
Smooth PLS regression for spectral data *

Authors: ATHANASIOS KONDYLLIS
– Philip Morris International R&D, Philip Morris Products S.A.,
Quai Jeanrenaud 5, 2000, Neuchâtel, Switzerland
(athanasios.kondylis@pmi.com)

Received: Month 0000 Revised: Month 0000 Accepted: Month 0000

Abstract:

- Partial least squares (PLS) regression reduces the regression problem from a large- p number of interrelated predictors to a small- m number of extracted factors. These use information for predicting the response making PLS regression models extremely good for prediction purposes. The PLS regression coefficient vector is determined by the PLS factor loadings which drive the dimension reduction process; it should therefore be smooth, especially when the factor subspace dimension is small. We explore smooth alternatives for PLS regression revisiting a topic that triggered the research interest over the last two decades. We use for this the discrete wavelet transform focusing on PLS regression applications in near infra-red spectroscopy.

Key-Words:

- *PLS regression, Krylov subspaces, discrete wavelet transform, spectroscopy.*

AMS Subject Classification:

- 62J07

*The opinions expressed in this text are those of the author and do not necessarily reflect the views of Philip Morris International.

1. INTRODUCTION

Spectral data are characterized by a large number of interrelated measurements, intensities and absorptions, which are regularly recorded across a range of wavelengths. They are recorded by means of modern instruments and are often used as predictors in regression problems. In near infra-red (NIR) spectroscopy, in the food industry, for instance, samples of meat are analyzed for their fat content, and their NIR spectra are then used to predict fat concentration. Similar applications may be found in agriculture for the determination of properties of grains, in oil industry, in the analysis of pharmaceuticals, etc.

Using the spectral measurements as predictors in a regression problem limits traditional regression methods and implies the use of high-dimensional regression techniques. Partial least squares (PLS) regression has been for a long time implemented to deal with such regression problems, see [1]. PLS methods are based on reducing the dimension of the regression problem to a small- m number of factors rather than a large- p number of variables. This is achieved using information on the response variable, making PLS regression models excellent for prediction purposes.

More than twenty years have passed since the first smooth PLS regression has been presented in [2]. The authors have been motivated by non-parametric regression techniques in [3], and established the link between PLS regression and functional data analysis. This link resulted in numerous publications on PLS regression for functional data; see [4, 5, 6, 7, 8, 9]. The increasing interest in using functional data techniques for spectral applications stems from the fact that spectral data are indeed functional. NIR spectra, for example, are discrete instances of the chemical spectrum of a sample on a range of different wavelengths. This is illustrated in Figure 1 for 60 gasoline samples for which their spectral measurements are recorded at every two nanometers (nm) from 900 to 1700 nm. They are discrete values of continuous functions which are also smooth. Following [2] the extracted factor loadings should resemble to the spectra, and therefore should exhibit some degree of smoothness; the same holds for the regression solution. The gasoline samples data together with other two spectral data sets will be used in the examples that follow.

We revisit smooth PLS regression after a short overview on PLS regression given in Section 2. Two smooth PLS regression using wavelets are presented in Section 3 and Section 4. Their theoretical properties are investigated in Section 5; proofs are given in the Appendix. In Section 6 three well-known NIR data sets are revisited in order to illustrate smooth PLS regression. Focus is mainly given on NIR applications. Nevertheless, the presented smooth PLS regression alternative naturally applies to other spectral data, as well. Conclusions are given in Section 7.

Throughout the paper bold face lower and upper case letters are used for

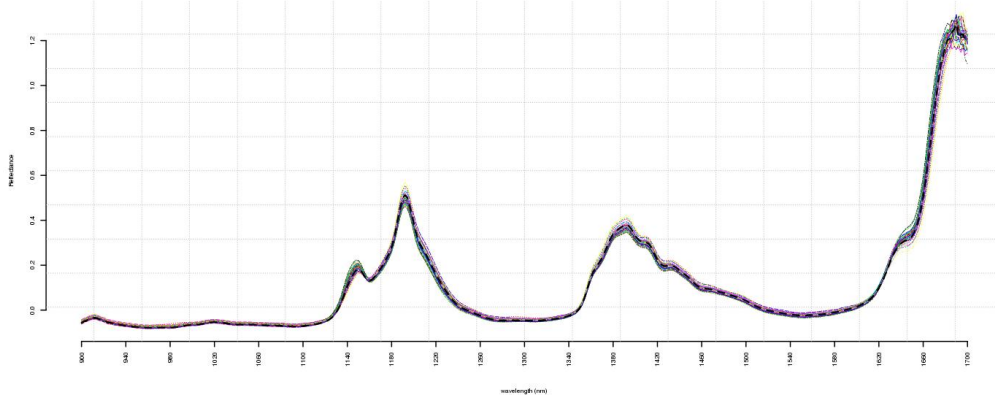


Figure 1: Gasoline data : Spectral data for 60 gasoline samples measured from 900 to 1700 nanometers (nm). The spectral data are registered every two nanometers.

vectors and matrices, respectively. The number of samples will be denoted by n while the number of predictors by p . The subscript m is used to denote the dimension of the PLS regression models, while the hat suffix is used for least squares fitted vectors. Further notations are introduced when needed.

2. PLS REGRESSION

Working within a linear model framework for regression problems the following linear model is assumed:

$$(2.1) \quad y_i = \mu + \mathbf{x}_i' \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \dots, n,$$

where y_i is the observed response for sample i , $\mathbf{x}_i$ are p -vectors of explanatory variables, $\boldsymbol{\beta}$ is the unknown p -vector of regression parameters, and ϵ_i the error term of the regression model. Without loss of generality we assume data to be centred to zero and therefore we freely assume $\mu = 0$. Using matrix notation: $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p)$ stands for the data matrix with predictors in its columns, $\mathbf{y}$ is the response vector, and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the unknown regression coefficient vector commonly estimated using least squares.

When the number of predictors (p) is large relative to the sample size (n) and/or the predictors are correlated, the least squares solution, when it exists, is highly variable due to rank deficiency of the data matrix $\mathbf{X}$. When $n < p$ the least squares solution doesn't even exist. In such cases, PLS regression offers an alternative by solving the regression problem after reducing its dimension; from hundreds of correlated predictors $\mathbf{x}_j$, $j = 1, \dots, p$, to a small set of orthogonal components $\mathbf{t}_m$ with $m \ll p$. These are linear combination of the original

predictors, and are used in the final regression on the response. PLS regression, therefore, iteratively approximates the least squares solution from a sequence of subspaces indexed by $m \leq p$. Using m orthogonal components in the final model, PLS regression lets for bias to decrease variance, and allows for a low mean square error for the final regression solution.

The restriction of orthogonal components may be relaxed in order to get PLS regression on orthogonal loadings. This has given rise to two different implementations of PLS regression, see [10] and [11]. The two algorithms are equivalent for prediction purposes; for a proof see [12]. Both PLS regression algorithms deflate data at each iteration, and $\mathbf{X}$ -residuals and $\mathbf{y}$ -residuals are used instead of $\mathbf{X}$ and $\mathbf{y}$ when $m > 1$. These are least squares residuals and will be denoted hereafter by $\mathbf{E}_m$ and $\mathbf{f}_m$, respectively, while we let $\mathbf{E}_0 = \mathbf{X}$ and $\mathbf{f}_0 = \mathbf{y}$. An important simplification when the response is a vector is the following: deflating $\mathbf{y}$ is not necessary; see [1]. More efficient computational algorithms for PLS regression without $\mathbf{X}$ -data deflation have been proposed in [13] and [14]. We provide in Algorithm (1) a sketch of the PLS regression on orthogonal loadings; see [11]. This implementation will be used in the PLS regression calculations throughout the rest of the paper.

