PARTIAL LEAST-SQUARES REGRESSION: A TUTORIAL

PAUL GELADI* and BRUCE R. KOWALSKI

Laboratory for Chemometrics and Center for Process Analytical Chemistry, Department of Chemistry, University of Washington, Seattle, WA 98195 (U.S.A.)

(Received 15th July 1985)

SUMMARY

A tutorial on the partial least-squares (PLS) regression method is provided. Weak points in some other regression methods are outlined and PLS is developed as a remedy for those weaknesses. An algorithm for a predictive PLS and some practical hints for its use are given.

The partial least-squares regression method (PLS) is gaining importance in many fields of chemistry; analytical, physical, clinical chemistry and industrial process control can benefit from the use of the method. The pioneering work in PLS was done in the late sixties by H. Wold in the field of econometrics. The use of the PLS method for chemical applications was pioneered by the groups of S. Wold and H. Martens in the late seventies after an initial application by Kowalski et al. [1]. In spite of the large amount of literature that emerged from these groups, most articles describing PLS give algorithms and theory that are incomplete and often difficult to understand. Two recent articles [2, 3] show that PLS is a good alternative to the more classical multiple linear regression and principal component regression methods because it is more robust. Robust means that the model parameters do not change very much when new calibration samples are taken from the total population.

This article is meant as a tutorial. The reader is referred to texts on linear algebra [4, 5] if needed. The two most complete articles on PLS available at present are by S. Wold et al. [4, 6]. The nomenclature used in Kowalski [6] will be used here. Furthermore, all vectors will be column vectors. The corresponding row vectors will be designated as transposed vectors. The notation will be kept as rigorous as possible. Table 1 lists the notation used. The paragraphs on multiple linear regression, principal component analysis and principal component regression are included because they are necessary for a good understanding of PLS. They do not represent a complete treatment of these subjects.

*Present address: Chemometrics Group, Department of Organic Chemistry, Umeå University, S-901 87 Umeå, Sweden.
TABLE 1

Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>a dummy index for counting samples (objects)</td>
</tr>
<tr>
<td>j</td>
<td>a dummy index for counting independent (x) variables</td>
</tr>
<tr>
<td>k</td>
<td>a dummy index for counting dependent (y) variables</td>
</tr>
<tr>
<td>h</td>
<td>a dummy index for counting components or factors</td>
</tr>
<tr>
<td>n</td>
<td>the number of samples in the calibration (training) set</td>
</tr>
<tr>
<td>m</td>
<td>the number of independent (x) variables</td>
</tr>
<tr>
<td>p</td>
<td>the number of dependent (y) variables</td>
</tr>
<tr>
<td>r</td>
<td>the number of factors used (≤ rank of X)</td>
</tr>
<tr>
<td>x</td>
<td>a column vector of features for the independent variables (size m x 1)</td>
</tr>
<tr>
<td>y</td>
<td>a column vector of features for the dependent variables (size p x 1)</td>
</tr>
<tr>
<td>X</td>
<td>a matrix of features for the independent variables (size n x m)</td>
</tr>
<tr>
<td>Y</td>
<td>a matrix of features for the dependent variables (size n x p)</td>
</tr>
<tr>
<td>b</td>
<td>a column vector of sensitivities for the MLR method (size m x 1)</td>
</tr>
<tr>
<td>B</td>
<td>a matrix of sensitivities for the MLR method (size m x n)</td>
</tr>
<tr>
<td>t_h</td>
<td>a column vector of scores for the X block, factor h (size n x 1)</td>
</tr>
<tr>
<td>p_h</td>
<td>a row vector of loadings for the X block, factor h (size 1 x m)</td>
</tr>
<tr>
<td>w_h</td>
<td>a row vector of weights for the X block, factor h (size 1 x m)</td>
</tr>
<tr>
<td>T</td>
<td>the matrix of X scores (size n x a)</td>
</tr>
<tr>
<td>P</td>
<td>the matrix of X loadings (size a x m)</td>
</tr>
<tr>
<td>u_h</td>
<td>a column vector of scores for the Y block, factor h (size n x 1)</td>
</tr>
<tr>
<td>q_h</td>
<td>a row vector of loadings for the Y block, factor h (size 1 x p)</td>
</tr>
<tr>
<td>U</td>
<td>the matrix of Y scores (size n x a)</td>
</tr>
<tr>
<td>Q</td>
<td>the matrix of Y loadings (size a x p)</td>
</tr>
<tr>
<td>M_h</td>
<td>a rank 1 matrix, outer product of t_h and p_h (size n x m)</td>
</tr>
<tr>
<td>E_h</td>
<td>the residual of X after subtraction of h components (size n x m)</td>
</tr>
<tr>
<td>F_h</td>
<td>the residual of Y after subtraction of h components (size n x p)</td>
</tr>
<tr>
<td>b_h</td>
<td>the regression coefficient for one PLS component</td>
</tr>
<tr>
<td>I_n</td>
<td>the identity matrix of size n x n</td>
</tr>
<tr>
<td>I_m</td>
<td>the identity matrix of size m x m</td>
</tr>
</tbody>
</table>

Mean-centering and scaling

Before the model is calibration set in order, the values for each average value for each subtracted from each variables, both dependents.

