ANALYSIS OF NONLINEAR PARTIAL LEAST SQUARES ALGORITHMS

S. Kumar * **U. Kruger** **,1 **E. B. Martin, and** *
A. J. Morris *

* *Centre of Process Analytics and Process Technology, University of Newcastle, NE1 7RU, U.K.*
** *Intelligent Systems and Control Group, Queen's University Belfast, BT9 5AH, U.K.*

Abstract: This paper presents an analysis of nonlinear extensions to Partial Least Squares (PLS) using error-based minimization techniques. The analysis revealed that such algorithms are maximizing the accuracy with which the response variables are predicted. Therefore, such algorithms are nonlinear reduced rank regression algorithms rather than nonlinear PLS algorithms.

Keywords: Nonlinear Systems, Identification Algorithms, Models, Gradient Methods, Prediction

1. INTRODUCTION

A number of extensions to conventional PLS have been proposed for identifying nonlinear relationship between two variable sets, which are usually referred to as the predictor set that is predictive and the response set that is predicted. For these extensions, the iterative nature of PLS is taken advantage of in terms of reducing the modelling task to the identification of score models in the latent variable space.

The score models rely on projecting recorded observations of the predictor and response variables on respective one-dimensional subspaces, defined by weight vectors. The projections of the response variables, further referred to as u-scores, are then predicted by nonlinear functions of the projections of the predictor variable, denoted as t-scores. (Wold *et al.*, 1989) suggested to identify the score models using second order polynomials. More flexible nonlinear models were proposed by (Frank, 1990) and (Wold, 1992). Whilst the former work included the use of a smoothing procedure, the technique by (Wold, 1992) incorporated spline functions.

(Qin and McAvoy, 1992; Holcomb and Morari, 1992; Wilson *et al.*, 1997) used artificial neural networks and more recently, (Patwardhan and Lakshminarayan, 1998) exploited the use of Hammerstein and Wiener models, (Hiden *et al.*, 1998) employed Genetic Programming and (Li *et al.*, 2001) utilized the Box-Tidwell transformation. However, (Berglund and Wold, 1997) argued that artificial neural networks and spline functions (i) inherent a considerable degree of flexibility and (ii) may lead to overfitting. A revision of the work by (Wold *et al.*, 1989), incorporating low order polynomials, was given by (Baffi *et al.*, 1999*a*), where a number of modifications were introduced to produce a simpler algorithm.

In this article, the work by (Wold *et al.*, 1989) and the modifications that (Baffi *et al.*, 1999*a*) introduced are investigated. This analysis revealed the following contributions, summarized in the

[1] Corresponding Author: uwe.kruger@ee.qub.ac.uk; Tel: +44(0)2890 274059; Fax: +44(0)2890 667023

presented work. Firstly, if a linear structure for each score model is considered, the technique by (Baffi *et al.*, 1999*a*) is equivalent to reduced rank regression (RRR). Therefore, each score model maximally reduces the variation of the response variables. Secondly, if nonlinear score models are identified, the algorithm by (Baffi *et al.*, 1999*a*) is consequently a nonlinear RRR (NLRRR) technique. Thus, it should not be considered as a nonlinear PLS technique, since minimizing the variation of the prediction error of the response variables does not guarantee maximum variance between pairs of score variables. A maximum covariance, however, is the criterion with which a conventional PLS model is identified. Thirdly, using NLRRR, the direction on which the response variables are projected is constrained to be equivalent to the direction for predicting the response variables by the predicted u-scores.

