Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 02-188

Linear and Non-Linear Modelling in Soft4s[1]

Oscar Barrero Mendoza, Luc Hoegaerts, Kristiaan Pelckmans and Bart De Moor[2]

November 7, 2002

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/sista/barrero/reports/Modsoft4s.ps.gz*

[2]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SISTA/SCD, Kastelpark Arenberg 10 3001 Leuven, Belgium, Tel. 32/16/32 17 09, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.ac.be/sista*. E-mail: *bart.demoor@esat.kuleuven.ac.be*. This work was supported in part by the Soft4s project of IWT Belgium.

Contents

1 Introduction **1**

2 Linear Modeling Techniques **1**
2.1 Notation 1
2.2 Some definitions 2
2.2.1 Covariance 2
2.2.2 Properties 2
2.3 Least Squares (LS) Regression 2
2.3.1 Generalities 2
2.3.2 Mathematical Description 3
2.4 Principal Component Regression (PCR) 4
2.4.1 Generalities 4
2.4.2 Mathematical Description 4
2.4.3 Algorithm 9
2.5 Partial Least Squares (PLS) 12
2.5.1 Generalities 12
2.5.2 Mathematical Description 12
2.5.3 Algorithm 17
2.6 Canonical Correlation Analysis (CCA) 20
2.6.1 Generalities 20
2.6.2 Mathematical Description 21
2.6.3 Algorithm 22
2.7 Discussion about differences and Similarities LS, PCR, PLS, CCA 24

3 Non-Linear Modeling Techniques **26**

4 Input Selection **26**
4.1 Automatic relevance detection 26
4.1.1 The Bayesian Framework 26
4.1.2 ARD for MLP 28
4.1.3 Some results of applying ARD 30

5 Ammonia Experimental Section **31**
5.1 Data Preprocessing 31
5.2 Model Validation 31
5.2.1 Quantitative validation and Component selection . . . 31
5.2.2 Quantitative validation results 33
5.2.3 Qualitative validation results 34

5.2.4 Table of Results 35

A Figures **36**
A.1 Quantitative test figures 36
A.2 Qualitative test figures 38
A.2.1 Residual Analysis (Validation test) 38
A.2.2 Model Correspondence 40

B Matlab Algorithms **41**
B.1 PCR-CCA1 Matlab algorithm 41
B.2 PLS1 Matlab algorithm 41
B.3 PLS2 Matlab algorithm 42
B.4 CCA2 Matlab algorithm 42

C CCA Several Outputs - Several Inputs **44**
C.0.1 Mathematical Description 44
C.0.2 Algorithm 45

D Bibliography **47**

1 Introduction

2 Linear Modeling Techniques

2.1 Notation

For clarity we first sum up some conventions on notation and used symbols.

A dataset typically consists of n samples of p random variables $\alpha_1, \alpha_2, \ldots, \alpha_p$. No special assumptions are made about the distribution of the variables. The data can be ordered in a matrix $\mathbf{A}$: rows are observations, columns are variables, and we suppose the data values for every variable to be centered around its mean. All values are in the real valued domain.

n = number of measurements, typically 100 000.
p = number of independent variables, typically 50-100.
q = number of dependent variables, typically 1.
$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_p)^T = p \times 1$ vector of independent variables.
$\mathbf{y} = (y_1, y_2, \ldots, y_q)^T = q \times 1$ vector of dependent variables.
$\mathbf{A} = n \times p$ data matrix of observations of α variables.
$\mathbf{Y} = n \times q$ data matrix of observations of y variables.
$\alpha_\mathbf{i} = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ip})^T$ with $i = 1, \ldots, n$ are the specific observations of the general p-element random vector α.
$\mathbf{y_i} = (y_{i1}, y_{i2}, \ldots, y_{ip})^T$ with $i = 1, \ldots, n$ are the specific observations of the general p-element random vector $\mathbf{y}$.
Linear combinations in α variables in matrix element notation : $u_{i1} = v_{11}\alpha_{i1} + v_{21}\alpha_{i2} + \ldots + v_{p1}\alpha_{ip} = \mathbf{v_1^T}\alpha_\mathbf{i}, \ldots, \mathbf{u_p} = \mathbf{v_p^T}\alpha$ (with $i = 1, \ldots, n$), or in matrix column vector notation, $u_1 = \mathbf{v_1^T}\alpha, \mathbf{u_2} = \mathbf{v_2^T}\alpha, \ldots, \mathbf{u_p} = \mathbf{v_p^T}\alpha$ or $\mathbf{u} = \mathbf{V^T}\alpha$.
Linear combinations in y variables in matrix element notation : $z_{i1} = w_{11}y_{i1} + w_{21}y_{i2} + \ldots + w_{p1}y_{ip} = \mathbf{w_1^T y_i}, \ldots, \mathbf{z_p} = \mathbf{w_p^T y}$ (with $i = 1, \ldots, n$), or in matrix column vector notation, $\mathbf{z_1} = \mathbf{w_1^T y}, \mathbf{z_2} = \mathbf{w_2^T y}, \ldots, \mathbf{z_q} = \mathbf{w_q^T y}$ or $\mathbf{z} = \mathbf{W^T y}$.
Coefficients of the linear combinations in α : $\mathbf{v_i}$ = column vector of $\mathbf{V}$ $(p \times p)$.
Coefficients of the linear combinations in y: $\mathbf{w_i}$ = column vector of $\mathbf{W}$ $(q \times q)$.
$\mathbf{U} = \mathbf{AV} = n \times p$ = observations in the u variables = matrix of transformed observations in x, or in matrix column vector notation, $\mathbf{u_i} = \mathbf{A v_i}$ for $i = 1, \ldots, n$.
$\mathbf{Z} = \mathbf{AW} = n \times p$ = observations in the w variables = matrix of transformed observations in y, or in matrix column vector notation, $\mathbf{z_i} = \mathbf{A w_i}$ for $i = 1, \ldots, n$.
$\hat{\mathbf{y}}$ = mean value of vector y.
$< \alpha, \mathbf{y} >= \alpha^\mathbf{T}\mathbf{y}$ = Inner product between the vectors α and y.
$< \mathbf{A}, \mathbf{Y} >= \mathbf{A^T Y}$ = Inner product between the matrices A and Y.

2.2 Some definitions

2.2.1 Covariance

Relevant information regarding the linear relation between variables is available through the covariance of every pair of variables (a measure of the tendency to vary together, to co-vary): it is the average of the products of the deviations of variable values around their mean.

Covariance between variable α_i and α_j:

$$cov(\alpha_i, \alpha_j) = \frac{1}{(n-1)} \sum_{k=1}^{n} (\alpha_{ki} - \bar{\alpha_i})(\alpha_{kj} - \bar{\alpha_j}) \tag{1}$$

These covariances between the x variables can be ordered in a covariance matrix

$$cov(\alpha, \alpha) = [cov(\alpha_i, \alpha_j)] = \mathbf{S_{AA}} = \frac{1}{(n-1)} \mathbf{A^T A}, \tag{2}$$

which is positive-semidefinite (symmetric, eigenvalues positive or zero) and which keeps information about the redundancy between variables. For most calculations involving $\mathbf{S_{AA}}$, the factor $\frac{1}{(n-1)}$ will be neglected.

When we regard the *alpha* and y variables as two separate sets, one speaks of the within-set covariance matrix in the α variables, $\mathbf{S_{AA}} = \mathbf{A^T A}$, and in the y variables, $\mathbf{S_{yy}} = \mathbf{Y^T Y}$, while there are also the between-set covariance matrices $\mathbf{S_{Ay}} = \mathbf{A^T Y}$ and $\mathbf{S_{yA}} = \mathbf{Y^T A} = (\mathbf{A^T Y})^\mathbf{T} = \mathbf{S_{Ay}^T}$.

2.2.2 Properties

$||\mathbf{Av}||^2 =< \mathbf{Av}, \mathbf{Av} >= \mathbf{v^T A^T A v} = \mathbf{v^T} cov(\alpha, \alpha)\mathbf{v} = \mathbf{v^T} var(\alpha)\mathbf{v} = var(\mathbf{v^T}\alpha)$

$\mathbf{S_{AA}} = cov(\alpha, \alpha) = var(\alpha)$ $\mathbf{S_{Ay}} = cov(\alpha, \mathbf{y}) =< \mathbf{A}, \mathbf{Y} > cov(\mathbf{a}\alpha, \mathbf{by}) = \mathbf{a^T} cov(\alpha, \mathbf{y})\mathbf{b}$ where a and b are deterministic vector. $< \mathbf{a}, \mathbf{b} >= ||\mathbf{a}||.||\mathbf{b}||.cos(\mathbf{a}, \mathbf{b})$

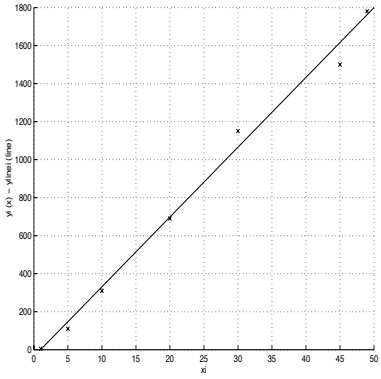
$$cor(\mathbf{v^T}\alpha, \mathbf{w^T y}) = \frac{cov(\mathbf{v^T}\alpha, w^T y)}{\sqrt{\left(var(\mathbf{v^T}\alpha)\ var(\mathbf{w^T y})\right)}}$$

2.3 Least Squares (LS) Regression

2.3.1 Generalities

One of the most basic analysis techniques is the "method of least squares" for fitting a straight line to a set of data.

