An overview of some non-linear techniques in Chemometrics

Marlene Mörtsell, Mid-Sweden University
Mårten Gulliksson, Mid-Sweden University

Rapportserie FSCN - ISSN 1650-5387 2001:6
FSCN rapport R-01-19
November, 2001
Content

Abstract ........................................................................................................................... 3

1. Introduction ............................................................................................................ 4

2. Linear PLS and PCA ............................................................................................ 5

3. Non-linear PCA methods ....................................................................................... 9

4. Transformation of data ........................................................................................ 11
   4.1 Implicit non-linear latent variable regression – INLR ................................... 11
   4.2 GIFI – PLS ..................................................................................................... 12

5. Non-linear PLS with neural networks ................................................................ 14
   5.1 Non-linear PLS with integrated neural networks ........................................... 14
   5.2 A coupled neural networks – PLS system ...................................................... 14

6. Non-linear PLS without neural networks ............................................................ 15
   6.1 Quadratic PLS – QPLS ................................................................................... 15
   6.2 Spline PLS – SPLPLS .................................................................................... 16

7. Conclusions ........................................................................................................... 16

Appendix A .................................................................................................................... 18

Appendix B .................................................................................................................... 19

Appendix C .................................................................................................................... 21

References ..................................................................................................................... 23
An overview of some non-linear techniques in Chemometrics

Marlene Mörtsell and Mårten Gulliksson
Mid-Sweden University, Sundsvall, Sweden

Abstract

Some of the most common linear methods within chemometry are Principal Component Analysis (PCA) and Partial Least Squares (PLS). PCA are most suitable for data overview while PLS is suitable for quantitative modelling and prediction. However, non-linear relationships are common in many chemicals and practical situations, for example, non-linear relationships appear during the consumption of reactants in a chemical reaction. For non-linear data it is mostly not suitable to use linear methods like PCA and PLS. To handle non-linear data a number of non-linear methods within the PLS and PCA approaches have been presented. The simplest way is to include non-linear terms or transform the data and then use the original PLS algorithms. More complicated approaches have a non-linear relation included in the traditional PLS algorithm, e.g., splines and neural networks.

Many different techniques can be used to solve different kinds of non-linear problems. The fact that there are many different non-linear methods indicates that non-linearity can not generally be handled in a simple way.
1. Introduction

Some of the most common and linear methods within chemometry are Principal Component Analysis (PCA) and Partial Least Squares (PLS). PCA are most suitable for data overview while PLS is suitable for quantitative modelling and prediction. The fundamental difference between PCA and PLS are that PCA captures maximum variance in the data, $X$, when PLS finds directions in predictor variables, $X$, and the responses, $Y$, corresponding to maximum covariance. The basic idea with both methods is to transfer the data into a new co-ordinate system.

Non-linear relationships are common in many chemical and practical situations. For example, non-linear relationships appear during the consumption of reactants in a chemical reaction. The non-linearities may have different degrees; mild, moderate and strong non-linearity, schematically presented by Lennart Eriksson et al. [13], see Figure 1. Especially for moderate and strong non-linearities it is not suitable to use linear methods like PCA and PLS.

![Figure 1. Different degrees of non-linearity. Mild non-linearity (left), Moderate non-linearity (middle), Strong non-linearity (right).](image)

The simplest way to overcome the problem with non-linearity is to transform the data and include non-linear terms, e.g. square terms to the datamatrix, and then use the original PLS algorithm. However, in many cases it is not possible to pretreat the data because the non-linearity is not known. The method with adding a square term is called the implicit non-linear PLS modelling (INRL) and has been presented by Anders Berglund and Svante Wold [11]. Another simple method to handle non-linearities is the GIFI approach introduced by Michailidis and de Leeuw (1998) and modified by Anders Berglund et al. [14].

Several attempts have been made to integrate non-linear features within the linear PLS algorithm in order to produce a non-linear PLS algorithm. Wold et al. [20] proposed a quadratic PLS algorithm and the algorithm led to a variety of non-linear algorithms presented by G. Baffi et al. [21]. Qin and McAvoy [16] proposed a neural network algorithm, which uses a sigmoidal activation function neural network to fit the inner mapping and G. Baffi et al. [10] introduced a neural networks PLS algorithm. Wold [22] introduced a spline-PLS algorithm where a spline function is used to fit the non-linear mapping between each pair of the latent variables. A PLS-algorithm can also be used for dynamic modelling, see G. Baffi et al. [1] for the non-linear dynamics PLS algorithm.
This report will introduce some methods and pretreatments for non-linear data within chemometry and all are, in some way, related to PLS or PCA.