2. PARTIAL LEAST SQUARES PRELIMINARIES

PLS is designed to analyze the relationships between the predictor and the response variables. The observations of each variable are stored in matrices, i.e. the predictor matrix $\mathbf{X} \in \mathbb{R}^{K \times N}$ and response matrix $\mathbf{Y} \in \mathbb{R}^{K \times M}$. Each column refers to a particular predictor or response variable and each row refers to a particular observation, K represents the number of observations, N is the number of predictor variables and M is the number of response variables. Typically, $\mathbf{X}$ and $\mathbf{Y}$ are mean centered and appropriately scaled prior to the applications of PLS. It is assumed throughout this article that $K \gg N, M$ and that the rank of $\mathbf{X}$ and $\mathbf{Y}$ is N and M, respectively. Note, however, that these assumptions are for the convenience of the presentation rather than imposing restrictions onto the presented methods.

2.1 PLS Algorithms

PLS determines the k^{th} pair of weight vectors, $\mathbf{v}_k$ and $\mathbf{w}_k$, and score vectors, $\mathbf{t}_k$ and $\mathbf{u}_k$, i.e. the vectors in which the t- and u-scores are stored, by maximizing $J_k^{(\mathbf{v},\mathbf{w})}$:

$$J_k^{(\mathbf{v},\mathbf{w})} = \mathbf{t}_k^T \mathbf{u}_k = \mathbf{w}_k^T \mathbf{X}_k^T \mathbf{Y}_k \mathbf{v}_k, \qquad (1)$$

which is subject to the constraints $G_k^{(\mathbf{v})}$ and $G_k^{(\mathbf{w})}$ (Höskuldsson, 1988):

$$G_k^{(\mathbf{v})} = \|\mathbf{v}_k\|_2^2 - 1 = 0 \quad G_k^{(\mathbf{w})} = \|\mathbf{w}_k\|_2^2 - 1 = 0, \qquad (2)$$

with $\|\circ\|_2^2$ being the squared norm of a vector. For determining subsequent weight and score vectors, PLS uses a deflation procedure, which involves the contribution of the k^{th} pair of score vectors to be subtracted or *deflated* from the predictor and response matrices, (Geladi and Kowalski, 1986):

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^T \qquad \mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k^T, \quad (3)$$

where $\mathbf{X}_k$, $\mathbf{Y}_k$ and $\mathbf{X}_{k+1}$, $\mathbf{Y}_{k+1}$ are the predictor and response matrices after $k+1$ and k deflation steps, respectively, and $\mathbf{p}_k$ and $\mathbf{q}_k$ are regression or loading vectors that represent the contribution of the t-score vector to the predictor and response matrices, respectively. The loading vectors are determined as follows:

$$\mathbf{p}_k = \frac{\mathbf{X}_k^T \mathbf{t}_k}{\mathbf{t}_k^T \mathbf{t}_k} \qquad \mathbf{q}_k = \frac{\mathbf{Y}_k^T \mathbf{t}_k}{\mathbf{t}_k^T \mathbf{t}_k}. \qquad (4)$$

After computing n pairs of weight, score and loading vectors, often denoted as latent variables (LVs), a parametric regression matrix, $\mathbf{B}_{PLS}^{(n)}$, can be calculated:

$$\mathbf{B}_{PLS}^{(n)} = \mathbf{W}_n \left[\mathbf{P}_n^T \mathbf{W}_n\right]^{-1} \mathbf{Q}_n^T, \qquad (5)$$

where $\mathbf{W}_n$, $\mathbf{P}_n$ and $\mathbf{Q}_n$ are matrices in which the n w-weight and p- and q-loading vectors are stored as column vectors in successive order.

The following properties of PLS are of importance in subsequent sections.

Remark 1. The vectors $\mathbf{v}_k$, $\mathbf{q}_k$ and $\mathbf{p}_k$ are functions of $\mathbf{w}_k$

Remark 2. The vectors $\mathbf{v}_k$ and $\mathbf{q}_k$ point to the same direction in the response space, i.e. $\mathbf{v}_k \propto \mathbf{q}_k$

Remark 3. The elements of each pair of score vectors, $\mathbf{t}_k$ and $\mathbf{u}_k$, are determined to have maximum covariance.

It follows from Remark 1 that the w-weight vector completely characterizes PLS, i.e. using $\mathbf{w}_k$, the score, loading and the v-weight vector can be computed.