Given a set of (x,y) points, find m and b such that the line "best" fits the data.



If you think about a single data point (xi, yi) there is a point on the line which corresponds to the x, but may differ from the y:

$$\mathbf{y_{linei} = mx_i + b}$$

We call the difference (distance) between the data point y and the line y the "residual":

$$\mathbf{e_i = y_i - y_{linei} = y_i - mx_i - b}$$

Every data point has an associated residual that measures its closeness to the line.

The line which "best fits" the data is the line that causes the residuals to be the smallest.

"Least Squares" says that the best fit is the one that minimizes the sum of the squares of the residuals. Squares are used to balance positive and negative differences and penalize big errors.

2.3.2 Mathematical Description

Now, given the set of linear equations

$$\mathbf{y = Ax + e} \ \textit{where}\ \mathbf{y} \in \mathbb{R}^{\mathbf{n}},\ \mathbf{A} \in \mathbb{R}^{\mathbf{n \times p}}\ \textit{has rank} = p,\ \textit{and}\ \mathbf{x} \in \mathbb{R}^{\mathbf{p}} \qquad (3)$$

it is asked to find the regression vector x which minimizes $||e||^2$. Therefore, the following optimization problem is formulated:

$$\begin{aligned} &\mathbf{min} \qquad (\mathbf{y - Ax})^{\mathbf{T}}(\mathbf{y - Ax}) \\ &\mathbf{x} \in \mathbb{R}^{\mathbf{n}} \end{aligned} \qquad (4)$$

which gives

$$\mathbf{x} = (\mathbf{A^T A})^{-1}\mathbf{A^T y} \;=\; \mathbf{S_{AA}^{-1} S_{Ay}} \tag{5}$$

The equation (5) is known as the Linear Least Squares solution.

Remarks

Linear Least Squares regression is a good method for fitting but not for prediction, it means that given a data set for some inputs and one output, **LS** can find a regression vector x which minimizes the $||e||^2$ of the given data set (fitting), but if the data set for the same inputs and output is changed, and the same regression vector x is used (prediction), the $||e||^2$ will be very much higher, it follows **LS** is a bad prediction method. On the other hand, if the matrix A is very ill-conditioned, which often occurs in real life, the computation of $(A^T A)^{-1}$ will cause some problems, hence to solve that, the pseudo-inverse of $A^T A$ is computed (see eq.(25)).

2.4 Principal Component Regression (PCR)

2.4.1 Generalities

PCR is based on PCA (Principal Components Analysis), where the goal of PCA is to reduce the dimensionality of a dataset in which there are a large number of interrelated variables, while retaining as much as possible discriminatory information (variance). The reduction is achieved by transforming to a new set of variables, the principal components, which are uncorrelated, and ordered so that the first few retain most of the variation present in all of the original variables. Computation of the PCs reduces to the solution of an eigenvalue problem for the real-valued symmetric covariance matrix.

In PCA one seeks for directions (linear combinations) of maximal variance that are orthogonal to each other, and with a scaling by the norm of the coefficient vector. Geometrically, the orthogonal axes in x space are rigidly rotated such that the variance with respect to the new axes is maximal, and the coefficient vectors lie on the unit sphere.

2.4.2 Mathematical Description

PCA Mathematical Description

Given p variables in α : $\alpha_1, \alpha_2, \ldots, \alpha_p$. Suppose that $\mathbf{v^T}\alpha$ is a linear combination in the α variables, where $\mathbf{v}$ is a $(p \times 1)$ vector of coefficients.

$$\max_{\substack{\mathbf{v} \in \mathbb{R}^{\mathbf{p}} \\ \|\mathbf{v}\| = 1 \\ \mathbf{v^T S_{AA} V_{k-1}} = \mathbf{0^T}}} var(\mathbf{v^T}\alpha) = \mathbf{v_k} \tag{6}$$

where $1 \leq k \leq \text{rank}(\mathbf{S_{AA}})$ and $\mathbf{V_{k-1}} = [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{k-1}}]$ with dimension $p \times (k-1)$.

Derivation of Principal Components

Step 1
Look for a linear combination

$$u_1 = v_{11}\alpha_1 + v_{21}\alpha_2 + ... + v_{p1}\alpha_p = \mathbf{v_1^T}\alpha \tag{7}$$

which maximizes

$$var(\mathbf{v_1^T}\alpha) = \|\mathbf{A v_1}\|^2 = \mathbf{v_1^T A^T A v_1} = \mathbf{v_1^T S_{AA} v_1}, \tag{8}$$

where we neglect the factor $1/(n-1)$. Since the coefficients vector $\mathbf{v_1}$ can be chosen arbitrarily high, it is constrained, which comes down to a normalization of the linear combination by

$$\|\mathbf{v_1}\|^2 = \mathbf{v_1^T v_1} = 1. \tag{9}$$

Using the technique of Lagrange multipliers, one maximizes

$$\mathbf{v_1^T S_{AA} v_1} \quad \text{subject to} \quad \|\mathbf{v_1}\| = 1 \tag{10}$$

by extremizing

$$\phi_1(\mathbf{v_1}) = \mathbf{v_1^T S_{AA} v_1} + \lambda_1(\mathbf{v_1^T v_1} - 1). \tag{11}$$

Differentiation with respect to $\mathbf{v_1}$ and equating to zero gives

$$\mathbf{S_{AA}}\ \mathbf{v_1} + \lambda_1 \mathbf{v_1} = 0, \tag{12}$$

which is an eigenvalue problem for matrix $\mathbf{S_{AA}}$. To decide which of the p eigenvectors is the maximizing value of $\mathbf{v_1}$, note that the quantity to be maximized, namely

$$\mathbf{v_1^T S_{AA} v_1} = \lambda_1 \mathbf{v_1^T v_1} = \lambda_1, \tag{13}$$

must be as large as possible and thus the solution for $\mathbf{v_1}$ is the eigenvector corresponding to the largest eigenvalue of matrix $\mathbf{S_{AA}}$.

Step 2
Look again for a linear combination

$$u_2 = v_{12}\alpha_1 + v_{22}\alpha_2 + ... + v_{p2}\alpha_p = \mathbf{v_2^T}\alpha \tag{14}$$

which maximizes

$$var(\mathbf{v_2^T}\alpha) = \|\mathbf{A v_2}\|^2 = \mathbf{v_2^T A^T A v_2} = \mathbf{v_2^T S_{AA} v_2}, \tag{15}$$

but which is uncorrelated (i.e. orthogonal) with the previous linear function u_1:

$$cov(\mathbf{v_1^T}\alpha, \mathbf{v_2^T}\alpha) =< \mathbf{v_1^T}\alpha, \mathbf{v_2^T}\alpha >= \mathbf{v_2^T S_{AA} v_1} = \lambda_1 \mathbf{v_2^T v_1} = 0, \tag{16}$$

Using the technique of Lagrange multipliers, one maximizes

$$\mathbf{v_2^T S_{AA} v_2} \quad \text{subject to} \quad \|\mathbf{v_2}\| = 1 \quad \text{and} \quad \mathbf{v_1^T S_{AA} v_2} = 0, \tag{17}$$

by maximizing

$$\phi_2(\mathbf{v_2}) = \mathbf{v_2^T s s^T v_2} + \lambda_2(\mathbf{v_2^T v_2 - 1}) + \mu_2(\mathbf{v_2^T S_{AA} v_1}). \tag{18}$$

Differentiation with respect to $\mathbf{v_2}$ and setting equal to zero gives

$$\mathbf{S_{AA} v_2} + \lambda_2 \mathbf{v_2} + \mu_2 \mathbf{S_{AA} v_1} = 0. \tag{19}$$

To get rid of the term in λ_2, we premultiply with $\mathbf{v_1}$:

$$\mathbf{v_1^T S_{AA} v_2} + \lambda_2 \mathbf{v_1^T v_2} + \mu_2 \mathbf{v_1^T S_{AA} v_1} = 0, \tag{20}$$

and solve for μ_2, which gives $\mu_2 = 0$, since the first two terms are zero and $\mathbf{v_1^T S_{AA} v_1} = 1$. This last equation reduces then to an eigenvalue problem for the same matrix $\mathbf{S_{AA}}$ from the previous step:

$$\mathbf{S_{AA}\ v_2} + \lambda_2 \mathbf{v_2} = 0, \tag{21}$$

and thus is the solution for $\mathbf{v_2}$ the eigenvector corresponding to the second largest eigenvalue of matrix $\mathbf{S_{AA}}$.