There are also differences in this dependent differently because they are essentially three with the three variables model. In the second block are measure in another divided by dividing all for that variable, so one can decide that can not influence the model.

Another text, all variables considered to be most.

MULTIPLE LINEAR REGRESSION

The multiple linear regression is the goal to establish a can be represented as

\[ y = b_1x_1 + b_2x_2 + b_3x_3 + \ldots + b_kx_k \]

Fig. 1. Data preprocessor and its center. (A) Most (C) The result after variable scaling.
Mean-centering and scaling of variables

Before the model is developed, it is convenient to tailor the data in the calibration set in order to make the calculations easier. For ease of explanation, the values for each variable are used in the mean-centered form. The average value for each variable is calculated from the calibration set and then subtracted from each corresponding variable. In the rest of the text, all variables, both dependent and independent, are assumed to be mean-centered.

There are also different ways of scaling the variables. It should be pointed out that the dependent variables and the independent ones can be scaled differently because the sensitivities absorb the differences in scaling. There are essentially three ways of treating variables. In one, no scaling is needed when all the variables in a block are measured in the same units, as in spectrometry. In the second, variance scaling is used when the variables in a block are measured in different units (e.g., ppm, %, km); scaling is accomplished by dividing all the values for a certain variable by the standard deviation for that variable, so that the variance for every variable is unity. Thirdly, one can decide that certain variables are of less importance and hence should not influence the model very much; so they are given a smaller weight.

An illustration of scaling and mean centering is given in Fig. 1. In the further text, all variables are assumed to have some type of scaling, whichever is considered to be most appropriate.

MULTIPLE LINEAR REGRESSION (MLR)

The multiple linear regression (MLR) problem can be stated as follows. Features are measured for \( m \) variables \( x_j (j = 1 \ldots m) \) and for a variable \( y \) with the goal to establish a linear (or first-order) relationship between them. This can be represented mathematically as

\[
y = b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots + b_m x_m + \epsilon
\]  

(1a)

![Fig. 1. Data preprocessing. The data for each variable are represented by a variance bar and its center. (A) Most raw data look like this. (B) The result after mean-centering only. (C) The result after variance-scaling only. (D) The result after mean-centering and variance-scaling.](image-url)
\[ y = \sum_{j=1}^{m} b_j x_j + e \quad (1b) \]

\[ y = x' b + e \quad (1c) \]

In Eqn. 1(a), the \( x_j \) are called independent variables and \( y \) is the dependent variable, the \( b_j \)'s are sensitivities and \( e \) is the error or residual. In Eqn. 1(c), \( y \) is a scalar, \( b \) is a column vector and \( x' \) is a row vector.

Equation 1 describes multilinear dependencies for only one sample. If one gets \( n \) samples, the \( y_i (i = 1 \rightarrow n) \) can be written as a column vector \( y \), \( b \) remains the same and the vectors, \( x'_j \), form the rows of a matrix \( X \):

\[ y = Xb + e \quad (2) \]

For a better understanding of these matrix equations, they are also given in graphical representation:

\[
\begin{bmatrix}
1 & \cdots & m
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_n
\end{bmatrix}
= 
\begin{bmatrix}
x_1' & \cdots & x_m'
\end{bmatrix}
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_m
\end{bmatrix}
+ 
\begin{bmatrix}
e_1 \\
e_2 \\
\vdots \\
e_n
\end{bmatrix}
\]

In this case, \( n \) is the number of samples and \( m \) the number of independent variables.

It is now possible to distinguish among three cases.

1. \( m > n \). There are more variables than samples. In this case, there is an infinite number of solutions for \( b \), which all fit the equation. This is not what is wanted.

2. \( m = n \). The numbers of samples and of variables are equal. This situation may not be encountered often in practical situations. However, it gives a unique solution for \( b \) provided that \( X \) has full rank. This allows us to write

\[ e = y - Xb = 0 \quad (3) \]

\( e \) is called the residual vector. In this case, it is a vector of zeroes: 0.

3. \( m < n \). There are more samples than variables. This does not allow an exact solution for \( b \). But one can get a solution by minimizing the length of the residual vector \( e \) in the following equation:

\[ e = y - Xb \quad (4) \]

The most popular method for doing this is called the "least-squares method". The least-squares solution is

\[ b = (X'X)^{-1}X'y \quad (5) \]

(Complete explanations are available elsewhere [5, 7, 8].) Equation 5 gives a hint towards the most frequent problem in MLR: the inverse of \( X'X \) may not exist. Collinearity, same problem. A good

At this point, it might samples as variables, bu of them is to delete so for choosing which var

Multiple linear regression

A popular misconception variable. This is the c most software package more dependent vari extention to more than

Suppose there are \( t \) simply write two MLR:

\[ y_1 = Xb_1 + e_1; y_2 = Xb_2 + e_2 \]

But one can then put \( y \) for \( b_1 \) and \( b_2 \) and \( e_1 \) an

\[ Y = XB + E \]

where \( Y = (y_1, y_2) \), \( B = \)

\[ \begin{bmatrix}
2-p \\
n \end{bmatrix} \]

This is the general case.