3. NONLINEAR PARTIAL LEAST SQUARES

Various nonlinear extensions of PLS have been proposed that rely on a nonlinear mapping, where the u-scores are predicted using a nonlinear function of the t-scores:

$$\mathbf{u}_k = f\left(\mathbf{t}_k\right) + \mathbf{e}_k, \qquad (6)$$

where $\mathbf{e}_k$ is the residual vector of $\mathbf{u}_k$. The following functions, $f(\circ)$, were introduced: second order polynomials (Wold *et al.*, 1989; Baffi *et al.*, 1999*a*), smoothing procedures (Frank, 1990), spline functions (Wold, 1992), artificial neural networks (Qin and McAvoy, 1992; Holcomb and Morari, 1992; Wilson *et al.*, 1997; Baffi *et al.*, 1999*b*) and Box-Tidwell transformations (Li *et al.*, 2001).

It follows from Remark 1 that the algorithms by (Wold *et al.*, 1989; Baffi *et al.*, 1999*a*; Baffi *et* *al.*, 1999*b*) exhibit a close relation to PLS. More precisely, $\mathbf{t}_k$, $\mathbf{v}_k$, $\mathbf{u}_k$, $\mathbf{p}_k$, $\mathbf{q}_k$ are linear or nonlinear functions of $\mathbf{w}_k$. The update procedure of the i^{th} iteration step for the w-weight vector, $\mathbf{w}_{i,k}$, is given by (Baffi *et al.*, 1999*a*):

$$\begin{aligned}\Delta\mathbf{w}_{i+1,k} &= \left[\mathbf{Z}_{i,k}^T\mathbf{Z}_{i,k}\right]^\dagger \mathbf{Z}_{i,k}^T\mathbf{e}_{i,k}\\ \mathbf{w}_{i+1,k} &= \frac{\mathbf{w}_{i,k}+\Delta\mathbf{w}_{i+1,k}}{\|\mathbf{w}_{i,k}+\Delta\mathbf{w}_{i+1,k}\|_2},\end{aligned} \tag{7}$$

where † represents the generalized inverse of a matrix and $\mathbf{Z}_{i,k}$ is a matrix in which the nonlinear function of the k^{th} t-scores are stored. For example, the j^{th} row of this matrix has the following elements for a second order polynomial, $u_{i,jk} = a_{i,1} + a_{i,2}\cdot t_{i,jk} + a_{i,3}\cdot t_{i,jk}^2 + e_{i,jk}$, $\mathbf{z}_{i,jk}^T = \left(a_{i,2}\cdot\mathbf{x}_{jk}^T \;\; 2\cdot a_{i,3}\cdot\mathbf{x}_{jk}^T\right)$, where $a_{i,1}$, $a_{i,2}$ and $a_{i,3}$ are coefficients that are determined using a standards least squares procedure for instance and $\mathbf{x}_{jk}^T$ is the j^{th} row vector of $\mathbf{X}_k$.

The algorithm by (Baffi *et al.*, 1999*a*) is based on the work by (Wold *et al.*, 1989). More precisely, (Baffi *et al.*, 1999*a*) introduced some simplifications to enhance the algorithm by (Wold *et al.*, 1989). The simplified algorithm was later extended to include the application of neural networks (Baffi *et al.*, 1999*b*). The error-based algorithm by (Baffi *et al.*, 1999*a*; Baffi *et al.*, 1999*b*) is analyzed in the next sections.