Next steps

The procedure can be continued in a similar way for further linear combinations. The maximum number of such directions that can be found is equal to the rank of $\mathbf{S_{AA}}$. The new constructed u variables are called principal components (pc's = transformed variables). On the other hand, transformed observations are called 'scores'. The coefficients of the linear combination, gathered in the v vector, are called 'loadings'.

Summary

Given p variables in α: $\alpha_1, \alpha_2, \ldots, \alpha_p$. Suppose that $\mathbf{u} = \mathbf{v^T}\alpha$ is a linear combination in the α variables, where $\mathbf{v}$ is a $(p \times 1)$ vector of coefficients.

In PCA one solves the following optimization problem:

$$\begin{array}{c} \max \\ \mathbf{v} \in \mathbb{R}^{\mathbf{p}} \\ \|\mathbf{v}\| = 1 \\ \mathbf{v^T S_{AA} V_{k-1}} = \mathbf{0^T} \end{array} \qquad var(\mathbf{v^T}\alpha) = \mathbf{v_k} \tag{22}$$

where $1 \leq k \leq$ (rank $\mathbf{S_{AA}}$) and $\mathbf{V_{k-1}} = [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{k-1}}]$ with dimension $p \times (k-1)$. The constraint that $\mathbf{v^T S_{AA} V_{k-1}} = \mathbf{0^T}$ is a compact notation for saying that $\mathbf{u_i} \perp \mathbf{u_j}\ \forall i, j = 1, \ldots, p$ with $i \neq j$.

It turns out that the k th PC, given by $\mathbf{u_k} = \mathbf{v_k^T}\alpha$, is the eigenvector of the covariance matrix $\mathbf{S_{AA}}$ corresponding to the kth largest eigenvalue λ_k. Furthermore, if the coefficient vectors $\mathbf{v_k}$ are chosen to have unit length (thus constrained to the unit sphere, $\mathbf{v_k^T v_k} = 1$), then $var(u_k) = \lambda_k$.

Remarks

-In other words, the idea is to linearly transform the variables, thus have a change to a new basis (rotation/reflection). In matrix form this is stated as $\mathbf{u} = \mathbf{V^T}\alpha$**.** As optimal choice for $\mathbf{V}$, it turns out that $\mathbf{V}$ has the eigenvectors as its columns, ordered according to decreasing values of the eigenvalues. And since the covariance matrix $\mathbf{S_{AA}}$ is symmetric, $\mathbf{V}$ forms an orthogonal matrix (non-singular, $\mathbf{V^{-1}} = \mathbf{V^T}$, det $\mathbf{V} = \pm 1$, respects angles and distances between data points, columns are orthogonal, rank/determinant/trace-invariance under transformation). Furthermore this transformation diagonalizes the covariance matrix, with the eigenvalues as the diagonal elements.

-Dimensionality reduction with keeping the maximum variance is then done by just retaining only the first few PCs which account for most of the variance: a mapping matrix $\mathbf{B}$ is constructed with the first few PCs as the columns. It is in fact a mapping/projection of the data points into a subspace, where the variance of the data is maximal along the subspace basis-vectors (which are the eigenvectors in the original space).

Lots of alternative derivations and different optimality criteria exist, leading to the same result. It has been shown that the mapping to a LOWER q-dimensional subspace, denoted by $\mathbf{y} = \mathbf{B^T x}$, is obtained in the optimal sense of maximizing:

- the determinant of $\mathbf{S_{AA}}$ (which is a sort of generalized variance; the hypervolume that the columns of the covariance matrix span is thus being maximized) or

- the trace of matrix $\mathbf{S_{AA}}$ (which is the total variation) or
- the sum of squared distances between all pairs of observations in the subspace, or
- the sum of squared perpendicular distances of the variables to the subspace (so PC1 is best-fitting line, PC1 and PC2 best-fitting plane, etc...) or
- minimizing the sum of square error between x and y,

by taking for $\mathbf{B}$ the first q columns of the matrix $\mathbf{V}$.

The variables that contribute most in the first few pc's are explaining most of the variation, and can be regarded as essential. Also the grouping of the largest/smallest contributions in a pc can be an indication of new future variables which should be chosen in experiments. So the pc's can be used to assess the importance of the variables. Also the fact that it is possible to chose the most important components and discard the rest, it gives to PCA a better generalization characteristic, it means better prediction. PCA is not a real input selection technique in the sense that it gives us a method to discard unimportant input variables, since all of them are used in the construction of the pc's. It is only indirectly that there is dimensionality reduction.

2.4.3 Algorithm

Given the set of linear equations

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e} \tag{23}$$

PCR is a procedure, where the regression vector x which relates the vector y with the matrix A is computed based on PCA.

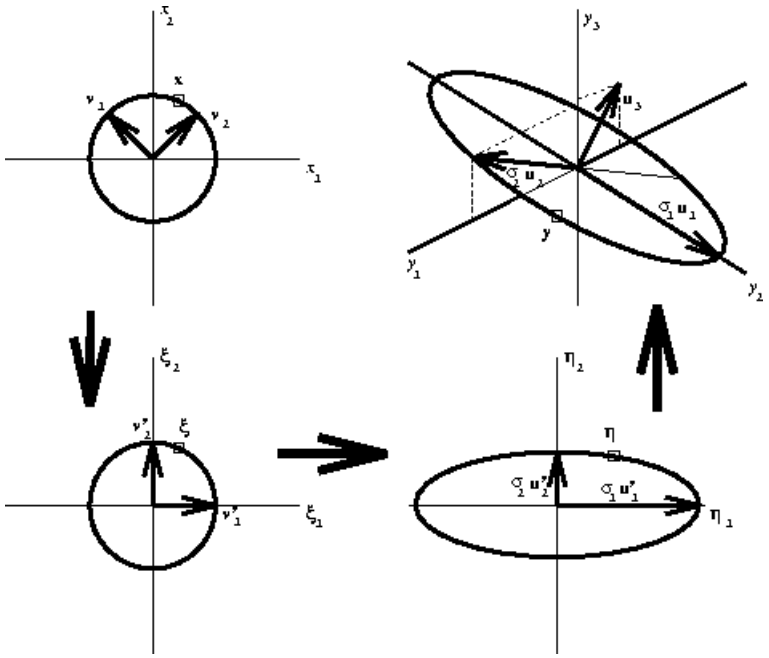


Figure 1: An intuitive picture of the effect of the different factor matrices to all vectors on the unit sphere.

Derivation

As we know the best solution for (23) in the sense of Least Squares is given by:

$$\mathbf{x} = (\mathbf{A^T A})^{-1}\mathbf{A^T y} \tag{24}$$

hence (24) can be extend to PCA, by making an eigenvalue decomposition of the symmetric matrix $A^T A$ as follows:

$$\begin{aligned} \mathbf{A^T A} &= \mathbf{V \Lambda V^T} = \mathbf{v_1}\lambda_1\mathbf{v_1^T} + \mathbf{v_2}\lambda_2\mathbf{v_2^T} + \ldots + \mathbf{v_p}\lambda_p\mathbf{v_p^T} \\ \mathbf{x} &= \mathbf{V \Lambda^{-1} V^T A^T y} \end{aligned} \tag{25}$$

where V is a squared and orthonormal matrix, which contains the eigenvector of $A^T A$; and Λ is a squared and diagonal matrix, which contains the eigenvalues of $A^T A$ ordered such that $\lambda_1 > \lambda_2 \ldots > \lambda_p$.

Taking into account the previous derivation, in PCR it is possible to choose as much components as you want in order to find a good predictor regression vector x in the PCA sense.

The Matlab implementation of that algorithm can be found in the appendix (B.1).

Practical issues

-Instead of using the covariance matrix, one could also use the correlation matrix, with normalized variables α. In practice, it is generally convenient to work with the variables in standard form (zero mean, unit variance). In such cases "the covariance matrix is identically a correlation matrix". It is advised to use correlation matrices.

	correlation matrix	covariance matrix
advantage	results of different analyses are comparable	matrix is that it makes statistical inference and interpretation more straightforward
disadvantage	has very different eigenvectors/eigenvalues and there exist no simple relationship with those of the corresponding covariance matrix possibly amplification of noise	variables with large values will tend to dominate the first PCs, which is inappropriate when having variables in different units; so PCA with the covariance matrix is sensitive to the scaling of the variables

-Near-zero or zero eigenvalues mean that there is a near-exact or exact linear relationship between some variables: these variables should be eliminated before doing PCA, since they are redundant. Nevertheless, the last few PCs will identify these relationships.

- How many PCs to retain? No objective criterion: only ad hoc guidelines are used:

- Look at T, the cumulative percentage of total variation for the first kth PCs and retain those that reach a certain threshold, e.g. 95%. In fact equivalent with looking at the error of the best k rank estimate (sum of first k terms in the spectral decomposition of the correlation matrix).

- Retain PCs with eigenvalues of the correlation matrix bigger than 0.7 or 1.

- Plot the error in function of the number of components on a validation set; the minimum will give an idea about the 'best' number.

-Outliers or influential observations may be detected from the few first and/or last PCs, where they form extreme data points with respect to the variance to the corresponding PC.