2. Linear PLS and PCA

Linear PLS and PCA are some of the most common projection methods in chemometrics and multivariate data analysis (MVDA). In the case of PCA, see Figure 2, there is only one data matrix, $X$, to consider and the method captures maximum variance direction in $X$. The key idea is to approximate the original matrix data by a product of three matrices, the score, the loading and the residual matrix. The main variance is presented in the score and loading matrices and the left over is presented in the residual matrix. The score matrix represent the observations while the loading matrix represent the variables. A new co-ordinate system is presented with the loadings and the scores.

$X = \bar{X} + TP + E$

**Figure 2.** The principals of PCA. $T$ – score matrix, $P$ – loading matrix, $E$ – residual matrix, $\bar{X}$ – mean value of the columns, $1$ – a vector of ones.

In the PLS approach [2], the NIPALS algorithm, see appendix A, the predictor variables, $X$, and the responses, $Y$, are decomposed in a similar way as in PCA. One fundamental difference is that while PCA captures maximum variance direction in $X$, PLS finds direction in $X$ and $Y$ corresponding to maximum covariance. The diagonal matrix $B$ represents the inner relation between $X$ and $Y$ and gives the possibility to correlate these both matrices, see Figure 3.
Figure 3. The principals of PLS. $T$, $U$ – score matrices, $P$, $Q$ – loading matrices, $E$, $F$ – residual matrix, $B$ – diagonal matrix, represent the inner relation between $X$ and $Y$, $\bar{x}$, $\bar{y}$ – mean value of the columns of $X$ and $Y$, $1$ – a vector of ones.

The analysis of experimental and process data consist of three primary stages [3], see Figure 4:

- evaluation of the raw data,
- model derivation and interpretation,
- model validation and use.

The aim of the first stage, evaluation of the raw data, is to focus on the re-shaping of the data into a format better suited for multivariate data analysis. This includes removing outliers and transform shew variables, i.e. variables with other distributions than the normal distributions. If needed, data may also be scaled and mean-centred.

The second stage, model derivation and interpretation, consists of the calculation of the PCA and PLS models. PCA is used for data overview, e.g. for detection outliers, groups, and trends among observations and also for evaluating relationships among variables and among variables and observations. In both kinds of modelling, it is important to understand, the actual real-world process (chemical, biological or physical).
In order to detect the number of components, e.g. the number of scores and loadings, it is common to use cross-validation. The cross-validation is an internal validation method and the number of components, which gives the best prediction given by cross-validation gives the number of components. For detection of strong outliers and observations groups score plots, T, are used and weakly deviating observations can be identified by the residuals to the model, normally called distance to model. One of the most used plots in the PLS modelling is the plot that shows the score from X and Y, the diagonal matrix B, see Figure 3. This plot offers a visualisation of the correlation structure between X and Y called the inner relation. A typical inner relation may look like a straight line when there is a linear relation between X and Y. Loading plots, U, and coefficient plots are used for detection of important variables. Loading plots can also give information of the variable correlation.

In the third stage, model validation and use, it is important to validate the model and investigate its predictive power for future outcomes. To find the optimal model it is important to use the richness of diagnostic tools and model parameters, i.e. scores, loadings, weights, residuals, cross-validation, leverage and external test sets, etc. The external test set, not used during the modelling, tests the predictive power for the PLS model and can show if the model is good or bad.

The multivariate data analysis is an iterative process because if the model is not acceptable after the evaluation with the internal and external test set it is possible to restart with the preparation of the data, see Figure 4.
1.1 Get acquainted
Univariate statistics
Distributions: Histogram, min-max, skewness
PCA for overview: X and/or Y

1.2 Prepare data
Transform, select observations, classes, variables (X, Y, exclude), scale and lag variables for time series modelling

2.1 Compute model
Compute with cross-validation, PCA, PLS

2.2 Evaluate model
Model summary (R², Q²)
Observation plots, check outliers;
   Inner relationship (t₁/u₁,…)
Scores
Residuals, check outliers;
   Distance to model
   Normal probability plot
Contribution plots;
   Scores
Variable plots;
   Loading
   Coefficient plot
Observed/Predicted for Y
Validate PLS by permutation