Summary: MLR

- For \( m > n \), there variables.
- For \( m = n \), there is c
- For \( m < n \), a least-
matrix inversion can c
- MLR is possible wit

PRINCIPAL COMPONENT

Principal components rank \( r \) as a sum of \( r m \).

\[ X = M_1 + M_2 + M_3 + \ldots \]
not exist. Collinearity, zero determinant and singularity are all names for the same problem. A good description of this situation is available [9].

At this point, it might appear that there always have to be at least as many samples as variables, but there are other ways to formulate this problem. One of them is to delete some variables in the case $m > n$. Many methods exist for choosing which variables to delete [7, 8].

**Multiple linear regression with more than one dependent variable**

A popular misconception is that MLR is only possible for one dependent variable. This is the case that is almost always found in textbooks. Also, most software packages run MLR in this way. It is easy to extend MLR for more dependent variables. The example given here is for two variables, but extension to more than two is straightforward.

Suppose there are two dependent variables, $y_1$ and $y_2$. In this case, one can simply write two MLR's and find two vectors of sensitivities, $b_1$ and $b_2$:

$$y_1 = Xb_1 + e_1; \quad y_2 = Xb_2 + e_2 \quad (6)$$

But one can then put $y_1$ and $y_2$ side by side in a $n \times 2$ matrix and do the same for $b_1$ and $b_2$ and $e_1$ and $e_2$. So one gets

$$Y = XB + E \quad (7)$$

where $Y = (y_1, y_2)$, $B = (b_1, b_2)$ and $E = (e_1, e_2)$. A more graphical representation for $2 - p$ dependent variables is

$$\begin{bmatrix}
  Y \\
  \hline
  \text{m} \\
  \hline
  \text{n} \\
\end{bmatrix} = \begin{bmatrix}
  \text{m} \\
  \hline
  \text{n} \\
\end{bmatrix} \begin{bmatrix}
  X \\
  \hline
  B \\
\end{bmatrix} + \begin{bmatrix}
  \text{m} \\
  \hline
  \text{n} \\
\end{bmatrix} E
$$

This is the general case that will be referred to in the further text.

**Summary: MLR**

- For $m > n$, there is no unique solution unless one deletes independent variables.
- For $m = n$, there is one unique solution.
- For $m < n$, a least-squares solution is possible. For $m = n$ and $m < n$, the matrix inversion can cause problems.
- MLR is possible with more than one dependent variable.

**PRINCIPAL COMPONENT ANALYSIS (PCA): NIPALS METHOD**

Principal component analysis (PCA) is a method of writing a matrix $X$ of rank $r$ as a sum of $r$ matrices of rank 1:

$$X = M_1 + M_2 + M_3 + \ldots + M_r \quad (8)$$
or in graphical representation:

\[
X = \begin{bmatrix}
  t_1 \\
  t_2 \\
  \vdots \\
  t_d
\end{bmatrix}
\begin{bmatrix}
  p'_1 \\
  p'_2 \\
  \vdots \\
  p'_d
\end{bmatrix}
\]

(Rank is a number expressing the true underlying dimensionality of a matrix.) These rank 1 matrices, \(M_i\), can all be written as outer products of two vectors, a score \(t_i\) and a loading \(p'_i\):

\[
X = t_1p'_1 + t_2p'_2 + \ldots + t_dp'_d
\]  
(9)

or the equivalent \(X = TP'\) (\(P'\) is made up of the \(p'\) as rows and \(T\) of the \(t\) as columns) or graphically:

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_d
\end{bmatrix}
= \begin{bmatrix}
  t_1 \\
  t_2 \\
  \vdots \\
  t_d
\end{bmatrix}
\begin{bmatrix}
  p'_1 \\
  p'_2 \\
  \vdots \\
  p'_d
\end{bmatrix}
\]

To illustrate what the \(t_i\) and \(p'_i\) mean, an example for two variables, in the two-dimensional plane, is shown in Fig. 2A. Extension to more dimensions is easy but difficult to show on paper. For the example in Fig. 2A, the principal component is the line of best fit for the data points that are shown in Fig. 2B. Best fit means that the sum of squares of \(x_1\) and \(x_2\) residuals is minimized. This is also the average of both regression lines. It goes from \(-\infty\) to \(+\infty\). The \(p'_i\) is a \(1 \times 2\) row vector. Its elements, \(p_1\) and \(p_2\), are the direction cosines, or the projections of a unit vector along the principal component on the axes of

![Diagram](image)

Fig. 2. A principal component in the case of two variables: (A) loadings are the angle cosines of the direction vector; (B) scores are the projections of the sample points (1-6) on the principal component direction. Note that the data are mean-centered.

The NIPALS algorithm:

1. take a vector \(x_i\) from \(X^i\) (the scores)
2. calculate \(p'_i\): \(p'_i = \frac{x_i}{||x_i||}\)
3. normalize \(p_i\) to length 1
4. calculate \(t_i\): \(t_i = X^i \cdot p'_i\)
5. compare the \(t_i\) used same, stop (the iterative

![Diagram](image)

Fig. 3. Scores and load column of \(X\) is projected into an element of
the plot. The scores vector, $t_h$, is a $n \times 1$ column vector. Its elements are the coordinates of the respective points on the principal component line (Fig. 2B). For this example, it can easily be understood why one wants the length of the $p_h$ to be one [$\cos(\theta_1)^2 + \cos(\theta_2)^2 = \cos(\theta_1)^2 + \sin(\theta_1)^2 = 1$]; similar rules exist for more than two dimensions.