Algorithm 1 Partial least squares regression on orthogonal loadings.

Input: For $i = 1, \dots, n$ and $j = 1, \dots, p$, $\mathbf{E}_0 = \mathbf{X}$ and $\mathbf{f}_0 = \mathbf{y}$.

For $m = 1, 2, \dots, k \leq p$

1. Compute $\mathbf{p}_m$ according to: $\mathbf{p}_m = \mathbf{E}'_{m-1} \mathbf{f}_{m-1}$.
2. Derive $\mathbf{t}_m = \mathbf{E}_{m-1} \mathbf{p}_m / \mathbf{p}'_m \mathbf{p}_m$ and store in $\mathbf{T}_m = (\mathbf{t}_1, \dots, \mathbf{t}_m)$.
3. $\mathbf{E}_m = \mathbf{E}_{m-1} - \mathbf{t}_m \mathbf{p}'_m$.
4. $\mathbf{f}_m = \mathbf{y} - \sum_{a=1}^m \mathbf{t}_a \hat{\mathbf{q}}_{ma}$ where
 $\hat{\mathbf{q}}_m = (\hat{q}_{m1}, \dots, \hat{q}_{ma}, \dots, \hat{q}_{mm})' = (\mathbf{T}'_m \mathbf{T}_m)^{-1} \mathbf{T}'_m \mathbf{y}$.

Output: Give the resulting sequence of the fitted vectors $\hat{\mathbf{y}}_m = \mathbf{T}_m \hat{\mathbf{q}}_m$.

The PLS regression coefficient vector $\hat{\boldsymbol{\beta}}_m^{pls}$ is determined by the matrix $\mathbf{P}_m$ containing in its columns the orthogonal loading vectors $\mathbf{p}_1, \dots, \mathbf{p}_m$. It is derived according to:

$$(2.2) \quad \hat{\boldsymbol{\beta}}_m^{pls} = \mathbf{P}_m \hat{\mathbf{q}}_m,$$

where $\hat{\mathbf{q}}_m$ is defined in Algorithm 1. Similar to principal components; see [15] the dimension reduction process of PLS implies a change of basis from the p -dimensional unit basis to a subspace of reduced dimension $m < p$. For principal components this corresponds to the subspace generated by a small set of selected eigenvectors. For PLS regression the new basis corresponds to the Krylov subspace of dimension up to m , defined as follows:

Definition 2.1. For matrix $\mathbf{A} = \mathbf{X}'\mathbf{X}$ and vector $\mathbf{b} = \mathbf{X}'\mathbf{y}$ the Krylov subspace of dimension $m \leq p$ is given by:

$$(2.3) \quad \mathcal{K}_m(\mathbf{b}, \mathbf{A}) = \text{span}(\mathbf{b}, \mathbf{A}^1 \mathbf{b}, \dots, \mathbf{A}^{m-1} \mathbf{b}).$$

The loading vectors in $\mathbf{P}_m$ (see Algorithm 1) span the Krylov subspace $\mathcal{K}_m(\mathbf{b}, \mathbf{A})$. The same holds for the PLS regression solution; see [12]. The PLS regression coefficient based on m components is given as the solution to:

$$(2.4) \quad \hat{\boldsymbol{\beta}}_m^{pls} = \operatorname{argmin}_{\boldsymbol{\beta}} \{(\mathbf{y} - \hat{\mathbf{y}})'(\mathbf{y} - \hat{\mathbf{y}})\} \quad \text{where } \hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}, \boldsymbol{\beta} \in \mathcal{K}_m(\mathbf{b}, \mathbf{A}).$$

Krylov spaces are location and scale invariant (see [16], chapter 12) and they further benefit from the following property:

Remark 2.1. For an orthogonal basis change in $\mathcal{K}_m(\mathbf{b}, \mathbf{A})$ induced by an orthogonal matrix $\mathcal{Q}$ we get an orthogonal similarity transformation of $\mathbf{A}$, that is:

$$(2.5) \quad \mathcal{K}_m(\mathcal{Q} \mathbf{b}, \mathcal{Q} \mathbf{A} \mathcal{Q}') = \mathcal{Q} \mathcal{K}_m(\mathbf{b}, \mathbf{A}), \text{ for } m \leq p.$$

The last property becomes even more interesting given that the Discrete Wavelet Transform (DWT), to be used in the following section, is such an orthogonal matrix.

3. Smooth PLS regression on wavelet transformed data

Spectral data are discrete values of continuous functions. Wavelets are used to approximate such functional data by means of the so-called mother and father wavelet, at different scales ℓ and locations k according to:

$$(3.1) \quad f(\mathbf{x}) = \sum_{k \in \mathbb{Z}} c_{\ell_0, k} \phi_{\ell_0, k}(\mathbf{x}) + \sum_{\ell_0 \leq \ell, k \in \mathbb{Z}} d_{\ell, k} \psi_{\ell, k}(\mathbf{x}),$$

where $c_{\ell, k}$ and $d_{\ell, k}$ are the scaling and detail wavelet coefficients, respectively. The father wavelet coefficient at scale zero (ℓ_0) reflects the global average of the spectrum, and when the data are centered it is equal to zero. The wavelet transform can be expressed as a matrix multiplication using the Discrete Wavelet Transform (DWT) matrix; see [17], Chapter 12 as well as [18], paragraph 4.3. This allows changing coordinates system from the original to the wavelet domain forwards and backwards. The operation is fast ([19]) and safe given that DWT is orthogonal. Each row spectrum $\mathbf{x}_i$ is mapped into a vector of wavelet coefficients $\tilde{\mathbf{x}}_i$ by means of matrix multiplication according to: $\tilde{\mathbf{x}}_i = \mathcal{W} \mathbf{x}_i$, where $\mathcal{W}$ is the DWT orthogonal matrix of dimension $p \times p$. Note that for a spectral data matrix $\mathbf{X}$ the DWT is given by postmultiplying the spectral data by $\mathcal{W}'$, to get:

$$(3.2) \quad \widetilde{\mathbf{X}} = \mathbf{X} \mathcal{W}'.$$

PLS regression on transformed data has been presented in [5]. It is run on the wavelet domain instead of the original spectra. The regression solution is then approximated on the wavelet domain as:

$$(3.3) \quad \hat{\boldsymbol{\beta}}_{m, \ell}^{pls} = \operatorname{argmin}_{\boldsymbol{\beta}} \{(\mathbf{y} - \hat{\mathbf{y}})'(\mathbf{y} - \hat{\mathbf{y}})\} \quad \text{where } \hat{\mathbf{y}} = \widetilde{\mathbf{X}} \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}} \in \mathcal{K}_m(\tilde{\mathbf{b}}, \tilde{\mathbf{A}}),$$

with $\tilde{\mathbf{A}} = \mathcal{W}_\ell \mathbf{A} \mathcal{W}_\ell'$ and $\tilde{\mathbf{b}} = \mathcal{W}_\ell \mathbf{b}$. The matrix $\mathcal{W}_\ell$ denotes the truncated DWT matrix of dimension $2^\ell \times p$. The use of the subscript ℓ for the coefficient vector in the transformed coordinates is used to highlight the wavelet truncation. Mother wavelet coefficients associated to the finest scales and very often the noisy part of the spectrum are truncated to zero. The final regression solution is recovered in original coordinates by means of the inverse DWT, denoted hereafter as iDWT. Using matrix multiplication this is the transpose of the DWT matrix. The PLS regression solution is smooth and given according to:

$$(3.4) \quad \hat{\beta}_m^{spls.1} = \mathcal{W}_\ell' \hat{\beta}_{m,\ell}^{pls}.$$