4. ANALYSIS OF ERROR-BASED NONLINEAR PLS

Since updating $\mathbf{w}_{i+1,k} = \frac{\mathbf{w}_{i,k}+\Delta\mathbf{w}_{i+1,k}}{\|\mathbf{w}_{i,k}+\Delta\mathbf{w}_{i+1,k}\|_2}$ is based on minimizing the residuals of the k^{th} score model, $\mathbf{e}_{i,k}$, it is not guaranteed that the elements in each pair of score vectors are determined to have maximum covariance after converged. This can be seen by analyzing the cost function with which the w-weight vector is determined:

$$\begin{aligned}J_k^{(\mathbf{w})} &= \min_{\mathbf{w}_k}\|\mathbf{e}_k\|_2^2 = \min_{\mathbf{w}_k}\|\mathbf{u}_k - f(\mathbf{t}_k)\|_2^2\\ &= \min_{\mathbf{w}_k}\|\mathbf{Y}_k\mathbf{v}_k - f(\mathbf{X}_k\mathbf{w}_k)\|_2^2,\end{aligned} \tag{8}$$

which is subject to the following constraints:

$$\begin{aligned}\mathbf{G}_k^{(\mathbf{v})} &= \mathbf{v}_k - \frac{\mathbf{Y}_k^T f(\mathbf{X}_k\mathbf{w}_k)}{\left\|\mathbf{Y}_k^T f(\mathbf{X}_k\mathbf{w}_k)\right\|_2} = \mathbf{0}\\ G_k^{(\mathbf{w})} &= \|\mathbf{w}_k\|_2^2 - 1 = 0.\end{aligned} \tag{9}$$

To simplify the analysis of the above error-based algorithm, the function $f(\mathbf{t}_k)$ is assumed to be linear. This allows a strict comparison between conventional PLS and "linear" error-based PLS.

Using a linear function gives rise to the following Theorem, proven in Appendix A.

Theorem 1. If a linear function of the t-scores is used to predict the u-scores, i.e. $\mathbf{u}_k = a_k\mathbf{t}_k + \mathbf{e}_k$ with a_k being a regression coefficient, the weight vector $\mathbf{w}_k$ is the most dominant eigenvector of the matrix expression $\left[\mathbf{X}_k^T\mathbf{X}_k\right]^\dagger \mathbf{X}_k^T\mathbf{Y}_k\mathbf{Y}_k^T\mathbf{X}_k$.

The solution for obtaining $\mathbf{w}_k$, however, is equivalent to the solution of the RRR cost function, which gives rise to the following Conclusion.

Conclusion 1. If a linear relationship between the score variables is considered, the error-based iteration procedure by (Baffi *et al.*, 1999*a*; Baffi *et al.*, 1999*b*) produces, in fact, a RRR solution.

For completeness, the cost function of RRR and its solution are provided in Appendix B. Conclusion 1 can be reformulated as follows.

Conclusion 2. Since the error-based iteration procedure converges to produce a RRR solution, it is not guaranteed that the elements in the score vectors, $\mathbf{t}_k$ and $\mathbf{u}_k$ describe a maximum covariance.

An examples to support Conclusion 2 is given in Section 6. The equivalence between a linear version of error-based PLS and RRR follows from the v-weight vector being a function of the w-weight vector.

Theorem 2. By incorporating the constraint $\mathbf{G}_k^{(\mathbf{v})} = \mathbf{v}_k - \frac{\mathbf{Y}_k\widehat{\mathbf{u}}_k}{\|\mathbf{Y}_k\widehat{\mathbf{u}}_k\|_2} = \mathbf{v}_k - \frac{\mathbf{Y}_k\mathbf{t}_k}{\|\mathbf{Y}_k\mathbf{t}_k\|_2} = \mathbf{0}$, the error-based cost function minimizes the residuals of the response variables instead of minimizing the residuals of the k^{th} score variables.

The proof of Theorem 1 can also be applied to prove Theorem 2. Note that the error-based procedure complies with Bellman's principle of optimality (Bellman, 1957) for predicting the response variables.

By applying a nonlinear mapping between the score variables, as shown in Equation (6), the k^{th} u-score variable is predicted more accurately in case the underlying process exhibits nonlinear relationships between the predictor and response variables. On the basis of Theorem 2, this increased accuracy translates into an increased accuracy for predicting the response variables. This is, again, a result of the constraint applied to the v-weight vector, which gives rise to the following conclusion.