Generally, what one wants is an operator that projects the columns of $X$ onto a single dimension and an operator that projects the rows of $X$ onto a single dimension (see Fig. 3). In the first case, each column of $X$ is represented by a scalar; in the second case, each row of $X$ is represented by a scalar. In the rest of this section it will be shown that these operators are of a very simple nature.

Nonlinear iterative partial least squares (NIPALS) does not calculate all the principal components at once. It calculates $t_1$ and $p_1'$ from the $X$ matrix, then the outer product, $t_1 p_1'$, is subtracted from $X$ and the residual $E_1$ is calculated. This residual can be used to calculate $t_2$ and $p_2'$

$$E_1 = X - t_1 p_1'$$
$$E_2 = E_1 - t_2 p_2'$$

$$E_n = E_{n-1} - t_n p_n$$
$$E_{\text{tanh}}(X) = 0 = E_{\text{tanh}}(X) - t_{\text{tanh}}(X)p_{\text{tanh}}(X)$$

The NIPALS algorithm is as follows:

1. take a vector $x_i$ from $X$ and call it $t_h$: $t_h = x_i$

2. calculate $p_h$: $p_h = t_h^T X / t_h^T t_h$

3. normalize $p_h$ to length 1: $p_{\text{new}} = p_{\text{old}} / \|p_{\text{old}}\|$

4. calculate $t_h$: $t_h = X p_h / \|p_h\|$

5. compare the $t_h$ used in step 2 with that obtained in step 4. If they are the same, stop (the iteration has converged). If they still differ, go to step 2.

Fig. 3. Scores and loadings are obtained by projecting $X$ into vectors. Loadings: each column of $X$ is projected into an element of the vector $p'$. Scores: each row of $X$ is projected into an element of the vector $t$. 

(A) Loadings are the angle of the sample points (1-6) mean-centered.
(Note that after the first component is calculated, X in steps 2 and 4 has to be replaced by its residual.)

An explanation of how NIPALS works can be seen when one realizes that $t_n^T p_n$ in Eqn. 13, $p_n^T e$ in Eqn. 13, and $p_n^T p_n$ in Eqn. 14 are scalars. These scalar constants are best combined in one general constant $C$. Then one can substitute Eqn. 12 into 14: $t_n = X p_n$ and $p_n^T = t_n^T X$ give $C p_n^T = (X p_n)^T X$, or $C p_n = p_n^T X^T X$, or $(C l_m - X^T X) p_n = 0$; or one can substitute Eqn. 14 into 12 and get $(C' l_n = X X') t_n = 0$. These are the eigenvalue/eigenvector equations for $X X'$ and $X X'$ as used in the classical calculation. ($I_n$ is the identity matrix of size $n \times n$; $l_n$ is that of size $m \times m$.) The classical eigenvector and eigenvalue theory is well described by Strang [10].

It has been shown that on convergence, the NIPALS solution is the same as that calculated by the eigenvector formulae. The NIPALS method is convenient for microcomputers; it is also necessary for a good understanding of PLS. Otherwise, it does not matter what method one uses. In practical situations, NIPALS usually converges; in the case of non-convergence, two or more very similar eigenvalues exist. Then it does not matter which combination or rotation of eigenvectors is chosen. The reader is referred to Mardia et al. [5] for a more detailed discussion of PCA.

**Summary: PCA**
- A data matrix $X$ of rank $r$ can be decomposed to a sum of $r$ rank 1 matrices.
- These rank 1 matrices are outer products of vectors called scores and loadings.
- The scores and loadings can be calculated pair-by-pair by an iterative procedure.

**PRINCIPAL COMPONENT REGRESSION (PCR)**

The results from the section on PCA can be used to explain the principal component transformation of a data matrix $X$. This is a representation of $X$ as its scores matrix $T$ (where dimensions having small eigenvalues are excluded). The transformation is

$$T = X P = (T P^T = T I_n)$$

or graphically:

```
  T   =   X   =   P
  n   m   n   m
```

So now the MLR formula can be written as

$$Y = TB + E \text{ (solution: } \hat{B} = (T^T)^{-1} T^T Y)$$

or graphically:

```
  Y   =   T   =   B
  o   p   o   p
```

The variables of $X$ and $Y$ (orthogonality) and also of $T^T T$ should give no scores. Score vectors of order to avoid collinear
PCR solves the collin
in the calculation of $B$
ponents allows some no
step method and there
will end up in discar
remain in the compone

**Detailed information:**
- Smith [7]. Gunst and

**Summary: PCR**
- A data matrix can be
- A regression of the
- This regression gives

**PARTIAL LEAST-SQUARES**

**Model building**

The PLS model is mentioned in the PLS the data matrix. A si
the scores for the $X$: consist of outer relation (linking both blocks).