-Remark that the PCA criterion only selects the variables which preserve most of the variation in α. There is no consideration made of a dependent

variable y. This is a form of unsupervised learning, and this neglect of target data can lead to substantial suboptimal results in subsequent regression or classification. This disadvantage can partly be overcome by extending the data matrix with the extra dependent variable, this technique is known as Total Least Squares (**TLS**, [11]).

-PCA is limited by virtue of being a linear technique. It is therefore unable to capture more complex non-linear correlations. One could generalize the choice of linear functions of the variables to nonlinear functions of the variables. This leads to a PC that forms a so-called principal curve, through the center of the data. Thus such a curve is said to pass through the middle of a dataset if every point on the curve is the average of the observations projecting onto it. Principal curves can be regarded as a generalization of linear principal components. However, in that case we are confronted with a nonlinear optimization problem (high computational costs and existence of local minima).

2.5 Partial Least Squares (PLS)

2.5.1 Generalities

PLS is maximizing the squared covariance between a linear combination of the input variables α and a linear combination of the output variables y. The linear combinations in α space must be orthogonal to each other and the coefficient vectors are constrained to the unit sphere.

Geometrically, the orthogonal axes in α space are rotated such that the covariance with a linear combination in y space is maximal, it is

$$\textbf{Max} \;\; ||\alpha\mathbf{v}||||\mathbf{yw}||\textbf{cos}(\alpha\mathbf{v}, \mathbf{yw})$$

When the $\mathbf{Y}$ matrix consists of only one column (i.e. 1 output), one speaks of PLS1, and one successively maximizes the squared covariance between a linear combination of the input variables α and y itself, instead of a linear combination in y space, since it is only one-dimensional.

For the case of a multivariate y, one speaks of two-block PLS, or PLS2. One can then also require uncorrelated scores in y space.

In this section will be analyzed the case for several inputs, one output known as **PLS1.**

2.5.2 Mathematical Description

Derivation PLS1

Step 1

The first step in PLS is to look for a linear combination

$$u_1 = v_{11}\alpha_1 + v_{21}\alpha_2 + ... + v_{p1}\alpha_p = \mathbf{v_1^T}\alpha \tag{26}$$

which maximizes

$$[cov(\mathbf{v_1^T}\alpha, \mathbf{y})]^2 = [\mathbf{v_1^T A^T Y}]^2 = [\mathbf{v_1^T S_{Ay}}]^2 = \mathbf{v_1^T S_{Ay} S_{yA} v_1}. \tag{27}$$

Since $\mathbf{v_1}$ can be chosen arbitrarily high, we have to constrain it, for example by normalization:

$$\mathbf{v_1^T v_1} = ||\mathbf{v_1}|| = 1. \tag{28}$$

Using the technique of Lagrange multipliers, one maximizes

$$\mathbf{v_1^T S_{Ay} S_{yA} v_1} \quad \text{subject to} \quad ||\mathbf{v_1}|| = 1 \tag{29}$$

by maximizing

$$\phi_1(\mathbf{v_1}) = \mathbf{v_1^T S_{Ay} S_{yA} v_1} - \lambda_1(\mathbf{v_1^T v_1} - 1). \tag{30}$$

Differentiation with respect to $\mathbf{v_1}$ and setting equal to zero gives

$$\mathbf{S_{Ay} S_{yA}}\ \mathbf{v_1} - \lambda_1 \mathbf{v_1} = 0, \tag{31}$$

which is an eigenvalue problem for matrix $\mathbf{S_{Ay} S_{yA}}$, and gives for $\mathbf{v_1}$ the eigenvector that has the largest eigenvalue.

Step 2

In a second step, look again for a linear combination

$$u_2 = v_{12}\alpha_1 + v_{22}\alpha_2 + ... + v_{p2}\alpha_p = \mathbf{v_2^T}\alpha \tag{32}$$

which maximizes

$$[cov(\mathbf{v_2^T}\alpha, \mathbf{y})]^2 = [\mathbf{v_2^T A^T Y}]^2 = [\mathbf{v_2^T S_{Ay}}]^2 = \mathbf{v_2^T S_{Ay} S_{yA} v_2}, \tag{33}$$

but which is uncorrelated (i.e. orthogonal) with the previous linear function u_1:

$$cov(\mathbf{v_1^T}\alpha, \mathbf{v_2^T}\alpha) =< \mathbf{v_1^T}\alpha, \mathbf{v_2^T}\alpha >= \mathbf{v_2^T S_{AA} v_1} = 0, \tag{34}$$

Using the technique of Lagrange multipliers, one maximizes

$$(\mathbf{v_2^T A^T y})^2 \text{ subject to } ||\mathbf{v_2}|| = 1 \text{ and } v_1^T \mathbf{S_{AA}} v_2 = 0, \tag{35}$$

by maximizing

$$\phi_2(\mathbf{v_2}) = \mathbf{v_2^T S_{Ay} S_{yA} v_2} - \lambda_2(\mathbf{v_2^T v_2} - 1) - 2\mu_2(\mathbf{v_2^T S_{AA} v_1}). \tag{36}$$

Differentiation with respect to $\mathbf{v_2}$ and setting equal to zero gives

$$\mathbf{S_{Ay} S_{yA} v_2} - \lambda_2 \mathbf{v_2} - \mu_2 \mathbf{S_{AA} v_1} = 0. \tag{37}$$

To get rid of the term in μ_2, we premultiply (37) with $\mathbf{v_2^T}$:

$$\mathbf{v_2^T S_{Ay} S_{yA} v_2} - \lambda_2 \mathbf{v_2^T v_2} - \mu_2 \mathbf{v_2^T S_{AA} v_1}, \tag{38}$$

and solve for λ_2:

$$\lambda_2 = \mathbf{v_2^T S_{Ay} S_{yA} v_2}. \tag{39}$$

To get rid of the term in λ_2, we premultiply (37) with $\mathbf{v_1^T S_{AA}}$:

$$\mathbf{v_1^T S_{AA} S_{Ay} S_{yA} v_2} - \lambda_2 \mathbf{v_1^T S_{AA} v_2} - \mu_2 \mathbf{v_1^T S_{AA} S_{AA} v_1} = 0, \tag{40}$$

and solve for μ_2:

$$\mu = [\mathbf{v_1^T S_{AA}^2 v_1}]^{-1} \mathbf{v_1^T S_{AA} S_{Ay} S_{yA} v_2} = 0. \tag{41}$$

Substitution of the parameter μ by its solution in equation (37) yields

$$\mathbf{S_{Ay} S_{yA} v_2} - \lambda_2 \mathbf{v_2} - \mathbf{S_{AA} v_1} [\mathbf{v_1^T S_{AA}^2 v_1}]^{-1} \mathbf{v_1^T S_{AA} S_{Ay} S_{yA} v_2} \tag{42}$$

gives an eigenvalue formulation of the previous equation:

$$[\mathbf{I} - \mathbf{S_{AA} v_1} [\mathbf{v_1^T S_{AA}^2 v_1}]^{-1} \mathbf{v_1^T S_{AA}}] \, \mathbf{S_{Ay} S_{yA} v_2} = \lambda_2 \, \mathbf{v_2}, \tag{43}$$

or when we define $\mathbf{P_2} = [\mathbf{I} - \mathbf{S_{AA} v_1} [\mathbf{v_1^T S_{AA}^2 v_1}]^{-1} \mathbf{v_1^T S_{AA}}]$,

$$\mathbf{P_2} \, \mathbf{S_{Ay} S_{yA}} \, \mathbf{v_2} = \lambda_2 \mathbf{v_2}. \tag{44}$$

Next steps

The next steps are similar (only extra orthogonality constraints in α space are added) and the procedure can be compactly formulated in general terms as follows.

Given the previous directions $v_1, v_2, \ldots, v_{k-1}$, in PLS we look for a linear combination

$$u_k = v_{1k}\alpha_1 + v_{2k}\alpha_2 + ... + v_{pk}\alpha_p = \mathbf{v_k^T}\alpha \tag{45}$$

with $1 \leq k \leq p$, which maximizes

$$[cov(\mathbf{v_k^T}\alpha, \mathbf{y})]^2 = [\mathbf{v_k^T A^T y}]^2 = [\mathbf{v_k^T S_{Ay}}]^2 = \mathbf{v_k^T S_{Ay} S_{yA} v_k}, \tag{46}$$

but which is uncorrelated (i.e. orthogonal) with all previous linear functions:

$$cov(\mathbf{v_k^T}\alpha, \mathbf{v_i^T}\alpha) =< \mathbf{v_k^T}\alpha, \mathbf{v_i^T}\alpha >= \mathbf{v_k^T S_{AA} v_i} = 0, \qquad \forall i = 1, \ldots, (k-1). \tag{47}$$

Using the technique of Lagrange multipliers, one maximizes

$$\phi_k(\mathbf{v_k}) = \mathbf{v_k^T S_{Ay} S_{yA} v_k} - \lambda_\mathbf{k}(\mathbf{v_k^T v_k} - 1) - \mu(\mathbf{v_k^T S_{AA} V_{k-1}}), \tag{48}$$

with $\mathbf{V_{k-1}} = [\mathbf{v_1}|\, \mathbf{v_2}|\, \ldots\, \mathbf{v_{k-1}}]$ with dimension $p \times (k-1)$ and $\mu = [\mu_1,\ \mu_2, \ldots,\ \mu_{k-1}]$.