Conclusion 3. For nonlinear score models, the error-based iteration procedure converges to minimize the residual variance of the response variables.

Conclusion 3 can be argued on the basis of the following Lemmas.

Lemma 1. The constraint of the v-weight vector, used to calculated the u-score vector, is determined to maximize the contribution of the predicted u-score vector to the response matrix.

Lemma 2. After the iteration procedure has converged, the q-loading vector is obtained to maximize the contribution of the predicted u-score vector to the response variables and points in the same direction as the v-weight vector.

Lemma 3. The error-based iteration procedure converges to minimize the variance of the score model prediction and after conversion under the constraint of the v-weight vector being a function of the w-weight vector.

Lemma 1 and 2 are proven in Appendix C and Lemma 3 follows from Equation (9). With respect to PLS, Conclusions 2 and 3 give rise to postulate the following conclusion.

Conclusion 4. The error-based iteration procedure by (Baffi *et al.*, 1999*a*; Baffi *et al.*, 1999*b*) is a nonlinear extension to RRR rather than a nonlinear extension to conventional PLS.

The iterative error-based procedure is, in fact, a gradient descent approach for minimizing Equation (8) by incorporating the constraints of Equation (9). It should therefore be noted that the solution that is computed may be suboptimal, i.e. the solution may present a local minimum rather than a global minima. This might be circumvented by applying a Genetic Algorithm. Although computationally expensive this technique showed to provide more accurate NLRRR models, as shown by (Sharma *et al.*, 2003).

Another problem is that the series of iteration steps may not converge for each set of LVs. This, however, has not yet been reported and the applications of NLRRR in Section 6 also showed convergence. In contrast, the "linear" error-based iteration procedure converges to compute the most dominant eigenvector of a matrix expression. This, in turn, implies that (i) the results are optimal and (ii) convergence problems will usually not arise (Golub and van Loan, 1996).

5. APPLICATION STUDIES

In this section, the NLRRR algorithm is applied to a simulation study to illustrate the above findings. In addition, the covariance of the elements of each pair of t- and u-scores are determined for PLS and NLRRR.

The simulated process contained 4 predictor variables, denoted as x_1, x_2, x_3 and x_4, and 3 output variables, referred to as y_1, y_2 and y_3. The predictor variables were generated as follows:

$$\begin{aligned} x_1 &= 0.821 \cdot t_1 + 0.444 \cdot t_2 \\ x_2 &= 0.615 \cdot t_1 + 0.744 \cdot t_2 \\ x_3 &= 0.921 \cdot t_1 + 0.738 \cdot t_2 \\ x_4 &= 0.176 \cdot t_1 + 0.405 \cdot t_2, \end{aligned} \tag{10}$$

where t_1 and t_2 were normally distributed sequences of zero mean and unit variance, i.e. $t_1, t_2 \in \mathcal{N}\{0, 1\}$. Using the 4 predictor variables, the response variables were then computed to be:

$$\begin{aligned} y_1 &= \exp\left(2 \cdot x_1 \cdot \sin\left(\pi \cdot x_4\right)\right) + \sin\left(x_2 \cdot x_3\right) \\ y_2 &= x_1^2 + \sin\left(\pi \cdot x_2 \cdot x_3\right) + x_4^4 \\ y_3 &= \exp\left(x_2 \cdot \cos\left(\pi \cdot x_4\right)\right) + x_1 \cdot x_4. \end{aligned} \tag{11}$$

A data set of 1000 samples was obtained as described above and an identically and independently distributed sequence of zero mean and variance 0.01, was superimposed on each variable to represent measurement noise. A division of the recorded data set was then carried out as follows. The first 800 samples were used as reference data to identify a NLRRR model and the remaining 200 samples were used to test the identified model.