The outer relation

$$X = TP' + E = \sum t_n p_n$$

One can build the out

$$Y = UQ' + F^* = \sum u_i$$

Graphically, Eqns. 1'
In steps 2 and 4 has to be seen when one realizes that Eqn. 14 are scalars. These scalar amounts $C$. Then one can substitute $Cp_i = (Xp_i)'X$, or $Cp_i = Tp_i$, to Eqn. 14 into 12 and get the identity matrix of size eigenvector and eigenvalue

The NIPALS solution is the same as the NIPALS method is concern of a good understanding of one uses. In practical situations non-convergence, two or not matter which combination is referred to Mardia

is a sum of $r$ rank 1 matrices. The factors called scores and loadings, y-pair by an iterative pro-

To explain the principal components is a representation of $X$ small eigenvalues are ex-

\begin{align}
\sum_{i=1}^{n} t_{hi} p_i 
\end{align}

or graphically:

\begin{align}
\begin{bmatrix}
Y \\
X
\end{bmatrix} = \begin{bmatrix} T \\ P \\
\end{bmatrix} \begin{bmatrix} B \\ E \end{bmatrix}
\end{align}

The variables of $X$ are replaced by new ones that have better properties (orthogonality) and also span the multidimensional space of $X$. The inversion of $T'T$ should give no problem because of the mutual orthogonality of the scores. Score vectors corresponding to small eigenvalues can be left out in order to avoid collinearity problems from influencing the solution [9].

PCR solves the collinearity problem (by guaranteeing an invertible matrix in the calculation of $B$) and the ability to eliminate the lesser principal components allows some noise (random error) reduction. However, PCR is a two-step method and thereby has the risk that useful (predictive) information will end up in discarded principal components and that some noise will remain in the components used for regression.

Detailed information on PCR is given by Mardia et al. [5] and Draper and Smith [7]. Gunst and Mason [8] give a slightly different definition of PCR.

**Summary: PCR**
- A data matrix can be represented by its score matrix.
- A regression of the score matrix against one or several dependent variables is possible, provided that scores corresponding to small eigenvalues are omitted.
- This regression gives no matrix inversion problems; it is well conditioned.

**PARTIAL LEAST-SQUARES REGRESSION**

**Model building**

The PLS model is built on the properties of the NIPALS algorithm. As mentioned in the PCR section, it is possible to let the score matrix represent the data matrix. A simplified model would consist of a regression between the scores for the $X$ and $Y$ block. The PLS model can be considered as consisting of outer relations ($X$ and $Y$ block individually) and an inner relation (linking both blocks).

The outer relation for the $X$ block (cf. PCA section) is

\begin{align}
X = TP' + E = \sum t_{hi} p_i + E 
\end{align}

One can build the outer relation for the $Y$ block in the same way:

\begin{align}
Y = UQ' + F^* = \sum u_{hi} q_i + F^* 
\end{align}

Graphically, Eqns. 17 and 18 can be shown as
The summations are from 1 to \( n \). One can describe all the components and thus make \( E = F^* = 0 \) or not. How and why this is done is discussed below. It is the intention to describe \( Y \) as well as is possible and hence to make \( \| F^* \| \) as small as possible and, at the same time, get a useful relation between \( X \) and \( Y \). The inner relation can be made by looking at a graph of the \( Y \) block score, \( u \), against the \( X \) block score, \( t \), for every component (Fig. 4). The simplest model for this relation is a linear one:

\[
\hat{u}_n = b_n t_n
\]  

(19)

where \( b_n = u'_n t_n / t'_n t_n \). The \( b_n \) play the role of the regression coefficients, \( b \), in the MLR and PCR models.

This model, however, is not the best possible. The reason is that the principal components are calculated for both blocks separately so that they have a weak relation to each other. It would be better to give them information about each other so that slightly rotated components result which lie closer to the regression line of Fig. 4.

**Over-simplified model: 2x PCA.** An over-simplified model can be written in algorithmic form as in the NIPALS section.

For the \( X \) block: (1) take \( t_{start} = \) some \( x_j \); (2) \( p' = t'X/t' \) \( (= u'X/u'u) \); (3) \( p'_{new} = p'_{old}/\|p'_{old}\| \); (4) \( t = Xp/p'p \); (5) compare \( t \) in steps 2 and 4 and if they are equal stop, else go to 2.

For the \( Y \) block, (3) \( q'_{new} = q'_{old}/\|q'_{old}\| \); they are equal stop, else:

**Improving the inner relation** written as completely as possible in parentheses in this sequence: (1) take \( u_{old} = p'_{old}/\|p'_{old}\| \); (2) \( w'_{new} = w'_{old} \); (3) \( q'_{new} = q'_{old}/\|q'_{old}\| \); (6) \( \hat{u}_n = b_n t_n \) (one in the preceding iteration error), stop; else go to a new variable, steps 5–8 can be repeated.

This algorithm usually for \( X \) and \( Y \) block.

**Obtaining orthogonal relations** does not give orthogonal relations that was used in PCA, but this turns out to be an extra loop can be inserted:

\[
p' = t'X/t't
\]

With \( p'_{new} = p'_{old}/\|p'_{old}\| \), but this makes the \( Y \) block.

Eqn. 20: \( t_{new} = t_{old}/p' \), but this makes the \( X \) block.