By analogously solving as above we obtain an eigenvalue problem for a rank one matrix

$$\mathbf{H_k\ S_{Ay} S_{yA} v_k} = \lambda_k \mathbf{v_k} \tag{49}$$

where the eigenvector v_k which corresponds to the maximum eigenvalue $\lambda_\mathbf{k}$ is:

$$\mathbf{v_k} = \frac{\mathbf{H_k S_{Ay}}}{||\mathbf{H_k SAy}||} \tag{50}$$

with

$$\mathbf{H_k} = \mathbf{I} - \mathbf{S_{AA} V_{k-1}}[\mathbf{V_{k-1}^T S_{AA}^2 V_{k-1}}]^{-1}\mathbf{V_{k-1}^T S_{AA}}. \tag{51}$$

Now in equation (49) it is possible to show that each column vector of V_{k-1} is scale independent as follows;

$$\mathbf{V_{k-1}} = [\mathbf{v_1}|\mathbf{v_2}|...|\mathbf{v_{k-1}}] \tag{52}$$

If V_{k-1} is scaled by different values by column we have

$$\mathbf{Vs_{k-1}} = [\mathbf{v_1}|\mathbf{v_2}|\ldots|\mathbf{v_{k-1}}] \begin{pmatrix} \gamma_1 & 0 & \ldots & 0 \\ 0 & \gamma_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \gamma_{k-1} \end{pmatrix}$$

replacing Vs_{k-1} by V_{k-1} in (51)

$$\begin{aligned} \mathbf{H_k} &= \mathbf{I} - \mathbf{S_{AA}V s_{k-1}}[\mathbf{Vs_{k-1}^T S_{AA}^2 Vs_{k-1}}]^{-1}\mathbf{Vs_{k-1}^T S_{AA}} \\ \mathbf{H_k} &= \mathbf{I} - \mathbf{S_{AA}V_{k-1}\Gamma}[\mathbf{\Gamma V_{k-1}^T S_{AA}^2 V_{k-1}\Gamma}]^{-1}\mathbf{\Gamma V_{k-1}^T S_{AA}} \\ \mathbf{H_k} &= \mathbf{I} - \mathbf{S_{AA}V_{k-1}\Gamma\Gamma^{-1}}[\mathbf{V_{k-1}^T S_{AA}^2 V_{k-1}}]^{-1}\mathbf{\Gamma^{-1}\Gamma V_{k-1}^T S_{AA}} \\ \mathbf{H_k} &= \mathbf{I} - \mathbf{S_{AA}V_{k-1}}[\mathbf{V_{k-1}^T S_{AA}^2 V_{k-1}}]^{-1}\mathbf{V_{k-1}^T S_{AA}}. \end{aligned} \quad (53)$$

Therefore the column vectors of V_{k-1} can be scaled without affecting (49). It has to be pointed out, because this is a very important issue for the algorithm simplification.

Summary

Given p variables in α: $\alpha_1, \alpha_2, \ldots, \alpha_p$. Suppose that $v^T\alpha$ is a linear combination in the α variables, where v is a $(p \times 1)$ vector of coefficients.

In PLS1, we have an iterative algorithm, where one solves the following optimization problem:

$$\begin{aligned} \max_{\substack{\mathbf{v} \in \mathbb{R}^\mathbf{p} \\ \|\mathbf{v}\| = 1 \\ \mathbf{v^T S_{AA} V_{k-1}} = \mathbf{0^T}}} \quad & \left[cov(\mathbf{v^T}\alpha, \mathbf{y})\right]^2 = \mathbf{v_k} \end{aligned} \quad (54)$$

where $1 \leq k \leq p$ and $\mathbf{V_{k-1}} = [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{k-1}}]$ with dimension $p \times (k-1)$.

It turns out that at every iteration step one looks for the PC of

$$\mathbf{H_k\ S_{Ay}S_{yA}} = [\mathbf{I} - \mathbf{S_{AA}V_{k-1}}[\mathbf{V_{k-1}^T S_{AA}^2 V_{k-1}}]^{-1}\mathbf{V_{k-1}^T S_{AA}}]\ \mathbf{S_{Ay}S_{yA}}. \quad (55)$$

We can regard the $(p \times p)$ matrix

$$\mathbf{M_k} = \mathbf{S_{AA}V_{k-1}}[\mathbf{V_{k-1}^T S_{AA}^2 V_{k-1}}]^{-1}\mathbf{V_{k-1}^T S_{AA}} \quad (56)$$

as a projection matrix that projects every vector on the column-space of $\mathbf{S_{AA}V_{k-1}}$. $\mathbf{M_k}$ spans also the same column space.

Likewise, the $(p \times p)$ matrix

$$\mathbf{H_k} = \mathbf{I} - \mathbf{S_{AA}V_{k-1}}[\mathbf{V_{k-1}^T S_{AA}^2 V_{k-1}}]^{-1}\mathbf{V_{k-1}^T S_{AA}} \quad (57)$$

can be regarded as the projection matrix that projects every vector on the orthogonal complement of the column-space of $\mathbf{S_{AA}V_{k-1}}$. $\mathbf{I} - \mathbf{M_k}$ spans the orthogonal column space.

2.5.3 Algorithm

For the PLS1 regression algorithm, factor extraction will be used as follows:

Problem Formulation

A regression using factor extraction computes the orthogonal factor score matrix

$$\mathbf{U} = \mathbf{AV}, \quad where\ A \in \mathbb{R}^{n\times p}\ and\ U \in \mathbb{R}^{n\times k} \tag{58}$$

for an appropriate weight matrix $V \in \mathbb{R}^{p\times k}$ $1 \leq k \leq p$, which maximize the covariance between A and y.
Then, it follows that the A matrix and the y vector can be decomposed in the linear regression models

$$\mathbf{y} = \mathbf{UQ^T} + \mathbf{E} = \mathbf{u_1 q_1^T} + \mathbf{u_2 q_2^T} + ... + \mathbf{u_k q_k^T} + \mathbf{E}, \quad Q \in \mathbb{R}^{1\times k} \tag{59}$$

$$\mathbf{A} = \mathbf{UP^T} + \mathbf{F} = \mathbf{u_1 p_1^T} + \mathbf{u_2 p_2^T} + ... + \mathbf{u_k p_k^T} + \mathbf{F}, \quad P \in \mathbb{R}^{p\times k} \tag{60}$$

where P and Q are matrices of regression coefficients (factor loading matrices) for U, and E and F are residuals terms. Once the loadings q are computed, the above regression model is equivalent to

$$\mathbf{y} = \mathbf{Ax} + \mathbf{E} \tag{61}$$

where

$$\mathbf{x} = \mathbf{V}(\mathbf{P^T V})^{+}\mathbf{Q^T} \tag{62}$$

with $(x)^+$=pseudo-inverse of x, see (25).

Pseudo-code implementation

Taking into account the above formulation and the optimization problem, it is possible to describe the PLS1 algorithm as follows:

Since it was shown in eq. (53) that V_{k-1} could be scaled without affecting v_k, and in order to make the calculation easier, we can scale v_k in such a way that the matrix U be orthonormal, i.e. $V_k = \frac{v_k}{||Av_k||}$. Therefore, the computation of the loading matrices P and Q becomes very simple:

$$\mathbf{p_k} = \mathbf{A^T u_k} = \mathbf{A^T A V_k} = \mathbf{S_{AA} V_k} \tag{63}$$

$$\mathbf{q_k} = \mathbf{y^T u_k} = \mathbf{y^T A V_k} = \mathbf{S_{Ay}^T V k} \tag{64}$$

so, replacing P in equation (51) we have

$$\mathbf{H_k} = \mathbf{I} - \mathbf{P_{k-1}}[\mathbf{P_{k-1}^T P_{k-1}}]^{-1}\mathbf{P_{k-1}^T} \tag{65}$$

now, in order to avoid the computation of $(P_{k-1}^T P_{k-1})^{-1}$, in each iteration we can subtract the previous p_{k-1} and q_{k-1} vectors from A and y respectively. Therefore in each iteration one get a new y_k vector and a new A_k matrix. This can be done without violate the constraints in eq. (54).

1. Initialization
   - $S_{AA} = A^T A$
   - $S_{Ay} = A^T y$
   - $v_1 = S_{Ay}$
2. For $k = 1$ to NC (NC = Number of Components)
   - Compute $V_k = \frac{v_k}{||Av_k||}$
   - Compute $p_k = S_{AA} V_k$
   - Compute $q_k = S_{Ay}^T V_k$
   - Compute $S_{AA} = S_{AA} - p_k p_k^T$
   - Compute $S_{Ay} = S_{Ay} - p_k q_k$
   - Compute $H_{k+1} = I - \frac{P_k P_k^T}{P_k^T P_k}$
   - Compute $v_{k+1} = H_{k+1} S_{Ay}$
3. Regression vector and estimated output computation
   - $x = V(P^T V)^+ Q^T$
   - $ye = Ax$

see appendix (B.2) for a matlab code.