3.1 Accept model?
Yes
No

3.2 Validate model
Validate by external test set

3.3 Use model

Figure 4. Flowchart for MVDA, the three main steps.
3. **Non-linear PCA methods**

One of the first non-linear PCA approaches was to use an auto associative neural network, developed by Kramer [4,6]. The neural network introduced by Kramer has a five-layer structure, see Figure 5. The network inputs use a “bottleneck layer”, a layer containing fewer nodes than the output layer, as the representation of the score vector. The score vector represents the connection to the PCA approach, as in linear PCA the score vectors give a new co-ordinate system. The effect of this bottleneck layer is to force the network to develop a compact representation of the input data. Kramer presented results for a simulated batch reactor in which four first order reactions were occurring simultaneously and the successful identification of a number of faults was demonstrated.

Another example of a non-linear technique is based on genetic programming [5], see Figure 6. Observe that the structure is close to the method that Kramer proposed. In this case the linearising vectors are based on genetic programming, via the operators direct reproduction, mutation or crossover. The structure between the linearising vectors represent the score matrix. In the same way as for the approach presented above the score vector represent the PCA approach. The reconstruction of $\mathbf{X}$ from the output layer is compared with the original $\mathbf{X}$ for error calculation.

![Figure 5. Layout of non-linear PCA, the auto associative neural network; Input layer, mapping layer, bottleneck layer (the score vector), de-mapping layer and output layer. $\sigma$ non-linear node, * linear node](image-url)
A third non-linear PCA method is based on an input-training neural network [6], see Figure 7. The input-training neural is not fixed but adjusted along with the weights to reproduce a corresponding output. The idea is based on that the scaled process observations, $X$, can be used as the output layer pattern and after the network has been trained the non-linear principal scores can be identified from the input layer.

Figure 6. Layout of non-linear PCA, genetic programming.

Figure 7. Layout of non-linear PCA, input-training neural network: Input – scores, Output – process observations, $w_1$ and $w_2$ – weight vectors.
4. Transformation of data

A simple method for modelling several non-linear relationships, is to replace the predictor variables, \( X \), or the responses, \( Y \), or both \( Y \) and \( X \) by non-linearly transformed variables or to add transformed variables [7]. The most common transformations of \( Y = Xb + f \) are, where \( f \) and \( f_i \) are residuals,

\[
g(Y) = Xb + f_1, \quad (1)
\]

\[
Y = H(X)b + f_2, \quad (2)
\]

\[
g(Y) = H(X)b + f_3, \quad (3)
\]

\[
Y = [XX^a]d + f_4, \quad (4)
\]

\[
g(Y) = [XX^a]d + f_5. \quad (5)
\]

In (1), \( Y \) is transformed non-linearly, usually by an exponential or logarithm. In (2), \( X \) is transformed in some non-linear fashion, usually by logarithm and (3) is a combination of the transformations in (1) and (2).

In Eq. (4), \( X \) is augmented with some exponent of itself. Observe that regression vector \( d \) is larger than \( b \) since the number of variables has been increased. The value of the exponent has to be found, but very often two and three are considered to be good starting points [8,9,10,12]. Finally, (5), (1) and (4) are combined.

In many cases, the best transformation, the smallest prediction error, is found through “trial and error” among the different transformation possibilities. A more sophisticated transformation is made with a neural network that is coupled together with PLS, presented in Section 6.2.

4.1 Implicit non-linear latent variable regression – INLR

Implicit non-linear latent variable regression (INLR) introduced by Anders Berglund and Svante Wold [11] is one of the simplest methods to handle non-linearity. The basic idea is to add terms to the predictor variables, \( X \), and the added terms to the matrix are for example square and cross terms. If square terms are added as

\[
B = \begin{bmatrix}
x_{11} & x_{12} & \ldots & x_{1n} \\
x_{21} & x_{22} & \ldots & x_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
x_{1n} & x_{2n} & \ldots & x_{nn}
\end{bmatrix},
\]

the variables are twice as many but if the cross terms also are added the variables will increase because the cross-product terms is always many more then the number of linear terms. It is also possible to add higher order terms to \( X \) to handle higher order non-linearities.

