3. the second kernel matrix K_2 used along with its parameters.

For all data set the identity kernel yielding the K_1 matrix given by $K_1 = XX^\top$ was used. Polynomial or Gaussian kernels were used for K_2 for all experiments. To illustrate the overall behavior of the MKPLS performance, the PRESS values of the three models are plotted for some data sets. As we can see in figures 9 and 10 the MKPLS modeling benefits

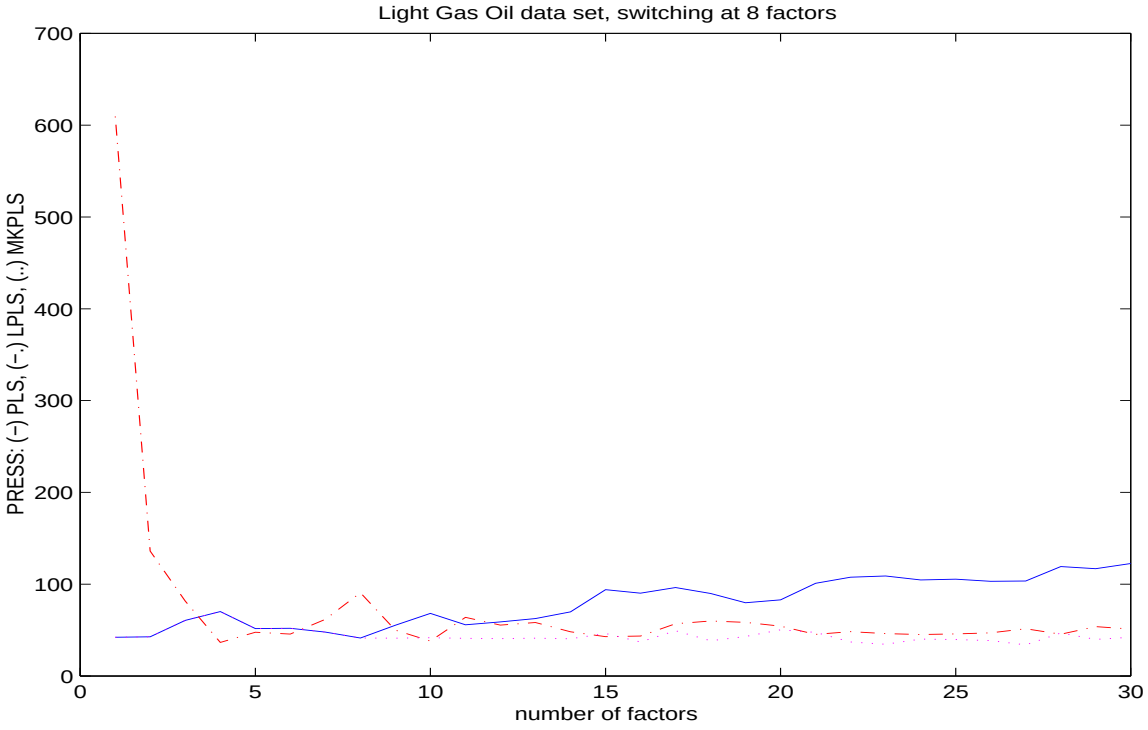

Figure 9: PRESS values of PLS, LPLS and MKPLS for Meat data set.

from both PLS and LPLS modeling. The poor performance of the non-linear model for the first factors is eliminated and the good predictive quality at higher factors is maintained. Table 2 shows the results for all data sets regarding the first 10 factors, whereas table 3 shows the performance of MKPLS over the two other models considering up to 30 factors. In both tables *MKPLS over PLS* means the minimum PRESS gain obtained with MKPLS when compared to PLS for the selected factors, or:

$$100 \cdot \left(1 - \frac{min(PRESS_{MKPLS})}{min(PRESS_{PLS})}\right)$$

The same applies to *MKPLS over LPLS*.