Prior to the identification of the NLRRR model, the variables of the reference data set were normalized, i.e. mean centered and scaled to unit variance. The nonlinear function was selected to be second order polynomials. To highlight that the NLRRR algorithm may not calculate pairs of t- and u-scores that have a maximum covariance, each pair of score vectors was evaluated for PLS and NLRRR. The results of this comparison are summarized in Table (1).

Table 1. Variance captured by score models for PLS and NLRRR

#LV	PLS Model	NLRRR Model
1	2.1083	0.1614
2	0.2180	0.0319
3	0.0003	0.0002
4	0.0000	0.0000

It could clearly be seen from Table (1), that the PLS model determined pairs of t- and u-scores that described a significantly larger covariance than those computed by the NLRRR model. The model performance of the NLRRR model was evaluated on the reference data using the percentage contribution of each set of LVs and their cumulative percentage contribution captured in the predictor and response matrices. Furthermore, the mean squared prediction error (MSPE) were also analyzed. Table (2) summarize the performance of the identified NLRRR model.

Table 2. Performance of NLRRR model

#LV	Contr to $\mathbf{X}$	Tot Contr to $\mathbf{X}$	Contr to $\mathbf{Y}$	Tot Contr to $\mathbf{Y}$	MSPE
1	88.38	88.38	42.86	42.86	1.7121
2	11.50	99.88	8.09	50.95	1.4698
3	0.07	99.95	0.17	51.12	1.4648
4	0.05	100.00	0.08	51.20	1.4626

Given the construction of the predictor variables, the predictor matrix without the introduction of measurement noise has rank 2. Thus, only two sets of LVs were required to represent the structure between the predictor and response variables. This could also be revealed by the NLRRR models as their variance contribution of the 3^{rd} and 4^{th} set of LVs was insignificant for (i) predicting the response matrix and (ii) reconstructing the predictor matrix. Hence, only two score models were required to represented the variation of the predictor and response variables, respectively.

6. CONCLUSIONS

In this paper, nonlinear extensions of PLS, introduced by (Wold *et al.*, 1989) and (Baffi *et al.*, 1999*a*), were analyzed. This analysis revealed, firstly, that these algorithms, in fact, determine a linear RRR model if the nonlinear functions, incorporated in the score models, are replaced by linear functions. Secondly, these algorithms should therefore be seen as nonlinear extensions of RRR, instead of nonlinear extensions to PLS. Thirdly, the weight and loading vectors of the response variables point in the same direction.

The analysis yielded further that the algorithm by (Baffi *et al.*, 1999*a*) are based on a gradient descent technique and may therefore produce a sub-optimal solution. In addition, the series of iteration steps may not converge for each set of LVs. Convergence problems, however, have not yet been reported and were also not experienced in this work. To alleviate the determination of an optimal solution for the associated cost functions and their constraints, (Sharma *et al.*, 2003) introduced the application of a Genetic Algorithm strategies.

The above findings were demonstrated using an application studies to a synthetic example.

Appendix A. LINEAR ERROR-BASED PARTIAL LEAST SQUARES

For simplicity, the subscripts on matrices and vectors, that indicate which LV is currently being determined, are omitted. For the i^{th} iteration step, the dependency of the weight vector $\mathbf{v}_i$, the regression coefficient b_i and the residual vector $\mathbf{e}_i$ upon the weight vector $\mathbf{w}_i$ is as follows:

$$\mathbf{v}_i = \frac{\mathbf{Y}^T\mathbf{X}\mathbf{w}_i}{\|\mathbf{Y}^T\mathbf{X}\mathbf{w}_i\|_2} \tag{A.1}$$
$$b_i = \frac{\mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i}{\mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i\,\|\mathbf{Y}^T\mathbf{X}\mathbf{w}_i\|_2}$$
$$\mathbf{e}_i = \frac{\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i}{\|\mathbf{Y}^T\mathbf{X}\mathbf{w}_i\|_2} - \mathbf{X}\mathbf{w}_i\frac{\mathbf{w}_i^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i}{\mathbf{w}_i^T\mathbf{X}^T\mathbf{X}\mathbf{w}_i\,\|\mathbf{Y}^T\mathbf{X}\mathbf{w}_i\|_2},$$