The authors in [5] used the term 'wavelet compressed data' to describe their algorithm motivated by the wavelet's outstanding performance to retain spectral information in a few wavelet coefficients. They truncated wavelet coefficients based on their variance spectrum, retaining most often the largest ones. Our motivation is smoothness. We truncate to zero wavelet coefficients associated to the finest resolution level scales. Other truncation strategies could be based upon other rules such as the universal threshold or using adaptive thresholding rules at each different resolution level; see [20], [21] and the references therein.

The smooth PLS regression algorithm based on wavelet transformed data is implemented using the orthogonal loadings PLS regression algorithm. It is similar to Algorithm 1, and therefore will not be given here. It uses all vectors and matrices z transformed in the wavelet domain and denoted $\tilde{z}$. For instance, the loading vector $\mathbf{p}_m$ is replaced by $\tilde{\mathbf{p}}_m$. The same holds for all data and residual data matrices, for the score vectors, and for the coefficient vectors $\mathbf{q}$ and β . Expression (3.4) is used in the end to recover the final regression solution back in the original coordinates system. The choice of ℓ is an additional argument in the algorithm's input.

4. PLS regression on smooth loadings

Transforming data to the wavelet domain is not the only one way to obtain a smooth PLS regression solution. Smoothness may be embedded directly on the loadings. This is done here by means of a PLS regression algorithm on smooth loadings. Wavelets are used on the loading vectors and data aren't transformed. At each iteration m the loading vector is reconstructed using a subset of the wavelet coefficients. The resulting loading vectors are both orthogonal and smooth. They are orthogonal due to the PLS algorithm, and smooth due to wavelet truncation. In terms of matrix multiplication we truncate the DWT matrix $\mathcal{W}$ to its first ℓ rows, that is, $\mathcal{W}_\ell$ which correspond to the coarsest scales. The resulting reconstructed smooth loading vector is given as: $\mathbf{p}_m^* = \mathcal{W}_\ell' \tilde{\mathbf{p}}_m$, with

$$(4.1) \quad \ddot{\mathbf{p}}_m = \sum_{\check{r}, \check{k} \in Z} d_{\check{r}, \check{k}} \psi_{\check{r}, \check{k}}(\mathbf{p}_m),$$

being the approximated loading vector using all the detail wavelet coefficients for scales up to $\check{r}$ and their associated locations $\check{k}$. The smooth loadings $(\mathbf{p}_1^*, \dots, \mathbf{p}_m^*)$ are stored in the matrix $\mathbf{P}_m^*$. Similarly the regression coefficients $\hat{q}_{ma}^*$ are stored in the vector $\hat{\mathbf{q}}_m^* = (\hat{q}_{m1}^*, \dots, \hat{q}_{ma}^*, \dots, \hat{q}_{mm}^*)'$. The final regression solution is given according to Expression (2.2) with matrix $\mathbf{P}_m^*$ taking over $\mathbf{P}_m$. The algorithm for PLS regression on smooth loadings is sketched in Algorithm 2.

Algorithm 2 PLS regression on smooth loadings.

Input: For $i = 1, \dots, n$ and $j = 1, \dots, p$, $\mathbf{E}_0 = \mathbf{X}$ and $\mathbf{f}_0 = \mathbf{y}$.

Select ℓ such that $2^\ell < p$ and compute $\mathcal{W}_\ell$.

For $m = 1, 2, \dots, k \leq p$

1. Compute $\mathbf{p}_m^*$ according to: $\mathbf{p}_m^* = \mathcal{W}_\ell' \ddot{\mathbf{p}}_m$,
where $\ddot{\mathbf{p}}_m$ as in Expression (4.1) with $\mathbf{p}_m = \mathbf{E}_{m-1}' \mathbf{f}_{m-1}$.
2. Derive $\mathbf{t}_m^* = \mathbf{E}_{m-1} \mathbf{p}_m^* / \mathbf{p}_m^{*'} \mathbf{p}_m^*$ and store in $\mathbf{T}_m^* = (\mathbf{t}_1^*, \dots, \mathbf{t}_m^*)$.
3. $\mathbf{E}_m = \mathbf{E}_{m-1} - \mathbf{t}_m^* \mathbf{p}_m^{*'}.$
4. $\mathbf{f}_m = \mathbf{y} - \sum_{a=1}^m \mathbf{t}_a^* \hat{q}_{ma}^*$ where
 $\hat{\mathbf{q}}_m^* = (\hat{q}_{m1}^*, \dots, \hat{q}_{ma}^*, \dots, \hat{q}_{mm}^*)' = (\mathbf{T}_m^{*'} \mathbf{T}_m^*)^{-1} \mathbf{T}_m^{*'} \mathbf{y}.$

Output: Give the resulting sequence of the fitted vectors $\hat{\mathbf{y}}_m^{pls} = \mathbf{X} \hat{\boldsymbol{\beta}}_m^{pls.2}$,
where $\hat{\boldsymbol{\beta}}_m^{pls.2} = \mathbf{P}_m^* \hat{\mathbf{q}}_m^*$ for $\mathbf{P}_m^* = (\mathbf{p}_1^*, \dots, \mathbf{p}_m^*)$.

The PLS regression on smooth loadings algorithm is computationally much faster than the algorithm for smooth PLS regression on wavelet transformed data. In the former algorithm the data are not transformed and only a few matrix-vector multiplications are required. In Algorithm 2 the wavelet expansion and truncation is done once for each loading vector. Normally the number of the extracted loadings is much smaller than the number of data samples. Moreover, the regression solution resulting from Algorithm 2 is on the original coordinates system and there is no need to be transformed back from the wavelet to the original domain. It turns out that the relation between the two algorithms is far more interesting from a theoretical point of view. This is further explored in the following section.

5. Theoretical aspects of smooth PLS regression

The relation between the two smooth PLS regression algorithms is explored here from a theoretical viewpoint. The loading and regression vectors resulting from the two smooth PLS regression implementations are investigated. Results

are given in the following propositions, while the proofs are provided separately in the Appendix.

Proposition 5.1. *The regression loadings $\tilde{\mathbf{p}}_m$ and $\ddot{\mathbf{p}}_m$ are identical.*

Proposition 5.2. *The smooth PLS regression loadings $\mathbf{p}_m^*$ computed in Algorithm 2 are orthogonal.*

Proposition 5.3. *The two smooth PLS regression algorithms generate the same sequence of approximate regression solutions, that is:*

$$(5.1) \quad \hat{\boldsymbol{\beta}}_{m,\ell}^{spls.1} = \hat{\boldsymbol{\beta}}_{m,\ell}^{spls.2} = \hat{\boldsymbol{\beta}}_{m,\ell}^{spls}.$$

Proposition 5.4. *Both algorithms approximate the solution of the linear system of equations*

$$(5.2) \quad \mathbf{M} \mathbf{A} \boldsymbol{\beta}_m^* = \mathbf{M}_2 \text{ with } \mathbf{M} = \mathcal{W}_\ell' \mathcal{W}_\ell \text{ for } m \leq p \text{ and } 2^\ell \leq p,$$

iteratively through Krylov subspace approximations.