The models have the limitation that they might contain too many components to be fit for monitoring or analysing the system of interest. Since \( X \) is expanded with terms, the chemical interpretability of PLS can be distorted. If \( X \) contain much noise and irrelevant
information, this will limit the performance of INLR. The expansion gives more noise in $X$, and this can be minimised by pre-weighting the variables. The weights of the square and the cubic block are less than one and the INLR solution will converge to the linear PLS. This approach can be handled using hierarchical PLS. The basic idea with the hierarchical PLS is to use the score values from a firstly PLS model as observations in a secondly PLS model [12].

The INLR can not handle extreme non-linearities but the INLR models are robust because of the simplicity and rigidity. The INLR is more stable against outliers compared to neural networks.

### 4.2 GIFI – PLS

GIFI-PLS [13, 14] can handle problems where the degree of non-linearity is mild to moderate. The approach is based on “binning” of quantitative variables into categorical variables, see Figure 9. Each categorical variable is then expanded into a set of linked dummy variables, for example 1 or 0. These variables enable the modelling of non-linearity.

The GIFI procedure can be separated in four different steps:

1. The range of each variable is computed. The ranges are determined intuitively.
2. An appropriate number of bins are selected for each variable, and the data points are then binned according to the different intervals and recorded as a “qualitative” categorical variable.
3. All the ”qualitative” binned variables are expanded into sets of dummy variables, example 0 and 1, which are then used as variables in the PLS model.
4. In the interpretation step, different regression coefficient profiles will indicate various relationships between $X$ and $Y$.

The coding 0 and 1 is not always the most suitable one [19] and the number of bins and the GIFI coding varies between different datasets. For multivariate data with many and collinear variables, the binning of individual variables becomes tedious. The GIFI procedure is not suitable for industrial data, because they often consist of many variables and collinearities, the GIFI-PLS is mostly used in quantitative structure-activity relationship (QSAR) modelling.
Figure 9. Illustration of the three main steps involved in the “GIFI-ing” procedure. The range of each column, $x$, in $X$ is computed (Step 1). An appropriate number of bins are selected for each variable, and the data points are then binned according to the different intervals and re-coded as new “qualitative” categorical variables, $x_1$, $x_2$ and $x_3$ (Step 2). All the “qualitative” binned variables are expanded into sets of dummy variables, example 1 and 0, which are then used as new variables in the PLS model (Step 3).
5. **Non-linear PLS with neural networks**

One of the most popular non-linear techniques is neural networks. The main drawback with the neural networks is that they are considered as a black-box technique with several limitations, for example the handling of collinear data. Another technique to handle non-linearity are modified PLS algorithms with a non-linear relation within the linear PLS algorithm. The non-linear inner relation can be e.g. neural networks or splines. The main reason to use modified PLS algorithms compared to neural networks are that the modified algorithms can handle collinearities and outliers in the same way as the traditional linear PLS algorithm.

There are two common techniques that use neural networks together with the traditional PLS algorithm, the NIPALS algorithm. Firstly a sigmoid neural network NN or a radial basis function (RBF) network can be fully integrated within the PLS algorithm, secondly a neural network may be used to pretreat the data.

5.1 **Non-linear PLS with integrated neural networks**

One of the techniques that can handle non-linear data is an algorithm where a neural network (NN) is fully integrated within the PLS algorithm. G. Baffi et al. [15] has presented an error-based NNPLS algorithm, see appendix B, and an RBF-PLS algorithm. The error-based part in the error-based NNPLS algorithm is built in the same way as the error-based quadratic PLS algorithm, presented later in Section 7.1. The idea is to train the sigmodial neural networks between the scores from $X$ and $Y$, i.e. $t$ and $u$, and than calculate the non-linear score for $Y$, $u_n$, as.

\[ u_n = f(t,u). \]  

Some of the most important references for the neural network PLS algorithm are Qin and McAvoy [16] and Qin [17] and for the PLS-RBF algorithm the most common reference is Wilson et al. [18].

The drawback with building NNPLS and RBF-PLS models is the selection of the network structures and that the model can easily be overfitted. The main advantages of using the neural network PLS algorithm as opposed to a direct application of neural network are that the neural networks PLS algorithm can handle variable correlations and the data set dimensionality.