Eqn. 21: rescaling to the weight \( w'_{new} = w'_{old}/\|p'_{old}\| \). Then the residuals can be calculated:

\[
E_n = F_n - t_n p'_n; \quad X
\]

\[
F^*_n = F^*_n - u_n q'_n; \quad Y
\]

But in the outer relation \( b_n t_n \), and a mixed relation

\[
F_n = F_n - b_n t_n q'_n
\]

(It is recalled that the \( \hat{u} \) vectors to use the more, because the rank can go on until the rank is given in the Appendix.)

---

Fig. 4. The inner relation. A linear regression of \( u \) against \( t \). Note that the data are mean-centered.
For the Y block, (1) take \( u_{\text{start}} = \text{some } y_i \); (2) \( q' = u'Y/u' = (t'Y/t') \); (3) \( q'_{\text{new}} = q'_{\text{old}}/\|q'_{\text{old}}\| \); (4) \( u = Yq/q \); (5) compare \( u \) in steps 2 and 4 and if they are equal stop, else go to 2.

**Improving the inner relation: exchange of scores.** The above relations are written as completely separated algorithms. The way each can get information about the other is to let \( t \) and \( u \) change place in step 2. (Note the parts in parentheses in this step.) Thus, the two algorithms can be written in sequence: (1) take \( u_{\text{start}} = \text{some } y_i \); (2) \( p' = u'X/u' = (w' = u'X/u') \); (3) \( p'_{\text{new}} = p'_{\text{old}}/\|p'_{\text{old}}\| \); (4) \( t = Xp/p'p = (t = Xw/w'w) \); (5) \( q' = t'Y/t't \); (6) \( q'_{\text{new}} = q'_{\text{old}}/\|q'_{\text{old}}\| \); (7) \( u = Yq/q \); (8) Compare the \( t \) in step 4 with the one in the preceding iteration step. If they are equal (within a certain rounding error), stop; else go to 2. (In the case for which the Y block has only one variable, steps 5–8 can be omitted by putting \( q = 1 \)).

This algorithm usually converges very quickly to give rotated components for X and Y block.

**Obtaining orthogonal X block scores.** There is still a problem; the algorithm does not give orthogonal \( t \) values. The reason is that the order of calculations that was used for the PCA has been changed. Therefore, the \( p' \) are replaced by weights \( w' \) (see formulas in parentheses in previous subsection). An extra loop can be included after convergence to get orthogonal \( t \) values:

\[
p' = t'X/t't
\]

With \( p'_{\text{new}} = p'_{\text{old}}/\|p'_{\text{old}}\| \), it becomes possible to calculate the new \( t = Xp/p'p \), but this turns out to be just a scalar multiplication with the norm of the \( p' \) in Eqn. 20: \( t_{\text{new}} = t_{\text{old}}p'_{\text{old}} \). Orthogonal \( t \) values are not absolutely necessary, but they make the comparison with PCR easier. One must give the same rescaling to the weights, \( w' \), if the prediction is to be made without error: \( w'_{\text{new}} = w'_{\text{old}}/\|p'_{\text{old}}\| \). Then \( t \) can be used for the inner relation as in Eqn. 19, and the residuals can be calculated from \( E = X - t_p p' \) and \( F = Y - u_q q' \). In general,

\[
E_n = E_{n-1} - t_n p_n; \quad X = E_0
\]

\[
F_n = F_{n-1} - u_n q_n; \quad Y = F_0
\]

But in the outer relation for the Y block, \( u_n \) is replaced by its estimator, \( \tilde{u}_n = b_n t_n \), and a mixed relation is obtained:

\[
F_n = F_{n-1} - b_n t_n q_n
\]

(It is recalled that the aim is to make \( \|F_n\| \) small.) This mixed relation ensures the ability to use the model parameters for predicting from a test set. Furthermore, because the rank of \( Y \) is not decreased by 1 for each component, one can go on until the rank of the X block is exhausted. The complete algorithm is given in the Appendix together with an illustration of the matrices and vectors.

Note that the data are mean-
Summary: PLS
- There are outer relations of the form $X = TP' + E$ and $Y = UQ' + F^*$.
- There is an inner relation $\hat{u}_h = b_h t_h$.
- The mixed relation is $Y = T B Q' + F$ where $\|F\|$ is to be minimized.
- In the iterative algorithm, the blocks get each other's scores, this gives a better inner relation.
- In order to obtain orthogonal X scores, as in the PCA, it is necessary to introduce weights.

Properties of the PLS factors
For the user of PLS, it is obviously of interest to know what kind of properties to expect from it. The main properties can be summarized as follows.
The quantities $p_h$ and $q_h$ have unit length for each $h$: $\|p_h\| = \|q_h\| = 1$, or $\Sigma p_h^2 = 1$ and $\Sigma q_h^2 = 1$ for $j = 1$ to $m$.
Both $t_h$ and $u_h$ are centered around zero for each $h$: $\Sigma t_{hi} = 0$ and $\Sigma u_{hi} = 0$ for $i = 1$ to $n$.
The $w_h$ are orthogonal: $w_i w_j = \delta_{ij} \|w_i\|^2$ where $\delta_{ij}$ is the Kronecker delta.
The $t_h$ are orthogonal: $t_i^* t_j = \delta_{ij} \|t_i\|^2$.
It is useful to check that these properties hold for a number of data sets. These properties are also good indicators for computer rounding errors.