As a direct consequence of Proposition 5.4 we state the following proposition.

Proposition 5.5. *For $m \leq p$ and increasing wavelet scale ℓ such that $2^\ell \rightarrow p$ the sequence of smooth PLS regression solutions generates the same subspaces and converges to the sequence of ordinary PLS regression solutions, that is:*

$$\hat{\boldsymbol{\beta}}_{m,\ell}^{spls} \rightarrow \hat{\boldsymbol{\beta}}_m^{pls}.$$

For both ordinary and smooth PLS regression the reduction of the dimension of the regression problem from large- p to small- m is almost identical. This is stated in the proposition below by employing the term of equivalence. The proof for Proposition 5.6 is given in the Appendix.

Proposition 5.6. *Ordinary and smooth PLS regression models are equivalent in reducing the dimension of the regression problem.*

Proper model selection is crucial for smooth PLS regression as it is for ordinary PLS regression. Prior to applying and assessing smooth PLS regression one needs to identify the dimension of the regression model, that is, the number of PLS regression components to be retained. This is done in the following section by means of cross validation prior to investigating smooth PLS regression on three well known NIR data sets.

6. Experience with NIR data

Three well-known data from NIR spectroscopy are used here to assess smooth PLS regression. These are the diesel, the gasoline, and the biscuit data sets. All of them are available through the internet. The diesel data has been downloaded from the Eigenvector Research site at <http://www.eigenvector.com/data/SWRI/>, while the gasoline and the biscuit data have been downloaded from the **R** packages `pls` ([22]) and `ppls` ([23]) through the **R** website at <http://www.r-project.org/>. All three NIR data sets have been extensively used in the literature; see for instance [2], [24], [7], [8], and [9].

The diesel and the gasoline data sets quantify the cetane and the octane number of 381 diesel and 60 gasoline samples, respectively. The cetane number for diesel samples is the equivalent of the octane number for gasoline samples. The biscuit data measure fat concentration of 71 cookies. The data include information on 72 biscuit samples, yet, observation 23 is removed as a reported outlier. One can find more information on these three NIR data sets in the references given above. All three data sets use spectra for predictors. The NIR for the analyzed samples are registered over a broad range of wavelengths, measured in nanometers (nm). We retained in the analysis the appropriate wavelength ranges in order to build spectra of appropriate length (equal to a power of 2). For all three data sets the length of the spectra equals $256 = 2^8$.

The data have been centered prior to regression analysis by subtracting column means. They have been randomly split on 10 folds, and a 10-fold cross validation (see [25], Chapter 7) has been used in order to assess the number of PLS components. The NIR data ($\mathcal{D}$) have been split into 10 mutually exclusive groups, forming a training set $\mathcal{D}_{train}$ (used for model construction) and a test set $\mathcal{D}_{test} = \mathcal{D}^*$ (used for model validation), where $\mathcal{D}_{train} \cap \mathcal{D}_{test} = \emptyset$ and $\mathcal{D}_{train} \cup \mathcal{D}_{test} = \mathcal{D}$. The cross validated mean squared prediction error MSEP^{cv} for a regression model based on m components, has been computed according to:

$$(6.1) \quad \text{MSEP}_m^{cv} = \mathbb{E}_K \left[\mathbb{E}_k \left(\mathcal{L}(\mathbf{y}^*, \hat{\mathbf{y}}_m^{*(-k)}) \right) \right],$$

where the superscript $*$ is used to indicate the observations in $\mathcal{D}^*$, and $k = 1, \dots, K$ the part of the $K = 10$ groups of data which are left out. The notation $\mathbb{E}_K$ highlights average over the K different splits, while $\mathbb{E}_k$ indicates average over the number of observations inside the k^{th} test set. The suffix $(-k)$ indicates that the fits are given by the investigated regression model on the data set excluding the k^{th} part. Using the same splits we did the same for the smooth PLS regression using wavelet approximation including wavelet scales up to $\ell = 6$ and $\ell = 7$. The results for the model selection study are reported in Table 1.

The PLS regression model selection results in Table 1 are similar to the ones already known from the existing literature. Furthermore, the model selection results for the smooth PLS regression are almost identical to the PLS regression

Table 1: NIR data: 10-fold cross-validation estimates for the prediction loss of the PLS and the smooth PLS regression models (sPLS $_{\ell}$) including 1 to 10 components for $\ell = 7$ and $\ell = 6$, respectively.

		1	2	3	4	5	6	7	8	9	10
diesel	PLS	3.09	2.84	2.64	2.27	2.09	2.26	1.99	2.09	2.17	2.15
	sPLS ₇	2.60	2.44	2.02	2.35	2.12	2.39	2.20	2.04	2.05	2.07
	sPLS ₆	2.04	2.03	1.98	1.98	1.77	1.75	1.53	1.53	1.55	1.55
gasoline	PLS	0.79	0.29	0.23	0.25	0.25	0.26	0.30	0.28	0.27	0.24
	sPLS ₇	0.83	0.23	0.11	0.15	0.14	0.13	0.15	0.19	0.15	0.13
	sPLS ₆	0.79	0.21	0.11	0.11	0.12	0.10	0.13	0.17	0.18	0.17
biscuit	PLS	1.25	1.33	0.79	0.42	0.25	0.30	0.28	0.30	0.28	0.27
	sPLS ₇	1.86	1.80	1.34	0.92	0.637	0.45	0.39	0.37	0.37	0.35
	sPLS ₆	1.07	1.12	0.58	0.43	0.40	0.28	0.23	0.27	0.25	0.24

results. As expected, the minimum prediction loss for smooth PLS regression is reached after retaining almost the same number of components as for ordinary PLS regression. The estimated out-of-sample prediction error for smooth PLS regression is sometimes even reduced compared to ordinary PLS regression prediction error. Notably for the gasoline data the prediction performance for smooth PLS improves substantially compared to ordinary PLS regression. Yet, this is not the case for the biscuit data.

Figures 2, 3, and 4 illustrate the regression solutions for PLS and smooth PLS regression. Black solid lines and points are used to depict the PLS regression solution, while dashed lines are used for smooth PLS regression results. For illustration purposes selected wavelength regions are magnified and plotted in the lower left and right panels. These allow better inspecting the smoothness induced by the use of the smooth PLS regression.

For the diesel and the gasoline data set in Figures 2 and 3 the smooth PLS regression solution efficiently smooths the PLS regression coefficient vector especially for $\ell = 6$, see the light gray (blue) dashed line. The lower panel plots help discriminating between the three solutions. The smooth PLS regression coefficient is less efficient in smoothing the final solution for the biscuit data; see Figure 4. The ordinary PLS regression solution for this data set was already rather smooth.