5.2 **A coupled neural networks – PLS system**

Greger Andersson et al. [19] coupled a neural network together with the linear used PLS regression to transform the responses, $Y$. The neural network was used as transformation function for the non-linear data, see Figure 10. The modelling was performed in two steps. Firstly, the optimal PLS dimensionality was determined. The PLS regression was calculated between $X$ and $Y$. The optimal dimensionality was developed by traditional cross-validation. Secondly, the linearising transform, i.e. the neural network, was optimised. Input to the neural network was $Y$ and the network is optimised to minimum prediction residual.

This technique can handle both moderate non-linearities. Leverage and unmodelled residuals can be calculated because the diagnostic characteristics of PLS are retained. The
chemical interpretability of PLS is preserved without distortion. This is often not the case if more PLS components are used or if $X$ is expended with square terms. The linearising transform, i.e. the neural network, is robust towards deviating objects, conversely to techniques which use non-linear inner relations. G. Andersson et al. [19] noticed that this approach had no tendency for overfitting in contrast to the quadratic PLS approach, presented in Section 7.1.

6. Non-linear PLS without neural networks

To handle non-linearity without neural networks one of the simplest way is to add terms, for example square and cross terms, to the predictor variables, $X$, or to the responses, $Y$, or both $Y$ and $X$. More advanced methods are modified PLS algorithms with a non-linear inner relation, using, e.g., splines.

There are mainly two techniques, a quadratic PLS algorithm (QPLS) or a multivariate adaptive regression spline PLS algorithm (SPLPLS).

6.1 Quadratic PLS – QPLS

The quadratic PLS algorithm was introduced by Wold et al. [20] and the algorithm has been modified by Baffi et al. [21]. The algorithm presented by Wold et al. is complicated and converges slowly when the data lack structures. There are three modified algorithms presented by Baffi et al.. The three modified PLS algorithms are still related to the original QPLS algorithm in the sense that they are based on the use of the Newton-Raphson method. One of the algorithm is an error based QPLS algorithm, see appendix C. The error based algorithm presented by Baffi et al. converges faster then the algorithm presented by Wold et al.

The non-linear PLS algorithm, presented by Baffi et al., modifies the linear inner relation between the predictor and the response latent variables to a non-linear relationship. Any continuous and differentiable function with respect to the weights and the parameters can be used to fit the inner relation. The relationship can be approximated by Newton-Raphson, i.e. a first order Taylor series expansion of the quadratic inner relationship.

The QPLS can handle highly non-linear data but the drawback of this technique is that the quadratic non-linear mapping might not be the most suitable functional representation. The power of the approach is that it can be generalised to any function that is continuous and differentiable.
The algorithm has been applied to industrial application where G. Baffi et al. [21] used the algorithm on data from an industrial fluidised-bed reactor. But the available QPLS algorithm is too unstable to be a multi-purpose tool.

### 6.2 Spline PLS – SPLPLS

Svante Wold [22] presented a spline PLS algorithm (SPLPLS). The algorithm has a spline inner relation (quadratic or cubic) within the traditional PLS algorithm. In the same way as the QPLS, the spline fitting is integrated within the linear PLS algorithm. From the linear NIPALS algorithm the scores from both $X$ and $Y$ are calculated. The scores are then used to calculate the spline function and after that, the non-linear score from $Y$ can be calculated. This approach can handle moderate non-linear data but as other non-linear methods, SPLPLS tends to overfit.

Another method is multivariate adaptive regression splines (mars) [23, 24] and it is one of the most complicated non-linear methods.

### 7. Conclusions

The fact that there are many different non-linear methods indicates that non-linearity can not be handled in a simple way. All methods presented in this report are seen in Figure 10. As always, certain techniques are more suitable to specific kinds of data. Two of the simplest methods are GIFI and INLR, and are suitable to use with few variables and collinearities, usually laboratory data sets. More complicated algorithms are modified PLS algorithms, e.g. quadratic PLS (QPLS) and neural network PLS (NNPLS) and these algorithms are better suited for industrial data. The available algorithms for QPLS, NNPLS and SPLPLS are too unstable to be standard tools.