Prediction
The important part of any regression is its use in predicting the dependent block from the independent block. This is done by decomposing the X block and building up the Y block. For this purpose, $p'$, $q'$, $w'$ and $b$ from the calibration part are saved for every PLS factor. It should be noted that the new X block has $r$ samples instead of $n$.
The independent blocks are decomposed and the dependent block is built up. For the X block, $t$ is estimated by multiplying X by $w$ as in the model building part

$$\hat{t}_h = E_{h-1} w_h$$

$$E_h = E_{h-1} - \hat{t}_h b_h$$

For the Y block:

$$Y = F_h = \Sigma b_h \hat{t}_h q_h'$$

where the summation is over $h$ for all the factors (a) one wants to include and $X = E_0$, $Y = F^*$.

Number of components
If the underlying model for the relation between X and Y is a linear model, the number of components needed to describe this model is equal to the model dimensionality. Nonlinear models require extra components to describe nonlinearities. The number of components to be used is a very important property of a PLS model.

Although it is possible of the X block matrix for this are that the smaller components carry the problems of earlier paragraphs, it should be small. Figure It is possible to choose the threshold. Another and previous $\|F_h\|$ value used to some previously established difference method. Sometimes the analysis used to validate the regression [7].

The above-mentioned of PLS. If prediction establish the number of validation. One can cal PRESS (prediction residual error sum of squares) statistic ago wants to use the number of components i.e., the smallest number of components [11] for more detail.

![Fig. 5. $\|F_h\|$ vs. the number of components](image)

![Fig. 6. Plot of PRESS against the number of components](image)
+ E and Y = UQ'*F.*. 

be minimized, 
other's scores, this gives a 
+ PCA, it is necessary to 
know what kind of pro-
the summarized as follows.
h: \|p_h\| = \|q'_h\| = 1, or 
h: \Sigma t_{hi} = 0 and \Sigma u_{hi} = 0 
is the Kronecker delta. 
or a number of data sets. 
or rounding errors. 

predicting the dependent 
composing the X block 
, w' and b from the cal-
be noted that the new 
dependent block is built 
X by w as in the model 

\begin{align}
(24) 
(25) 
(26) 
\end{align}

one wants to include 

and Y is a linear model, 
model is equal to the 
extra components to dese-
e used is a very impor-

Although it is possible to calculate as many PLS components as the rank 
of the X block matrix, not all of them are normally used. The main reasons 
for this are that the measured data are never noise-free and some of the 
smaller components will only describe noise, and that, as mentioned in 
earlier paragraphs, it is common to leave out small components because they 
carry the problems of collinearity. 

This means that there must be one or several methods to decide when to stop. One possible criterion can be found in Eqn. 23, where the norm of F_h 
should be small. Figure 5 gives a plot of \|F_h\| vs. the number of components. 
It is possible to choose a threshold level and to stop when \|F_h\| goes below 
that threshold. Another possibility is to look at the difference between actual 
and previous \|F_h\| values and to stop when this becomes small compared to 
some previously established measurement error. A combination of threshold 
and difference methods would be preferable. 

Sometimes the analysis of variance with F-test on the inner relation can be 
used to validate the model. In this case, one uses the F-test on the linear 
regression [7].

The above-mentioned methods are valuable for the model-building stage 
of PLS. If prediction is desired, another class of methods must be used to 
establish the number of components needed. These methods are called cross-
validating. One can calculate a statistic for lack of prediction accuracy called 
PRESS (prediction residual sum of squares). Figure 6 gives a sample plot of 
the PRESS statistic against the number of components. It is obvious that one 
wants to use the number of components that gives a minimal PRESS. The 
location of this minimum is not always well defined. The evaluation of the 
number of components is analogous to the concept of detection limits, 
\text{i.e.}, the smallest signal that can be detected in the presence of noise. See 

![Fig. 5] \(\|F_h\|\) vs. the number of PLS components. A threshold and/or a difference criterion (see \(\Delta\) in figure) can be used to stop the algorithm.

![Fig. 6] Plot of PRESS against the number of components. This criterion evaluates the predictive power of the model. The number of components giving a minimum PRESS is the right number for the model that gives optimal prediction. In this example, models with 4–8 components would be acceptable.
Statistics

From the matrices of residuals $E_h$ and $F_h$, sums of squares can be calculated as follows: the total sum of squares over a matrix, the sums of squares over rows, and the sums of squares over columns. These sums of squares can be used to construct variance-like estimators. The statistical properties of these estimators have not undergone a rigorous mathematical treatment yet, but some properties can be understood intuitively.

The sum of squares of the $F_h$ is the indicator of how good the model is (Eqn. 23). The sum of squares of $E_h$ is an indicator of how much of the $X$ block is not used in the model. In some cases, a substantial part of the $X$ block does not participate in the model, which means that the independent variables have unexpected properties or large errors.

Fig. 7. Statistics for the variables. The data are shown as bars representing the sum of squares per variable (for the model building both $X$ and $Y$ variables, for the prediction only $X$ variables). After 0 PLS components, the data is in mean-centered and variance-scaled form. As the number of PLS components increases, the information in each variable is exhausted. The hatched bar shows the behavior of a "special" variable, one that contributes little to the model.

Fig. 8. Statistics for the objects (samples). The data are shown as bars representing the sum of squares per object. As the number of PLS components increases, the sum of squares for each object decreases. The hatched bar shows the behavior of a "special" object, probably an outlier.