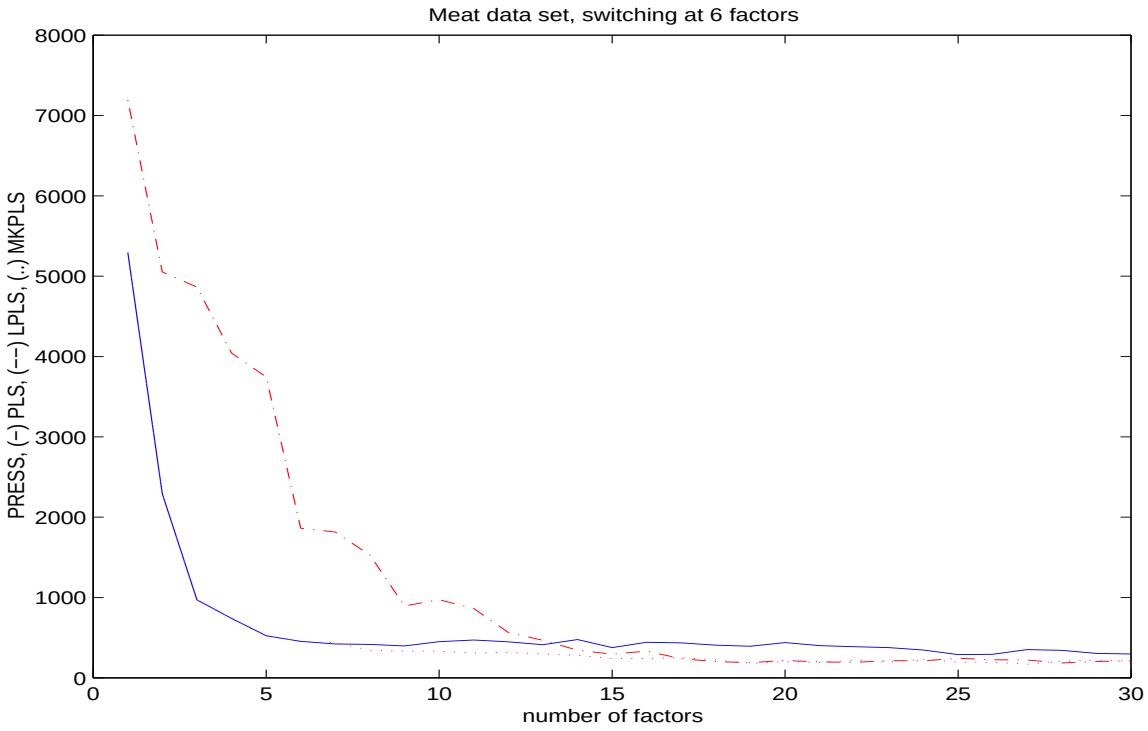

Figure 10: PRESS values of PLS, LPLS and MKPLS for Meat data set.

Table 2: MKPLS PRESS comparison between PLS and LPLS for first 10 factors.

Data Set	MKPLS over PLS				MKPLS over LPLS			
	mean	std.dev.	% draw	% win	mean	std.dev.	% draw	% win
Wheat	5.17	4.31	20.0	80.0	33.79	21.73	0.0	90.0
Meat	7.87	11.93	0.0	80.0	53.10	8.52	0.0	100.0
Combustible	11.40	12.32	25.0	70.0	66.31	23.75	0.0	100.0
Corn	0.24	0.87	86.7	13.3	27.18	13.34	0.0	100.0
Light gas oil	2.13	3.52	30.0	70.0	22.08	20.63	0.0	86.7

5 Conclusions

We introduce MKPLS, a multi-kernel based algorithm for Partial Least-Squares regression. Instead of using only one kernel, many can be used during the training and prediction steps.

Table 3: MKPLS PRESS comparison between PLS and LPLS for all factors

Data Set	MKPLS over PLS				MKPLS over LPLS			
	mean	std.dev.	% draw	% win	mean	std.dev.	% draw	% win
Wheat	20.15	13.88	5.0	90.0	12.56	18.19	0.0	85.0
Meat	48.06	16.91	0.0	100.0	-8.02	10.27	0.0	20.0
Combustible	21.52	20.12	25.0	75.0	66.28	24.76	0.0	100.0
Corn	1.89	2.72	26.7	73.3	18.12	17.77	0.0	86.7
Light Gas Oil	8.79	8.82	0.0	100.0	22.92	18.43	0.0	90.0

We have made experiments with 5 chemometric data sets using the identity kernel (resulting into the standard linear PLS algorithm) as the first one and a polynomial or Gaussian when appropriate, for the second. It turns out that the main characteristics of MKPLS are:

1. more compact model;
2. same learning rate as PLS for first factors;
3. competitive prediction quality when compared to LPLS;
4. at least the same performance as other models.

As we can see, MKPLS can be considered an alternative approach when using kernel based PLS regression. Furthermore, the same approach of MKPLS for switching kernels could be used to other kernel based regression schemes Rosipal et al. (2000); Rosipal and Trejo (2001).

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.

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.

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

Milidiú, R., Machado, R., Rentería, R., September 1998. Time series forecasting throught wavelets and a mixture of expert 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.

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

Rosipal, R., Trejo, L., Cichocki, A., 2000. Kernel principal component regression with em approach to nonlinear principal components extraction. Tech. rep., University of Paisley, School of Information and Communication Technologies.

Rosipal, R., Trejo, L. J., december 2001. Kernel partial least squares regression in reproducing kernel hilbert space. Journal of Machine Learning 2, 97–123.

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. Canadian Journal of Chemistry 77, 391–400.

Wold, H., 1966. Estimation of principal components and related models by iterative least squares. In: Krishnaiah, P. (Ed.), Multivariate analysis II. Academic Press, New York, pp. 391–420.

Wold, S., Albano, C., III, W. D., Esbensen, K., Hellberg, S., Johansson, E., Sjöström, H., 1983. Pattern recognition: finding and using regularities in multivariate data. In: Martens, J. (Ed.), Proc. IUFOST Conf. Food Research and Data Analysis. Applied Science Publications, London.