In a forthcoming application to modelling an oxygen reactor for softwood pulp, the most useful approach could be to use the methodology where a neural network is coupled to PLS. Instead of using neural networks, which pretreats the predictor variables, $X$, or the responses, $Y$, or both $Y$ and $X$, some chemical-physical model would be used. The pretreated data could then be evaluated using the traditional PLS algorithm. One of the main reason for this approach is that it is possible to evaluate the models in blocks, but also that you can use the chemical knowledge when pretreating the data for PLS modelling.
Figure 10. Schematic presentation of the different non-linear approaches presented in this report.
Appendix A

NIPALS algorithm for linear PLS

<table>
<thead>
<tr>
<th>Step</th>
<th>Summary of step</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Mean centre and scale X and Y</td>
</tr>
<tr>
<td>1</td>
<td>Set the output scored u equal to a column of Y</td>
</tr>
<tr>
<td>2</td>
<td>Compute input weights w by regressing X on u. ( w^T = \frac{u^T X}{u^T u} )</td>
</tr>
<tr>
<td>3</td>
<td>Normalise w to unit length ( w = \frac{w}{|w|} )</td>
</tr>
<tr>
<td>4</td>
<td>Calculate the input scores t ( t = \frac{Xw}{w^T w} )</td>
</tr>
<tr>
<td>5</td>
<td>Compute output loading q by regressing Y on t ( q^T = \frac{t^T Y}{t^T t} )</td>
</tr>
<tr>
<td>6</td>
<td>Normalise q to unit length ( q = \frac{q}{|q|} )</td>
</tr>
<tr>
<td>7</td>
<td>Calculate new output scores u ( u = \frac{Yq}{q^T q} )</td>
</tr>
<tr>
<td>8</td>
<td>Check convergence on u: if yes go to 9 else go to 2</td>
</tr>
<tr>
<td>9</td>
<td>Calculate the input loading p by regressing X on t ( p^T = \frac{t^T X}{t^T t} )</td>
</tr>
<tr>
<td>10</td>
<td>Compute inner model regression coefficient ( b = \frac{t^T u}{t^T t} )</td>
</tr>
<tr>
<td>11</td>
<td>Calculate input residual matrix ( E = X - tp^T )</td>
</tr>
<tr>
<td>12</td>
<td>Calculate output residual matrix ( F = Y - bt \times q^T )</td>
</tr>
<tr>
<td>13</td>
<td>If additional PLS dimensions are necessary, replace X and Y by E and F, respectively and repeat steps 1 to 13.</td>
</tr>
</tbody>
</table>
Appendix B

Summary of the steps in centred sigmoid neural networks PLS, with error-based input weights updating procedure, Baffi et al.

<table>
<thead>
<tr>
<th>Step</th>
<th>Summary of step</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Mean centre and scale $X$ and $Y$</td>
</tr>
<tr>
<td>1</td>
<td>Set the output score $u$ equal to a column of $Y$</td>
</tr>
<tr>
<td>2</td>
<td>Compute input weights $w$ by regressing $X$ on $u$. $w^T = \frac{u^T X}{u^T u}$</td>
</tr>
<tr>
<td>3</td>
<td>Normalise $w$ to unit length $w = \frac{w}{|w|}$</td>
</tr>
<tr>
<td>4</td>
<td>Calculate the input scores $t$ $t = \frac{Xw}{w^T w}$</td>
</tr>
<tr>
<td>5</td>
<td>Train the centred sigmoid neural networks between $t$ and $u$ $(\omega_1, \omega_2, \beta_1, \beta_2) \leftarrow (t, u)$</td>
</tr>
<tr>
<td>6</td>
<td>Calculate the non-linear prediction of $u$ $\hat{u} = \omega_2 \sigma$ $(\omega_1 t + \beta_1) + \beta_2$</td>
</tr>
<tr>
<td>7</td>
<td>Regress the columns of $Y$ on $r$ $q^T = \frac{\hat{u}^T Y}{\hat{u}^T r}$</td>
</tr>
<tr>
<td>8</td>
<td>Normalise $q$ to unit length $q = \frac{q}{|q|}$</td>
</tr>
<tr>
<td>9</td>
<td>Calculate new output scores $u$ $u = \frac{Yq}{q^T q}$</td>
</tr>
<tr>
<td>10</td>
<td>Compute the input weights updating parameters $\Delta w$ as described in reference [8]</td>
</tr>
<tr>
<td>11</td>
<td>Compute new input weights $w$ $w = w + \Delta w$</td>
</tr>
<tr>
<td>12</td>
<td>Normalise $w$ to unit length $w = \frac{w}{|w|}$</td>
</tr>
<tr>
<td>13</td>
<td>Calculate new input scores $t$ $t = \frac{Xw}{w^T w}$</td>
</tr>
</tbody>
</table>
14 Check convergence on $t$: if yes go to 15
   else go to 5
15 Calculate the input loadings $p$ by regression $X$ on $t$
   \[ p^T = \frac{t^TX}{t^Tt} \]
16 Train the centred sigmoid neural networks between $t$ and $u$
   \[ (\omega_1, \omega_2, \beta_1, \beta_2) \leftarrow (t, u) \]
17 Calculate the non-linear prediction of $u$
   \[ \hat{u} = \omega_2 \sigma \]
   \[ (\omega_1 t + \beta_1) + \beta_2 \]
18 Calculate input residual matrix
   \[ E = X - tp^T \]
19 Calculate output residual matrix
   \[ F = Y - \hat{u}q^T \]
20 If additional PLS dimensions are necessary, replace $X$ and $Y$ by $E$ and $F$, respectively and repeat steps 1 to 20.
### Appendix C