Finally it is worth noting that smooth PLS regression may improve the prediction performance notably when the PLS regression solution is noisy. Smoothing reduces the prediction error in the diesel and the gasoline data. In contrast this is not the case in the biscuit data where PLS regression is already smooth.

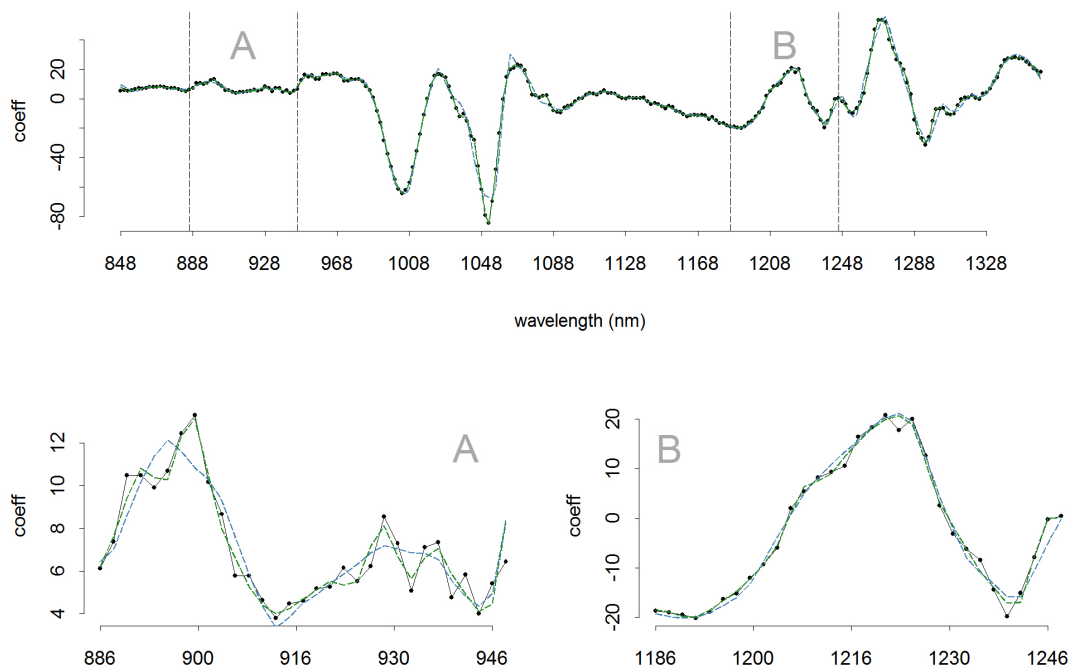


Figure 2: Diesel data. Regression coefficient for a regression model including 7 components. Response is the cetane number of the diesel samples and predictors are the NIR spectra over the wavelength region from 848 to 1358 nanometers (nm). Black points and black thin line correspond to the PLS regression coefficient. The smooth PLS regression coefficients with $\ell = 7$ and $\ell = 6$ are plotted in green and blue dashed lines, respectively. Selected wavelength regions (A and B) are magnified in the lower left and right panels.

7. Conclusions

Most spectral data used in chemometrics are high dimensional and very often functional. PLS regression methods are well suited for high dimensional data. Wavelets are well suited for functional data. We explored the combination of these two in order to build smooth alternatives for PLS regression. The rationale behind smooth PLS regression stemmed from the fact that PLS regression coefficients are low dimensional approximations for the regression solution and should exhibit some degree of smoothness.

We showed that PLS regression can be effectively combined to wavelets for functional data analysis and provide smooth regression solutions to high di-

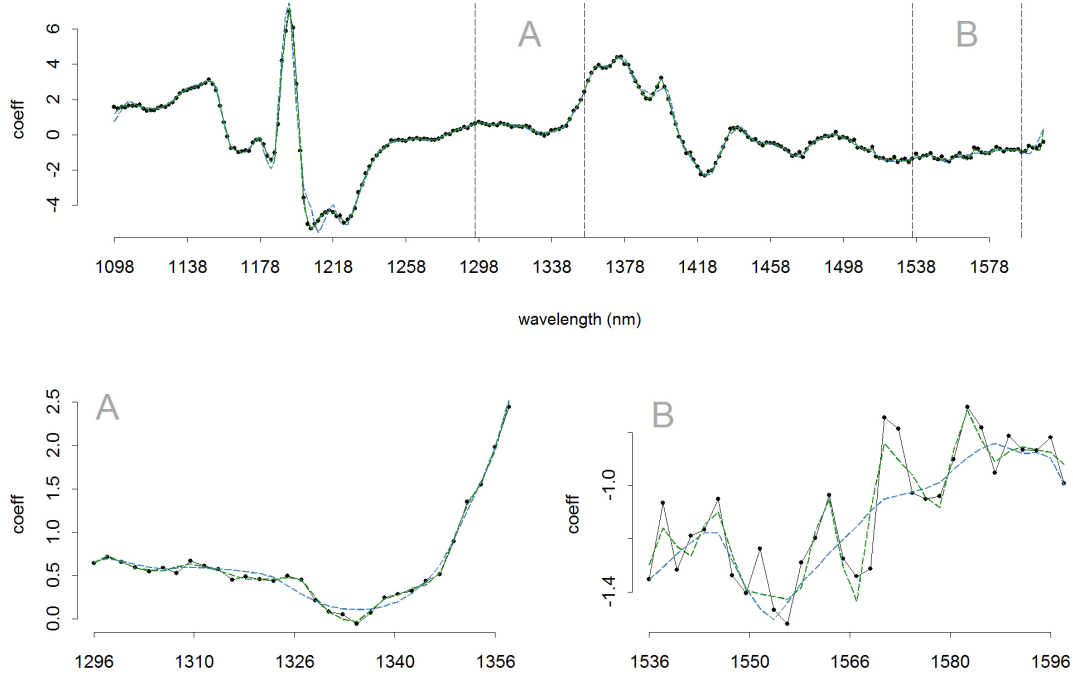


Figure 3: Gasoline data. Regression coefficient vector for a regression model including 3 components. Response is the octane number of 60 gasoline samples and predictors are the NIR spectra over the wavelength region from 1098 to 1608 nanometers (nm) in steps of two. Black points and black thin line correspond to the PLS regression coefficient. The smooth PLS regression coefficients with $\ell = 7$ and $\ell = 6$ are plotted in green and blue dashed lines, respectively. Selected wavelength regions (A and B) are magnified in the lower left and right panels.

mensional regression problems. Wavelet expansion and truncation allowed us building two equivalent smooth PLS regression algorithms. The two algorithmic implementations for smooth PLS regression have been proven to be equivalent and to produce the same sequence of approximate solutions. These are regression solutions approximated through Krylov subspaces of dimension $m \leq p$. They are, therefore, PLS regression solutions. Working in the framework of spectral data we focused on near infra-red experiments which have been used to illustrate the potential of smooth PLS regression using wavelets. Three well known NIR data sets from the literature have been used to confirm that smooth PLS regression is a valuable alternative to ordinary PLS regression for smoothing the final regression solution while maintaining good prediction performance and dimension reduction.