Remarks

-Like in PCR, the goal in PLS1 is to find an optimal basis transformation in *alpha* space. Yet in PLS1, the successive basisvectors must account not only for most of the variation in the data cloud in the α space, but they must take also the variation in the y space into consideration. So in PLS1 the directions will be forced more into the direction of the one output. (In PLS2 the directions will be forced more into the direction of largest variation in the y cloud.) At the other extreme a PCA analysis on the y variables would just give successive directions of largest variation in y space. When effectively ignoring the inputs, one speaks of reduced rank regression (RRR), which is PCA on the output. Thus PLS is a sort of compromise between these approaches and gives a direction between those of PCA and RRR.

- Dimensionality reduction with keeping the maximum variance is then done by computing iteratively the directions (components) wich maximizes the covariance between the input variables with the residuals ($e = y - Ax$) in each iteration, till either a residual error is reached or the components do not account for too much variance. This is done via a singular value decomposition of the matrix A projected into the residuals.

-One can translate the mathematical derivation of the iterative pls-procedure in more loose terms as follows. At each step the principal component is taken from a -on first sight- obscure matrix. This matrix can be seen as a weighted covariance matrix, projected on the orthogonal complement of $S_{AA}V_{k-1}$. The covariance matrix S_{AA} is 'weighted' by the output in the sense that data in the A matrix are weighed by the magnitude of the values in the Y matrix, which gives A^TYY^TA. The projection means a restriction in possibilities for the principal component, due to the orthogonality constraint for successive principal components. In other words: at each step one searches for a pls-direction v_k in the column space of the weighted covariance matrix, which is weighted by the output, and at the same time constrained to be orthogonal to the column space of $S_{AA}V_{k-1}$. This latter column space is the space spanned by the previous pls-directions, merely expressed in a particular basis determined by S_{AA}.

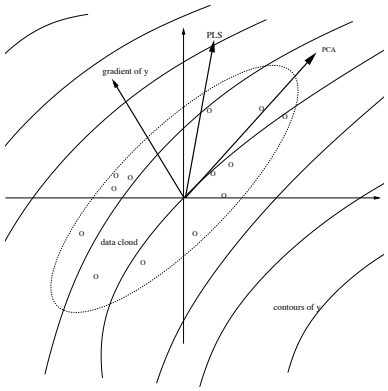


Figure 2: The PLS direction will also take the direction of largest variation in y into account, while the PCA direction is purely concerned with the direction of largest variation in x.

In PLS the covariance between a linear combination in x space and a linear combination in y space is maximized. This covariance is a scalar product:

$$cov(\mathbf{v^T x}, \mathbf{w^T y}) = \mathbf{v^T} cov(\mathbf{x}, \mathbf{y})\mathbf{w} = \mathbf{v^T A^T Y w} = < \mathbf{Av}, \mathbf{Yw} > = ||\mathbf{Av}||||\mathbf{Yw}||\mathbf{cos}(\mathbf{Av}, \mathbf{Yw}) \tag{66}$$

By maximizing the covariance one tries to minimize -at the same time- the magnitudes of the linear combinations in both spaces and the angle between them.

-In the case of several inputs and outputs, are referred to [8]; Analso you can find and algorithm in appendix (B.3)

2.6 Canonical Correlation Analysis (CCA)

Reference: R. Gittins, Canonical analysis, Springer-Verlag, 1985.

2.6.1 Generalities

CCA is maximizing the correlation between a linear combination of the input variables x and a linear combination of the output variables y. At most min(p,q) directions can be calculated. The coefficient vectors are chosen so that the linear combination vectors in α and y space are constrained to the unit sphere and orthogonal to each other.

For univariate $\mathbf{Y}$ we denote the method by **CCA1**, and for the multivariate case as **CCA2**, just like in PLS. In this section will be analyzed the **CCA1** case, and if you are interested in the case for several inputs-several outputs (**CCA2**), please see appendix (C).

Geometrical interpretation

Consider the column vectors that span a p-dimensional α space and separate a q-dimensional y space. Then a unit vector (linear combination) in α and y space is sought, so that the correlation of the projections of the data on these vectors is maximal. The next unit vectors are chosen orthogonal to the first.

One can equivalently consider the α and y subspaces in one big p+q-dimensional space, and look for unit vectors in α and y subspace for which the angle is minimal (=canonical angles or correlations). This comes from the fact that we maximize $cor(\mathbf{v^T}\alpha, \mathbf{w^T y}) = cos(\mathbf{v^T}\alpha, \mathbf{w^T y})$, when $\mid v^T\alpha|| = 1$ and $|w^T y|| = 1$. The next unit vectors are chosen in the subspaces (of the column-space generated by A) orthogonal to the first unit vectors. This process goes on until one of the subspaces is spanned by the basis of its unit vectors (=canonical basis with canonical variates).

2.6.2 Mathematical Description

Given p variables in $A = [a_1|\ a_2|\ ...\ a_p|]$ with zero mean, where $A \in R^{(n \times p)}$, and one variable in $y = y_1$ with zero mean, where $y \in \mathbb{R}^{(nx1)}$. Now, Let $u_k = Av_k$ be a linear combination in the A variables, where $v_k \in \mathbb{R}^{(px1)}$, and let $z = wy$ be the product between the vector y and the scalar w.

In **CCA1** the following optimization problem is solved:

For $k = 1$ to s **...** $min(p)$

$$\max_{\mathbf{v_k} \in \mathbb{R}^{\mathbf{p}}} \quad \mathbf{cor(u_k, z)^2} \tag{67}$$

Subject to

$$\begin{aligned} \| u_k \| &= 1 \\ \| z \| &= 1 \\ \mathbf{V_{k-1}^T S_{AA} v_k} &= 0 \end{aligned} \tag{68}$$

where,

$$\begin{aligned} \mathbf{V_{k-1}} &= \mathbf{[v_1\ v_2\ ...\ v_{k-1}]} \in \mathbb{R}^{\mathbf{(px(k-1))}} \\ \mathbf{S_{AA}} &= \mathbf{A^T A} \in \mathbb{R}^{\mathbf{(pxp)}} \\ 1 \leq k \leq s &= \mathbf{min(p)} \end{aligned}$$

taking into account the above constraints, notice that

$$\mathbf{cor(u_k, z)^2} = \mathbf{(v_k^T S_{Ay} w)^2} \tag{69}$$

where, $S_{Ay} = A^T y \in \mathbb{R}^{(px1)}$

therefore the optimization problem in (67) becomes:

For $k = 1$ to s **...**

$$\max_{\mathbf{v_k} \in \mathbb{R}^{\mathbf{p}}} \quad \mathbf{v_k^T S_{Ay} w^2 S_{Ay}^T v_k} \tag{70}$$

Subject to

$$\begin{aligned} \| \mathbf{u_k} \| &= 1 \\ \| \mathbf{z} \| &= 1 \\ \mathbf{V_{k-1}^T S_{AA} v_k} &= 0 \end{aligned} \tag{71}$$

where, $1 \leq k \leq s = min(p)$

2.6.3 Algorithm

The algebraic problem therefore is to find a v_k to maximize (70) subject to (71). Using the technique of Lagrange Multipliers one maximizes the function $\phi(v_k)$ as follows

$$\underset{\mathbf{v_k} \in \mathbb{R}^p}{\mathbf{max}} \qquad \phi(\mathbf{v_k}) = \mathbf{w^2 v_k^T S_{Ay} S_{Ay}^T v_k} - \lambda_{\mathbf{k1}}(\mathbf{v_k^T S_{AA} v_k - 1}) - \lambda_{\mathbf{k2}}(\mathbf{V_{k-1}^T S_{AA} v_k}) \tag{72}$$

where, $w^2 = \frac{1}{y^T y}$.