which implies that the update of the weight vector $\mathbf{w}_i$ is given by:

$$\Delta\mathbf{w}_{i+1} = \left[\mathbf{X}^T\mathbf{X}\right]^\dagger \frac{\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i\,\|\mathbf{X}\mathbf{w}_i\|_2^2}{\|\mathbf{Y}^T\mathbf{X}\mathbf{w}_i\|_2^2} - \mathbf{w}_i. \tag{A.2}$$

Consequently, the w-weight vector for the $(i+1)^{th}$ iteration step is calculated as:

$$\mathbf{w}_{i+1} = \frac{\left[\mathbf{X}^T\mathbf{X}\right]^\dagger\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i}{\left\|\left[\mathbf{X}^T\mathbf{X}\right]^\dagger\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}_i\right\|_2}. \tag{A.3}$$

The above iteration is equivalent to the Power method (Golub and van Loan, 1996) for determining the most dominant eigenvector of the positive definite or semi-definite matrix $\left[\mathbf{X}^T\mathbf{X}\right]^\dagger\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$. On convergence, the most dominant eigenvalue is equal to $\mathbf{w}^T\left[\mathbf{X}^T\mathbf{X}\right]^\dagger\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}$.

Appendix B. COST FUNCTION AND SOLUTION FOR REDUCED RANK REGRESSION

The residuals of the response matrix are given by:

$$\mathbf{E} = \mathbf{Y} - \mathbf{t}\mathbf{t}^T\mathbf{Y}, \tag{B.1}$$

which is based on the following constraints:

$$\mathbf{G}^{(\mathbf{t})} = \mathbf{t} - \mathbf{X}\mathbf{w} = \mathbf{0} \qquad G^{(\mathbf{w})} = \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 1 = 0 \tag{B.2}$$

Using the above equations, the RRR cost function for $\mathbf{w}$ is:

$$\begin{aligned} J^{(\mathbf{w})} &= \min_{\mathbf{w}} \text{trace}\left\{\mathbf{E}^T\mathbf{E}\right\} \\ &= \min_{\mathbf{w}} \text{trace}\left\{\mathbf{Y}^T\mathbf{Y}\right\} - \text{trace}\left\{\mathbf{Y}^T\mathbf{t}\mathbf{t}^T\mathbf{Y}\right\}. \end{aligned} \tag{B.3}$$

The solution of the above cost function can be obtained as follows:

$$\frac{\partial J^{(\mathbf{w})}}{\partial \mathbf{t}}\frac{\partial \mathbf{t}}{\partial \mathbf{w}} - \lambda\frac{\partial G^{(\mathbf{w})}}{\partial \mathbf{w}} = \mathbf{0} \tag{B.4}$$

and is given by:

$$\begin{aligned} 2\mathbf{w}^T\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X} - \lambda 2\mathbf{w}^T\mathbf{X}^T\mathbf{X} &= \mathbf{0} \\ \lambda\mathbf{w} &= \left[\mathbf{X}^T\mathbf{X}\right]^\dagger\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w}. \end{aligned} \tag{B.5}$$

Consequently, $\mathbf{w}$ is the eigenvector associated with the largest eigenvalue of $\left[\mathbf{X}^T\mathbf{X}\right]^\dagger\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}$.