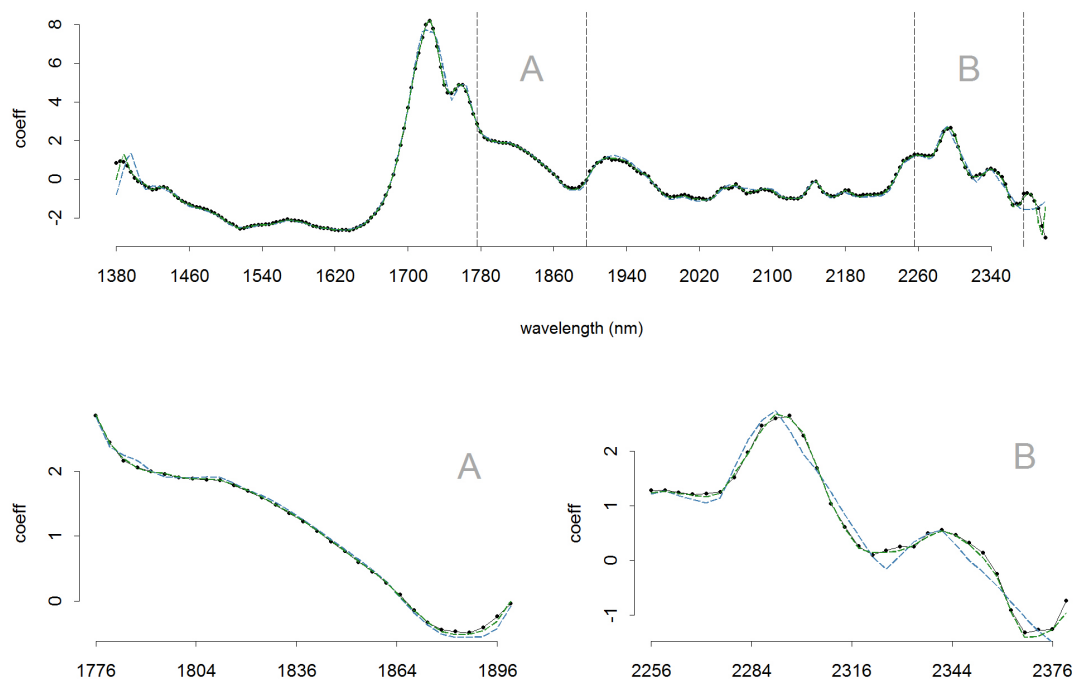


Figure 4: Biscuit data. Regression coefficient vector for a regression model including 5 components. Response is the fat concentration of biscuit samples and predictors are the NIR spectra over the wavelength region from 1100 to 2498 nanometers (nm). Black points and black thin line correspond to the PLS regression coefficient. The smooth PLS regression coefficients with $\ell = 7$ and $\ell = 6$ are plotted in green and blue dashed lines, respectively. Selected wavelength regions (A and B) are magnified in the lower left and right panels.

The two presented smooth PLS regression algorithms have been implemented based on the PLS regression algorithm on orthogonal loadings. It is straightforward to implement both using the PLS regression algorithm on orthogonal scores; the results will be identical. The implementation of the proposed methods is straightforward. We used the S-PLUS wavelet package S+WAVELETS in our implementation; see [17]. Similar computer packages for wavelet analysis exist in **R**, as well; see for instance the `wavethresh` package in **R** (see [22]). Existing computational tools give all that is required for further smooth PLS regression developments.

ACKNOWLEDGMENTS

The author would like to thank the two referees for their very constructive comments and their suggestions.. The author would also like to express his gratitude to Prof. Joe Whittaker and Prof. Alina Matei for their support.

Appendix

Prior to the proof of the propositions in Section 5 we state two lemmas required for the development of the proofs. The proof for Lemma 7.1 is a direct consequence of wavelet properties and is omitted; the interest reader can see [18], paragraph 4.3.1. The proof for Lemma 7.2 is provided below using mathematical induction. Finally the notation $2^\ell \rightarrow p$ is used to denote the increasing order approximation of $\mathbf{X}$ by allowing finer scales to be included in the rows of matrix $\mathcal{W}_\ell$.

Lemma 7.1. *For the truncated matrix $\mathcal{W}_\ell$ of dimension $2^\ell < p$ we have:*

1. *All cross-product matrices $\mathcal{W}_\ell' \mathcal{W}_\ell$ with $2^\ell < p$ are block-diagonal, with*

$$\mathcal{W}_\ell' \mathcal{W}_\ell \rightarrow I_p \text{ as } 2^\ell \rightarrow p,$$

where I_p is used to denote the identity matrix of order p .

2. *All cross-product matrices $\mathcal{W}_\ell \mathcal{W}_\ell'$ with $2^\ell \leq p$ satisfy:*

$$\mathcal{W}_\ell \mathcal{W}_\ell' = I_p.$$

Lemma 7.2. *For all $m \leq p$, $\mathbf{E}_m \mathcal{W}_\ell' = \widetilde{\mathbf{E}}_m^{(\ell)}$.*

Proof of Lemma 7.2: We use mathematical induction. For $m = 1$ the lemma holds given:

$$\mathbf{E}_0 \mathcal{W}_\ell' = \mathbf{X} \mathcal{W}_\ell' = \widetilde{\mathbf{X}}^{(\ell)} = \widetilde{\mathbf{E}}_0^{(\ell)}.$$

Let it be true for $m - 1$, that is assume that:

$$\mathbf{E}_{m-1} \mathcal{W}_\ell' = \widetilde{\mathbf{E}}_{m-1}^{(\ell)}.$$

We will prove that this also holds for m , that is:

$$(7.1) \quad \mathbf{E}_m \mathcal{W}_\ell' = \widetilde{\mathbf{E}}_m^{(\ell)}.$$

We develop separately both sides of Expression (7.1). For the left hand side of Expression (7.1) we have:

$$\begin{aligned} \mathbf{E}_m \mathcal{W}_\ell' &= (\mathbf{E}_{m-1} - \mathbf{t}_m^* \mathbf{p}_m^{*'}) \mathcal{W}_\ell' \\ &= (\mathbf{E}_{m-1} - \mathbf{E}_{m-1} \mathbf{p}_m^* \mathbf{p}_m^{*'}) \mathcal{W}_\ell' \\ &= (\mathbf{E}_{m-1} - \mathbf{E}_{m-1} \mathcal{W}_\ell' \ddot{\mathbf{p}}_m \ddot{\mathbf{p}}_m' \mathcal{W}_\ell) \mathcal{W}_\ell' \\ &= \mathbf{E}_{m-1} \mathcal{W}_\ell' - \mathbf{E}_{m-1} \mathcal{W}_\ell' \ddot{\mathbf{p}}_m \ddot{\mathbf{p}}_m' \mathcal{W}_\ell \mathcal{W}_\ell' \\ &= \mathbf{E}_{m-1} \mathcal{W}_\ell' - \mathbf{E}_{m-1} \mathcal{W}_\ell' \ddot{\mathbf{p}}_m \ddot{\mathbf{p}}_m' \\ &= \mathbf{E}_{m-1} \mathcal{W}_\ell' (I - \ddot{\mathbf{p}}_m \ddot{\mathbf{p}}_m'). \end{aligned}$$

For the right hand side of Equation (7.1) we have:

$$\begin{aligned}\tilde{\mathbf{E}}_m^{(\ell)} &= \tilde{\mathbf{E}}_{m-1}^{(\ell)} - \tilde{\mathbf{t}}_m \tilde{\mathbf{p}}_m' \\ &= \tilde{\mathbf{E}}_{m-1}^{(\ell)} - \tilde{\mathbf{E}}_{m-1}^{(\ell)} \tilde{\mathbf{p}}_m \tilde{\mathbf{p}}_m' \\ &= \tilde{\mathbf{E}}_{m-1}^{(\ell)} (I - \tilde{\mathbf{p}}_m \tilde{\mathbf{p}}_m').\end{aligned}$$

Furthermore, given Expression (4.1) we have:

$$\ddot{\mathbf{p}}_m \ddot{\mathbf{p}}_m' = \mathcal{W}_\ell \mathbf{p}_m \mathbf{p}_m' \mathcal{W}_\ell' = \tilde{\mathbf{p}}_m \tilde{\mathbf{p}}_m',$$

which completes the proof.