Taking the partial derivatives of $\phi(v_k)$ with respect to v_k and equating to the null vector, we obtain

$$\frac{\partial \phi(\mathbf{vk})}{\partial \mathbf{v_k}} = \mathbf{w^2 S_{Ay} S_{Ay}^T v_k - \lambda_{k1} S_{AA} v_k - \lambda_{k2} S_{AA} V_{k-1}} = \mathbf{0} \tag{73}$$

Now, premultiplying (73) by $\mathbf{V}_{k-1}^T$ and by v_k^T gives

$$\mathbf{V_{k-1}^T} \frac{\partial \phi(\mathbf{v_k})}{\partial \mathbf{v_k}} = \mathbf{w^2 V_{k-1}^T S_{Ay} S_{Ay}^T v_k - \lambda_2 = 0} \tag{74}$$

$$\mathbf{v_k^T} \frac{\partial \phi(\mathbf{v_k})}{\partial \mathbf{v_k}} = \mathbf{w^2 v_k^T S_{Ay} S_{Ay}^T v_k - \lambda_{k1} = 0} \tag{75}$$

then

$$\lambda_{\mathbf{k2}} = \mathbf{w^2 V_{k-1}^T S_{Ay} S_{Ay}^T v_k} \tag{76}$$

$$\lambda_{\mathbf{k1}} = \mathbf{w^2 v_k^T S_{Ay} S_{Ay}^T v_k = (n-1)^2 cov(u_k, z)^2 \geq 0} \tag{77}$$

Substituting (76) into (73) we have

$$\frac{\partial \phi(\mathbf{vk})}{\partial \mathbf{v_k}} = \mathbf{w^2 S_{Ay} S_{Ay}^T v_k - \lambda_{k1} S_{AA} v_k - w^2 S_{AA} V_{k-1} V_{k-1}^T S_{Ay} S_{Ay}^T v_k} = \mathbf{0} \tag{78}$$

Making some manipulations on (78) we obtain a generalized eigenproblem on v_k as follows

$$\Big((\mathbf{I} - \mathbf{S_{AA}V_{k-1}V_{k-1}^T})\mathbf{S_{Ay}S_{Ay}^T} - \mathbf{S_{AA}}\lambda_{\mathbf{k1}}\Big)\mathbf{v_k} = \mathbf{0} \tag{79}$$

Notice for the case $k = 1$ $V_0 = 0$, that the equation (79) becomes

$$(\mathbf{S_{AA}^{-1}S_{Ay}S_{Ay}^T} - \lambda_{\mathbf{k1}})\mathbf{v_k} = \mathbf{0} \tag{80}$$

where the solution is:

$$\mathbf{vk} = \mathbf{S_{AA}^{-1}S_{Ay}} \tag{81}$$

which is the Least Squares solution, therefore we can not iterate anymore because the best approximation was already found. It means that there is not more information we can catch in other directions.

Remarks

The purpose of canonical correlation analysis is to find canonical variables (linear combinations of the variables) in each set such that the correlation between the corresponding canonical variables in each set are as large as possible.

If there is only a single variable in y space then CCA reduces to multiple regression analysis with the coefficients for the linear combination of the variables becoming standardized partial regression coefficients. Thus, CCA can be viewed as a generalization of multiple regression. The components can be calculated straightforward by solving the generalized SVD problem.

It is possible for the linear combinations to be quite different from those obtained from PLS2 because different criteria are being optimized. It can thus happen that the linear combinations with the highest correlation between the two sets need not explain much of the covariance between the two sets of variables.

In CCA the correlation between a linear combination in x space and a linear combination in y space is maximized. This correlation is a scalar product:

$$cor(\mathbf{v^T x}, \mathbf{w^T y}) = \frac{\mathbf{v^T}cov(\mathbf{x}, \mathbf{y})\mathbf{w}}{var(v^T x)var(w^T y)} = cos(Av, Yw) \tag{82}$$

By maximizing the correlation one tries to minimize the angle between the linear combinations in both spaces.

2.7 Discussion about differences and Similarities LS, PCR, PLS, CCA

Remark that for PLS the coefficient vectors must be on the unit sphere. If we change this constraint and impose that the linear combinations must lie on the unit sphere (or, equivalently, have unit variance), plus that we maximize correlation instead of covariance, then we speak of Canonical Correlation Analysis (CCA). The difference is in the choice of the scaling of the coefficients of the linear combinations (which are only determined up to a proportionality constant), and the choice of matrix.

CCA is scale independent, so choosing different units for some variable does not affect the result of CCA, in contrast with PLS, PCA and LS, where normalization of the variables is suggested to remedy this at the cost of introducing possibly noise-amplification. Furthermore all methods are translation invariant, since this doesn't affect the (co)variance or correlation.

When the goal is to predict a signal as well as possible in a least square error sense, the basis must be chosen so that this error measure is minimized. This corresponds to a low-rank approximation of multivariate linear regression (MLR). This is also known as reduced rank regression or as redundancy analysis.

In general, these 4 different criteria for selecting basis vectors lead to four different solutions. But the problems are related to each other and can be formulated in very similar ways. An important problem which is directly related to the situations discussed above is the generalized eigenproblem or two-matrix eigenproblem. It is shown that the 4 important methods discussed above (principal component regression (PCR), partial least squares (PLS), canonical correlation analysis (CCA), least squares (LS) and also multivariate linear regression (MLR) emerge as solutions to special cases of the generalized eigenproblem. We refer for the elaboration of this idea to the thesis of Borga (see [4]).

Schematically we can order some remarks about the different methods:

	Advantages	Disadvantages
PCR	Equivalent to 1 SVD computation	Does not take into account the dependent variables
	Widely used, relatively simple	Not scale invariant; variables must be scaled
PLS	Flexibility of algorithm	Distributional properties of estimates not known
	Size dimensions do not matter e.g. there can be more variables than observations	Sometimes difficult to interpret loadings
	Some versions of the algorithm can accept missing data	Not scale invariant; variables must be scaled
	Can fit data with fewer components than other modeling approaches	
	For PLS1 it is not necessary to make eigenvalue decompositions, and the computation of the loading matrices P and Q becomes a multiplication, due to the fact that the U matrix can be orthonormal without affecting the results see eq. (53).	
	With PLS1 is possible to deal with the problem of several outputs (PLS2), by making PLS1 over each output independently.	
CCA	Inherently scale invariant	It is useful just for several inputs-several outputs, otherwise for several inputs-one output it is Least Squares.
LS	Gives the best fit	Very bad for prediction due to the fact that it always gives the best approximation to the training data, so, it over fits the prediction model.

Table 1: General Remarks PCR PLS CCA LS

	usefulness	user friendliness	accuracy	implemenability	maintanibility
PLS	high (recommended)	high	high	high	high
PCR	high	high	medium	high	high
CCA	low (1 output) high(>1 output)	high	low	high	high
LS	low	high	low	high	high

Table 2: Classification of the different used techiques for the Ammonia prediction case

3 Non-Linear Modeling Techniques

4 Input Selection

4.1 Automatic relevance detection

An interesting method that can be used for input selection in the nonlinear static case is ARD.

Automatic Relevance Detection is a bayesian method to determine the relevance of the different input variables; the method can be used for input-selection. Input selection is crucial for retrieving an optimal generalization capacity, because irrelevant variables have random correlations with the output, causing overfittting.

Automatic Relevance Determination is a model-dependent algorithm, in that the algorithm asks the repeated training of the mode, and because it uses model specific derivations.

In subsect. 4.1.1, we look at the high-level bayesian framework; in subsect. 4.1.2 we give a short description of ARD applied in Multi Layer Perceptrons. In subsect. 4.1.3, we give some test-results.

4.1.1 The Bayesian Framework

Bayes´ law:

$$P(w|D,H) = \frac{P(D|w,H) * P(w|H)}{P(D|H)} \tag{83}$$

$$or \tag{84}$$

$$\text{posterior probability} = \text{likelihood} * \text{prior} / \text{evidence} \tag{85}$$

We introduce a regularization vector α in this formula formula :

$$P(w|\alpha, D, H) = \frac{P(D|\alpha, w, H) * P(w|\alpha, H)}{P(D|\alpha, H)}$$

α determines the inverse of the relevance of the weight of the model associated with the resp. inputs. In the case of Multiple Layer Perceptrons (**MLP**'s), α_i is the decay term on the class of weights connected with input x_i. In the case of Support Vector Machines, α is the decay on the lagrange multiplier of each support vector (input vector).

To determine α, we apply again bayes rule:

$$P(\alpha|D, H) = \frac{P(D|\alpha, H) * P(\alpha|H)}{P(D|H)}$$

The posterior probability has to be maximized. $P(D|H)$ is independent of α, and $P(\alpha|H)$ is constant, this terms are not relevant to calculate the maximum posterior probability. So we get α by:

$$max_\alpha(P(D|\alpha, H)) \tag{86}$$

Remark the equality of this equation with the *evidence* of the previous equation. This provides a way of searching the most likely α: iteration between re-estimating α given the w's, and re-training the w's, given α, will converge to the trained model with the most likely α.

A model specific proof that the most likely α is found, can be derived. (see subsect. 4.1.2) ([1])

Input selection can be done in 2 ways:

- α is determined, and the n inputs x_i's that corresponds with the least relevant weights w_i, are removed. n can be determined by the performance of the resulting models on a test-set.
- α is determined, and the input x_i that corresponds with the least relevant w_i, is removed. This process is repeated n times, until the performance of the resulting model on a test-set falls under a threshold.

The second way gives better results, because of the recalculation of α after each single adaption of the model. The first way is less computational intensive. The total high-level algorithm for the second way, looks like this:

```
X = inputs;
while ( performance(test set) > treshold ),
  initialise alpha.
```

```
  while not stopcriterium,
    compute w by P(w|\alpha,D,H) \% training
    re-estimate alpha, max_alpha ( P(D|\alpha,H) ) \% evidence
  end
  X = X without x_i, corresponding with least relevant w_i
end
```

It is clear that this leads to a heavy computational process, because of the intern training of the model. The algorithm is not an independent input-selection, but is a kind of model optimalization, in that it uses repeated training s of the model for increasing the generalization ability.