Sums of squares over a certain component describe how well the objects fit the model. Illustrations are given in the following pages. More on this elsewhere [4, 6].

An advantage of PLS is its ability to handle non-linearities and interactions among variables. The evolution of these variables as more and more components are added is also of interest. The criterion for model selection is how many components contribute mainly to the model.

Conclusion

The topic of partial least squares is not covered in this book. Some subjects are covered in [4, 6]. The treatment of non-linearities, leverage, and interactions in PLS is also of interest. There are also other methods for model selection, and elegant ones when applied to simulated data is given.

The authors wish to thank D. W. and F. P. for their help in the preparation of this report. The Science Department of P. Geladi with a NA
Sums of squares can be calculated for the sums of squares matrix, the sums of squares matrix. These sums of squares can be statistical properties of the data. A statistical treatment of these properties can be useful in determining how good the model is or of how much of the X data is explained. A substantial part of the X data is explained if the independent variables are significant. A graphical representation of the matrices and vectors used in PLS is shown in Fig. 9.

![Fig. 9. A graphical representation of the matrices and vectors used in PLS.](image)

Sums of squares over the columns indicate the importance of a variable for a certain component. Sums of squares over the rows indicate how well the objects fit the model. This can be used as an outlier detection criterion. Illustrations are given in Fig. 7 for variable statistics and in Fig. 8 for sample statistics. More on this can be found in the articles by S. Wold et al. as cited elsewhere [4, 6].

An advantage of PLS is that these statistics can be calculated for every component. This is an ideal means of following the model-building process. The evolution of these statistics can be followed (as shown in Figs. 7 and 8) as more and more components are calculated so that an idea of how the different objects and variables fit can be obtained. In combination with a criterion for model dimensionality, the statistics can be used to estimate which objects and variables contribute mainly to the model and which contribute mainly to the residual.

**Conclusion**

The topic of partial least squares is much larger than the material covered above. Some subjects not discussed at all or not in detail are: outlier detection, treatment of missing data, F and t statistics, classification/pattern recognition, leverage, selection of variables, data transformations, extensions to more blocks and hierarchical models, and lack of fit.

There are also other PLS algorithms. Each may have some advantage in a particular application. The algorithm given here is one of the most complete and elegant ones when prediction is important. An example of its application to simulated data is given in the next paper [11].

The authors thank Svante Wold and Harald Martens for their contributions to the PLS method. This paper results from a number of discussions at the Laboratory for Chemometrics. We thank all colleagues and visitors to the lab and especially Dave Veithkamp for their support and stimulating discussions. The Science Department of the Belgian Ministry of Foreign Affairs provided P. Geladi with a NATO travel grant 10/B/84/B.E. Graphics were done by
Appendix: The PLS algorithm

It is assumed that X and Y are mean-centered and scaled:

For each component: (1) take $u_{\text{start}} = \text{some } y_i$

In the X block: (2) $w' = u'X / u'u$

(3) $w_{\text{new}} = w_{\text{old}} / \|w_{\text{old}}\|$ (normalization)

(4) $t = Xw / w'w$

In the Y block: (5) $q' = t'Y / t't$

(6) $q_{\text{new}} = q_{\text{old}} / \|q_{\text{old}}\|$ (normalization)

(7) $u = Yq / q'q$

Check convergence: (8) compare the $t$ in step 4 with the one from the preceding iteration. If they are equal (within a certain rounding error) go to step 9, else go to step 2. (If the Y block has only one variable, steps 5–8 can be omitted by putting $q = 1$, and no more iteration is necessary.)

Calculate the X loadings and rescale the scores and weights accordingly:

(9) $p' = t'X / t't$

(10) $p_{\text{new}} = p_{\text{old}} / \|p_{\text{old}}\|$ (normalization)

(11) $t_{\text{new}} = t_{\text{old}} / \|t_{\text{old}}\|$

(12) $w_{\text{new}} = w_{\text{old}} / \|w_{\text{old}}\|$

($p'$, $q'$ and $w'$ should be saved for prediction; $t$ and $u$ can be saved for diagnostic and/or classification purposes).

Find the regression coefficient $b$ for the inner relation:

(13) $b = u't/t't$

Calculation of the residuals. The general outer relation for the X block (for component $h$) is

$E_h = E_h - t_h p_h' X = E_x$

The mixed relation for the Y block (for component $h$) is

$F_h = F_h - t_h q_h' Y = F_y$

From here, one goes to Step 1 to implement the procedure for the next component. (Note: After the first component, X in steps 2, 4 and 9 and Y in steps 5 and 7 are replaced by their corresponding residual matrices $E_h$ and $F_h$.)

Matrices and vectors are shown graphically in Fig. 9.

REFERENCES

7 D. Belsley, E. Kuh and and Sources of Collinearity.
10 G. Strang, Linear Algebra.
11 P. Geladi and B. Kowalski.
from the preceding iteration. 
step 9, else go to step 2. (If the
by putting $q = 1$, and no more
points accordingly:

be saved for diagnostic and/or

or the X block (for component

figure for the next component.

Y in steps 6 and 7 are replaced

elses under Indirect Observation,
Observation, North-Holland,