Proof of Proposition 5.1: Recall that for univariate PLS regression there is no need to deflate the response vector $\mathbf{y}$. The loading vector $\ddot{\mathbf{p}}_m$ in Expression (4.1) can be written in matrix form as $\mathcal{W}_\ell \mathbf{p}_m$; it then follows:

$$\ddot{\mathbf{p}}_m = \mathcal{W}_\ell \mathbf{p}_m = \mathcal{W}_\ell \mathbf{E}_{m-1}' \mathbf{y} = (\mathbf{E}_{m-1} \mathcal{W}_\ell')' \mathbf{y} = \tilde{\mathbf{E}}_{m-1}^{(\ell)'} \mathbf{y} = \tilde{\mathbf{p}}_m.$$

Proof of Proposition 5.2: Using Proposition 5.1 and noting that the loading vectors $\tilde{\mathbf{p}}$ are orthogonal by construction (they are the ordinary PLS regression loadings in the wavelet domain), it follows that:

$$\mathbf{p}_i^* \mathbf{p}_j^* = \ddot{\mathbf{p}}_i' \mathcal{W}_\ell \mathcal{W}_\ell' \ddot{\mathbf{p}}_j = \tilde{\mathbf{p}}_i' \mathcal{W}_\ell \mathcal{W}_\ell' \tilde{\mathbf{p}}_j = \tilde{\mathbf{p}}_i' \tilde{\mathbf{p}}_j = 0, \text{ for } i \neq j \text{ and } i, j \leq p.$$

Therefore the smooth PLS regression loadings $\mathbf{p}^*$ are orthogonal.

Proof of Proposition 5.3: The smooth regression coefficients $\hat{\beta}_m^{spls.1}$ and $\hat{\beta}_m^{spls.2}$ are identical, as:

$$\begin{aligned}\hat{\beta}_m^{spls.2} &= \mathbf{P}_m^* \hat{\mathbf{q}}_m^* \\ &= \mathcal{W}_\ell' \ddot{\mathbf{P}}_m \hat{\mathbf{q}}_m^* \\ &= \mathcal{W}_\ell' \tilde{\mathbf{P}}_m \hat{\mathbf{q}}_m \\ &= \mathcal{W}_\ell' \hat{\beta}_m^{pls} = \hat{\beta}_m^{spls.1}.\end{aligned}$$

Note that $\hat{\mathbf{q}}_m = \hat{\mathbf{q}}_m^*$. This is justified by the fact that both are implied by the loading's matrix $\tilde{\mathbf{P}}_m$ and $\mathbf{P}_m$, respectively. These are, yet, identical as shown in Proposition 5.1.

Proof of Proposition 5.4: The link between PLS regression and conjugate gradients for solving large linear system of equations is well-known; see for instance [26]. The solution to the system of equations is approximated through Krylov subspaces. The system in (5.2) is pre-multiplied by a non-singular matrix $\mathbf{M}$. This is sometimes referred to in numerical analysis as a preconditioned system. While preconditioning mainly focuses on improvement in the convergence of iterative solution methods, such as the Krylov methods, here it is used to induce smoothness. This is done by using $\mathbf{M} = \mathcal{W}_\ell' \mathcal{W}_\ell$. The two smooth PLS regression

algorithms are two facets of preconditioning the conjugate gradients. While the former operates on transformed coordinates ($\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$), the latter (Algorithm 2) iterates starting from directions determined by matrix $\mathbf{M}$. The equivalence between these two algorithms is sketched below:

$$\begin{aligned} \mathbf{M} \mathbf{A} \boldsymbol{\beta}_m^* &= \mathbf{M} \mathbf{b} \\ \mathcal{W}'_\ell \mathcal{W}_\ell \mathbf{A} \boldsymbol{\beta}_m^* &= \mathcal{W}'_\ell \mathcal{W}_\ell \mathbf{b} \\ \mathcal{W}_\ell \mathbf{A} \mathcal{W}'_\ell \tilde{\boldsymbol{\beta}}_m &= \mathcal{W}_\ell \mathbf{b} \\ \tilde{\mathbf{A}} \tilde{\boldsymbol{\beta}}_m &= \tilde{\mathbf{b}}, \text{ for } m \leq p. \end{aligned}$$

The final solution $\tilde{\boldsymbol{\beta}}$ can be transformed back in the original coordinates according to:

$$\boldsymbol{\beta}_m^* = \mathcal{W}'_\ell \tilde{\boldsymbol{\beta}}_m,$$

in exactly the same manner that the loading vectors $\tilde{\mathbf{p}}$ can be also transformed back in original coordinates as:

$$\mathbf{p}_m^* = \mathcal{W}'_\ell \tilde{\mathbf{p}}_m.$$

Proof of Lemma 5.5: For $\mathbf{M} = I_p$ in the system of equations (5.2) the ordinary PLS regression solution is recovered. This happens for increasing ℓ as $2^\ell \rightarrow p$. The PLS regression solution is a Krylov solution, that is:

$$\hat{\boldsymbol{\beta}}_m^{pls} \in \mathcal{K}_m(\mathbf{b}, \mathbf{A}) \text{ for } m \leq p.$$

The smooth PLS regression solution given in Expression (3.4) as:

$$\hat{\boldsymbol{\beta}}_m^{spls} = \mathcal{W}'_\ell \hat{\boldsymbol{\beta}}_m^{pls}, \text{ for } m \leq p,$$

is a Krylov solution. Combining remark (2.1) and expression (2.5) to the orthogonality of the DWT matrix $\mathcal{W}$, as long as $2^\ell \rightarrow p$ one gets:

$$\hat{\boldsymbol{\beta}}_m^{spls} \in \mathcal{W}'_\ell \mathcal{K}_m(\mathcal{W}_\ell \mathbf{b}, \mathcal{W}_\ell \mathbf{A} \mathcal{W}'_\ell) = \mathcal{W}'_\ell \mathcal{W}_\ell \mathcal{K}_m(\mathbf{b}, \mathbf{A}) \cong \mathcal{K}_m(\mathbf{b}, \mathbf{A}) \text{ for } m \leq p.$$

Proof of Proposition 5.6: The dimension reduction performance of both ordinary and smooth PLS regression is determined by the minimum number of iterations required to achieve the best approximate solution to the system of equations in (5.2). This is strongly dependent on the spectrum of $\mathbf{A}$ and $\mathcal{M}\mathbf{A}$ for ordinary and smooth PLS regression, respectively. Let $S(\mathbf{A})$ be the spectrum of a symmetric matrix $\mathbf{A}$ as given by its eigen decomposition $\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}'$ with $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_p)$ denoting the diagonal matrix of eigenvalues of $\mathbf{A}$, and $\mathbf{V}$ its orthonormal set of eigenvectors. Similarly, let $S(\mathcal{M}\mathbf{A})$ be the spectrum of the symmetric matrix $\tilde{\mathbf{A}}$. A sufficient condition for Proposition 5.6 to hold is given below:

Ordinary and smooth PLS regression are approximately equivalent in reducing the dimension of the regression problem whenever:

$$S(\mathcal{M}\mathbf{A}) \approx S(\mathbf{A}).$$

Consider the eigen decomposition of matrix $\tilde{\mathbf{A}}$ as follows:

$$\begin{pmatrix} \tilde{\mathbf{V}}_\ell & \tilde{\mathbf{V}}_{\bar{\ell}} \end{pmatrix} \times \begin{pmatrix} \tilde{\Lambda}_\ell & 0 \\ 0 & \tilde{\Lambda}_{\bar{\ell}} \end{pmatrix} \times \begin{pmatrix} \tilde{\mathbf{V}}'_\ell \\ \tilde{\mathbf{V}}'_{\bar{\ell}} \end{pmatrix},$$

for $\tilde{\mathbf{V}}_\ell = \mathcal{W}_\ell \mathbf{V}_\ell$ and $\tilde{\mathbf{V}}_{\bar{\ell}} = \mathcal{W}_{\bar{\ell}} \mathbf{V}_{\bar{\ell}}$, where the subscript ℓ is used to denote the ℓ -scales wavelet approximation and $\bar{\ell}$ used to denote the excluded wavelet scales. The expression above simplifies to:

$$(7.2) \quad \mathbf{W}_\ell \mathbf{V}_\ell \tilde{\Lambda}_\ell \mathbf{V}'_\ell \mathbf{W}'_\ell + \mathbf{W}_{\bar{\ell}} \mathbf{V}_{\bar{\ell}} \tilde{\Lambda}_{\bar{\ell}} \mathbf{V}'_{\bar{\ell}} \mathbf{W}'_{\bar{\ell}},$$

We discuss the two following cases:

1. When $2^\ell = p$, the second term in Expression (7.2) disappears and $S(\tilde{\mathbf{A}}) = S(\mathbf{A})$ since $\mathcal{W}_\ell$ is the identity matrix and $\mathbf{V}_\ell = \mathbf{V}$. The two regression methods are then identical in reducing the dimension of the regression problem.
2. When $2^\ell < p$ the second term in Expression (7.2) is generally much smaller than the first term, especially for collinear and functional data (such as the NIR data) where PLS regression is used. The diagonal entries in $\tilde{\Lambda}_{\bar{\ell}}$ are close to zero and the second term in Expression (7.2) vanishes; hence the spectrum of $\mathbf{A}$ is approximated by the first term and $S(\mathcal{M}\mathbf{A}) \approx S(\mathbf{A})$.

REFERENCES

- [1] HÖSKULDSSON, A. (1988). PLS regression methods, *Journal of Chemometrics*, **2**, 211–228.
- [2] GOUTIS, C. AND FEARN, T. (1996). Partial Least Squares Regression on Smooth Factors, *Journal of the American Statistical Association*, **91**, 434, 627–632.
- [3] SILVERMAN, B.W. (1986). *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London.
- [4] DURAND, J.F. AND SABATIER, R. (1997). Additive Splines for Partial Least Squares Regression, *Journal of the American Statistical Association*, **92**, 440, 1546–1554.
- [5] TRYGG, J. AND WOLD, S. (1998). PLS regression on wavelet compressed NIR spectra, *Chemometrics and Intelligent Laboratory Systems*, **42**, 1, 209–220.
- [6] PREDAC, C. AND SAPORTA, G. (2005). PLS regression on a stochastic process, *Computational Statistics & Data Analysis*, **48**, 1, 149–158.
- [7] REISS, P.T. AND OGDEN, R.T. (2007). Functional Principal Component Regression and Functional Partial Least Squares, *Journal of the American Statistical Association*, **102**, 984–996.
- [8] KRÄMER, N. AND BOULESTEIX, A.L. AND TUTZ, G. (2008). Penalized Partial Least Squares with applications to B-spline transformations and functional data, *Chemometrics and Intelligent Laboratory Systems*, **94**, 1, 60–69.
- [9] KONDYLLIS, A. AND WHITTAKER, J. (2013). Feature Selection for functional PLS, *Chemometrics and Intelligent Laboratory Systems*, **121**, 82–89.
- [10] WOLD, S. AND MARTENS, H. AND WOLD, H. (1983). *The multivariate calibration problem in chemistry solved by the PLS method*. In "Lecture notes in mathematics. Proc. Conf. Matrix Pencils" (A. Ruhe, and B. Kagström, Eds.), Springer-Verlag, 286–293.
- [11] MARTENS, H. AND T. NAES (1989). *Multivariate Calibration*, John Wiley & Sons, New York.
- [12] HELLAND, I.S. (1988). On the structure of partial least squares regression, *Communications in Statistics - Simulation and Computation*, **17**, 581–607.
- [13] DE JONG, S. (1993). SIMPLS: An alternative approach to partial least squares regression, *Chemometrics and Intelligent Laboratory Systems*, **18**, 251–263.
- [14] GOUTIS, C. (1997). A Fast Method to Compute Orthogonal Loadings Partial Least Squares, *Journal of Chemometrics*, **11**, 33–38.
- [15] JOLLIFFE, I.T. (2002). *Principal Component Analysis*, Springer Verlag.
- [16] PARLETT, B. (1980). *The Symmetric Eigenvalue Problem*, Prentice-Hall Series in Applied Mathematics, New Jersey.
- [17] BRUCE, A. AND GAO, H.Y. (1996). *Applied Wavelet Analysis with S-PLUS*, Springer-Verlag, New York.
- [18] VIDA KOVIC, B. (1999). *Statistical Modeling by Wavelets*, John Wiley & Sons, New York.

- [19] MALLAT, S.G. (1989). A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 7, 674–693.
- [20] DONOHO, D.L. AND JOHNSTONE, I.M. (1995). Adapting to unknown smoothness via wavelet shrinkage, *Journal of the American Statistical Association*, **90**, 432, 1200–1224.
- [21] JANSEN, M. (2000). *Noise Reduction by Wavelets Thresholding*, Springer, Lecture Notes in Statistics, New York.
- [22] NASON, G. (2013). wavethresh: Wavelets statistics and transforms, <http://CRAN.R-project.org/package=wavethresh>, R package version 4.6.6.
- [23] KRÄMER, N. AND BOULESTEIX, A.L. (2014). ppls: Penalized Partial Least Squares, <http://CRAN.R-project.org/package=ppls>, R package version 1.6.1.
- [24] KALIVAS, J.H. (1997). Two data sets of near infrared spectra, *Chemometrics and Intelligent Laboratory Systems*, **37**, 255–259.
- [25] HASTIE, T. AND TIBSHIRANI, R. AND FRIEDMAN, J. (2009). *Elements of Statistical Learning, 2nd Edition*, Springer Series in Statistics, New York.
- [26] PHATAK, A. AND DE HOOG, F. (2001). *PLSR, Lanczos, and Conjugate Gradients*, Commonwealth Scientific and Industrial Research Organisation, CMIS 01/122.