Appendix C. DETERMINATION OF THE V-WEIGHT AND Q-LOADING VECTOR

The response matrix can be predicted using the prediction of the u-score vector, $f(\mathbf{t})$, and the loading vector $\mathbf{q}$:

$$\mathbf{Y} = f(\mathbf{t})\,\mathbf{q}^T + \mathbf{E}, \tag{C.1}$$

where $\mathbf{E}$ is a residual matrix. Given the case that $f(\mathbf{t})$ is predetermined, $\mathbf{q}^T$ can be regarded as an independent vector and the minimum variance of $\mathbf{E}$ is given by:

$$\mathbf{q}^T = \left(f(\mathbf{t})^T f(\mathbf{t})\right)^{-1} f(\mathbf{t})^T\,\mathbf{Y}, \tag{C.2}$$

which is equal to:

$$\mathbf{q} = \frac{\mathbf{Y}^T f(\mathbf{t})}{f(\mathbf{t})^T f(\mathbf{t})} \propto \mathbf{Y}^T f(\mathbf{t}), \tag{C.3}$$

as $f(\mathbf{t})^T f(\mathbf{t})$ is a scalar.The v-weight vector is also proportional to $\mathbf{Y}^T f(\mathbf{t})$, which implies that (i) the v-weight and the q-loading vector lie in the same direction and (ii) both are obtained to minimize the variance of the residual matrix $\mathbf{E}$.

REFERENCES

Baffi, G., E. B. Martin and A. J. Morris (1999*a*). Non-linear projection to latent structures revisited: the quadratic pls algorithm. *Computers and Chemical Engineering* **23**, 395–411.

Baffi, G., E. B. Martin and A. J. Morris (1999*b*). Non-linear projection to latent structures revisited: the neural network pls algorithm. *Computers and Chemical Engineering* **23**(9), 1293–1307.

Bellman, R. (1957). *Dynamic Programming.* Princeton University Press. Priceton, N.J., U.S.A.

Berglund, A. and S. Wold (1997). Inlr, implicit non-linear latent variable regression. *Journal of Chemometrics* **11**, 141–156.

Frank, I. E. (1990). A nonlinear pls model. *Chemometrics and Intelligent Laboratory Systems* **8**, 109–119.

Geladi, P. and B. R. Kowalski (1986). Partial least squares regression: A tutorial. *Analytica Chimica Acta* **185**, 231–246.

Golub, G. H. and C. F. van Loan (1996). *Matrix Computation.* 3 ed.. John Hopkins. Baltimore.

Hiden, H., B. McKay, M. Willis and G. Montague (1998). Non-linear partial least squares using genetic programming. In: *Proceedings of the 2^{nd} Annual Conference on Genetic Programming (GP'98).* Madison, Wisconsin, U.S.A.

Holcomb, T. R. and M. Morari (1992). Pls/neural networks. *Computers and Chemical Engineering* **16**(4), 393–411.

Höskuldsson, A. (1988). Pls regression models. *Journal of Chemometrics* **2**, 211–228.

Li, B., E. B. Martin and A. J. Morris (2001). Box-tidwell transformation based partial least squares regression. *Computers and Chemical Engineering* **25**, 1219–1233.

Patwardhan, R. and S. Lakshminarayan (1998). Constraint nonlinear mpc using hammerstein and wiener models. *AIChE Journal* **44**(7), 1611–1622.

Qin, S. J. and T. J. McAvoy (1992). Nonlinear pls modelling using neural networks. *Computers and Chemical Engineering* **16**(4), 379–391.

Sharma, S., U. Kruger and G. W. Irwin (2003). Genetic learning methods for enhanced nonlinear partial least squares. In: *Proceedings of the IFAC International Conference on Intelligent Control Systems and Signal Processing (ICONS2003).* pp. 37–42. Faro, Portugal.

Wilson, D. J. H., G. W. Irwin and G. Lightbody (1997). Nonlinear pls modelling using radial basis functions. In: *Proceedings of the American Control Conference.* Albuquerque, New Mexico.

Wold, S. (1992). Non-linear partial least squares modelling. ii spline inner functions. *Chemometrics and Intelligent Laboratory Systems* **14**, 71–84.

Wold, S., N. Kettaneh-Wold and B Skagerberg (1989). Non-linear pls modelling. *Chemometrics and Intelligent Laboratory Systems* **7**, 53–65.