Summary of the steps in non-linear quadratic PLS, Baffi et al.

<table>
<thead>
<tr>
<th>Step</th>
<th>Summary of step</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Mean centre and scale X and Y</td>
</tr>
<tr>
<td>1</td>
<td>Set the output scored u equal to a column of Y</td>
</tr>
<tr>
<td>2</td>
<td>Compute input weights w by regressing X on u. [ w = \frac{u^T X}{u^T u} ]</td>
</tr>
<tr>
<td>3</td>
<td>Normalise w to unit length [ w = \frac{w}{|w|} ]</td>
</tr>
<tr>
<td>4</td>
<td>Calculate the input scores t [ t = \frac{X w}{w^T w} ]</td>
</tr>
<tr>
<td>5</td>
<td>Fit the non-linear inner relation [ c \leftarrow \text{fit}[u = f(t) + h] ]</td>
</tr>
<tr>
<td>6</td>
<td>Calculate the non-linear prediction of u [ r = f(t, c) ]</td>
</tr>
<tr>
<td>7</td>
<td>Regress the columns of Y on r [ q = \frac{r^T Y}{r^T r} ]</td>
</tr>
<tr>
<td>8</td>
<td>Normalise q to unit length [ q = \frac{q}{|q|} ]</td>
</tr>
<tr>
<td>9</td>
<td>Calculate new output scores u [ u = \frac{Y q}{q^T q} ]</td>
</tr>
<tr>
<td>10</td>
<td>Update input weights w as described in reference [19]</td>
</tr>
<tr>
<td>11</td>
<td>Normalise w to unit length [ w = \frac{w}{|w|} ]</td>
</tr>
<tr>
<td>12</td>
<td>Calculate new input scores t [ t = \frac{X w}{w^T w} ]</td>
</tr>
<tr>
<td>13</td>
<td>Check convergence on t: if yes go to 14 else go to 5</td>
</tr>
<tr>
<td>14</td>
<td>Fit the non-linear inner relation [ c \leftarrow \text{fit}[u = f(t) + h] ]</td>
</tr>
<tr>
<td>15</td>
<td>Calculate new non-linear prediction of u [ r = f(t, c) ]</td>
</tr>
<tr>
<td>16</td>
<td>Calculate the X loadings [ p = \frac{t X}{t^T t} ]</td>
</tr>
</tbody>
</table>
17 Calculate input residual matrix \[ E = X - tp^T \]

18 Calculate output residual matrix \[ F = Y - rq^T \]

19 If additional PLS dimensions are necessary, replace \( X \) and \( Y \) by \( E \) and \( F \), respectively and repeat steps 1 to 19.
References


7. P. Geladi, L. Hadjiiski and P. Hopke, Multiple regression for environmental data: nonlinearities and prediction bias, Chemometrics and Intelligent Laboratory Systems, 47 (1999) 165-173


18 D.J.H. Wilson, G.W. Irwin and G. Lightbody, Non-linear PLS modelling using radial basis functions, American Control Conference, Albuquerque, New Mexico, June 4-6 (1997)


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


22 S. Wold, Non-linear partial least squares modelling II. Spline inner relation, Chemical and Intelligent Laboratory Systems, 14 (1992) 71-84