4.1.2 ARD for MLP

ARD Is applied for classical Multi Layer Perceptrons.(**MLP**) ([7]) A supervised neural network is a non-linear parametrized mapping from an input x to an output
$y = MLP(x; w, A)$, where A is the architecture of the net, and w is the vector containing the weights (parameters) of the connections in the net. We consider here only the regression case, because of the project requirements. Classification can be seen as a special case of regression. We also restrict to the case of networks with a single hidden layer.

A neural network is be described by following equations:

$$a_j^{(1)} = \sum_l w_{jl}^{(1)} x_l + \theta_j^{(1)}; h_j = f^{(1)}(a_j^{(1)}) \tag{87}$$

$$a_i^{(2)} = \sum_j w_{ij}^{(2)} h_j + \theta_i^{(2)}; y_i = f^{(2)}(a_i^{(2)}) \tag{88}$$

with y_i the outputs of the network, x_l the inputs, the θ's the biases, w_{ij} the weight of the connection between neuron i of the previous layer, and neuron j of the next layer. f is for example, $f^{(1)}(a) = tanh(a)$,and $f^{(2)}(a) = a$. This network is trained on dataset $D = x^{(m)}, t_m^{(m)}$, by minimizing the objective function

$$E_D(w) = \frac{1}{2} \sum_m \sum_i (t_i^{(m)} - MLP(x^{(m)}; w, A))^2 \tag{89}$$

A regularization term is included:

$$M(w) = \beta E_D + \alpha E_W \tag{90}$$

$$\text{where } E_W(w) = \frac{1}{2} W^T W \tag{91}$$

This objective function is minimized by iteratively updating the w's, for example with back propagation. (see [3])

This classical theory is given a probabilistic interpretation; Equation 89 is interpreted as minus the log likelihood.

$$P(D|w,\beta,A) = \frac{1}{Z_D(\beta)} exp(-\beta E_D) \tag{92}$$

the sum-squared error of eq.89, corresponds to an assumption of gaussian noise (with $\sigma^2 = 1/\beta$) on the target variables. Z_D is a normalization term.

Similarly, the regulariser E_W is interpreted as the log prior probability over the weights w:

$$P(w|\alpha,A) = \frac{1}{Z_W(\alpha)} exp(-\alpha E_W) \tag{93}$$

so, a gaussian distribution is assumed over the weights w (with $\sigma^2 = 1/\alpha$). Again, Z_W is a normalization term.

The objective function $M(w)$ corresponds with:

$$P(w|D,\alpha,\beta,A) = \frac{P(D|w,\beta,A)P(w|\alpha,A)}{P(D|\alpha,\beta,A)} \tag{94}$$

$$= \frac{1}{Z_M} exp(-M_W) \tag{95}$$

The w found by minimizing $M(W)$ is thus interpreted as the most probable parameter vector W_{MP}. This concludes the conversion to the bayesian framework.

As seen in eq.86, to determine the most likely α, we need to maximize the evidence of eq.94. The evidence is known, if we can approximate the posterior of eq. 94. We impose the posterior equal to a single gaussian:

$$P(w|D|\alpha,\beta,A) \simeq \frac{1}{Z'_M} exp(-M(w_{MP}) - \frac{1}{2}(w - w_{MP})^T L(w - w_{MP})) \tag{96}$$

$$\text{where } L = -\nabla\nabla log P(w|\alpha,\beta,A)|w_{MP} \tag{97}$$

then the evidence is:

$$log P(D|\alpha,\beta,A) = log \frac{Z'_M}{Z_W(\alpha) Z_D(\beta)} \tag{98}$$

$$= -M(w_{MP}) - \frac{1}{2} logdet(\frac{A}{2\pi}) - log Z_W(\alpha) - log Z_D(\beta) \tag{99}$$

we need the maximum of this function. It is shown ([2]) that the most likely evidence fulfills certain conditions, namely:

- the maximum- condition for α is:

$$\frac{1}{\alpha_M P} = \frac{\sum_i w_i^{MP}}{\gamma} \tag{100}$$

  where w_{MP} is the weight vector that minimizes the objective function, (eq.90). γ is a measure for the number of well-determined parameters (it depends on the eigenvalues of the Hessian of eq.89).

- the maximum evidence value for β satisfies:

$$\frac{1}{\beta} = \frac{2E_D}{N - \gamma} \tag{101}$$

Equation 100 and eq.101 are used for a re-estimating procedure, calculating the α_{MP} and β_{MP}.

Now we return to ARD. The trick is to extend α - as a single value determining the variance of the gaussian distribution on the weights w - to a vector, with α_i containing the variance of the distribution on the weights $w_{ij}; j = 1..l$, associated with each different input x_i. The formulas are slightly changed, in that we mutate every term depending from α, to a summation of the term over the different α_i's.

The extra hyper-parameters will not cause more overfitting. The intuitive idea is that overfitting will only decrease by increasing an α_i (neutralizing the effect of an input). The worst case of overfitting is when the α's are low. The overfitting is then solely due to the weights w_i.

4.1.3 Some results of applying ARD

under construction (-:

5 Ammonia Experimental Section

5.1 Data Preprocessing

For the experimental test four(4) data sets of the Ammonia plant were used: PRBNS1, PRBNS2, PRBNS3, PRBNS4. As training data sets the first three were used, and for validation the fourth one. In order to avoid data scaling problems i.e. different unit systems, and data noise, all data sets were preprocessed as follows:

- Mean Subtraction: To linearize the model around the operation point
  $X_c = X - \bar{X}$
- Filtering; applied to all data sets: To filter noise
  Type of filter: Causal-Anticausal Low pass digital filter Butterworth 5th order.
  Cut-off frequency: 1.5 mHz.
  $$X_f = filtfilt(X_c,\ 1.5e-3Hz,\ low\ pass\ filter)$$
- Scaling to unit variance: To work with normalized data
  $X_s = \frac{X_f}{\sqrt{var(X_f)}}$, where $var(Xf) = \frac{X_f^T X_f}{n-1}$

5.2 Model Validation

See [6]

5.2.1 Quantitative validation and Component selection

The optimal number of principal components is chosen based on a validation statistic. Since the calibration models will be used to predict, a prediction error statistic, the Root Mean Square Error of Prediction (RMSEP) can be found by:

$$\mathbf{RMSEP} = \sqrt{\frac{\sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{x_i} - \hat{\mathbf{x}}_\mathbf{i})^\mathbf{2}}{\mathbf{n}}} \tag{102}$$

where n is the number of additional prediction samples that were not used in the model training step, x_i are the measured outputs and $\hat{x}_i$ are the predicted outputs. As n gets large the **RMSEP** will approach the prediction error of the population of all future samples.

On the other hand, the selection of the number of components is important because the model should describe the significant sources of variance but no over fit the data. A model is over fitting if it is including non-relevant

information in the model from the calibration set, such as non-random components in the noise or fluctuations due to experimental conditions. A model that is over fitting will have poorer prediction errors. Several schemes have been proposed for model selection given the **RMSEP** at different components. The simplest criterion is to choose the number of components corresponding to the first minimun in a curve of number of components vs. RMSEP. Further theories have been developed beyond this, because since the RMSEP is calculated with a finite number of samples, there is an error associated with it. Therefore in this paper the **PRESS** (Predictive Residual Error Sum of Squares) criterion which uses the variance term of **RMSEP** will be applied:

$$\mathbf{PRESS} = \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{x_i} - \mathbf{\hat{x}_i})^{\mathbf{2}} \tag{103}$$

First of all, we have to compute,

$$\mathbf{F(nc)} = \frac{\mathbf{PRESS(nc)}}{\mathbf{PRESS(nc^*)}}$$

where nc is the number of components, and nc^* is the number of components that yields the minimun PRESS. This statistic tells if the two PRESS values are significantly different. The optimal number of components then is the smallest value of nc that has a PRESS which is not significantly different from $PRESS(n^*)$.

Component Selection

In order to make a component selection for **PLS** and **PCR** PRBNS1, PRBNS2 and PRBNS3 were chosen as training data sets, and PRBNS4 as validation data set.

For the training step, PRBNS1, PRBNS2 and PRBNS3 were appended in one matrix, where:

$$\mathbf{A_T} = \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} \quad \mathbf{Y_T} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}$$

then, the new data sets were divided in 100 subsets in order to make a cross validation. Thus, we compute the PRESS leaving out one subset as validation data set, and the rest as training data set. This is repeated until all the data subset have been left out once. Afterwards we obtain 100 PRESS for each component, so, for the final result, we took the average of the PRESS per component in order to get an smoother curve, which give us a better idea about the number of components we can choose.

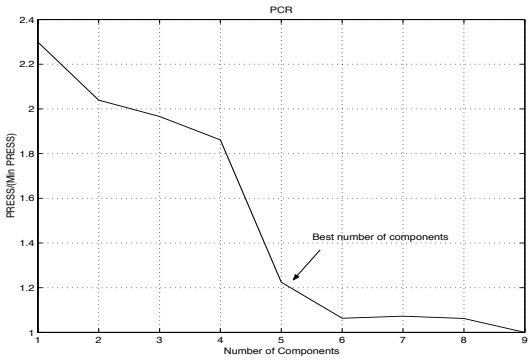


Figure 3: PCR-Component Selection

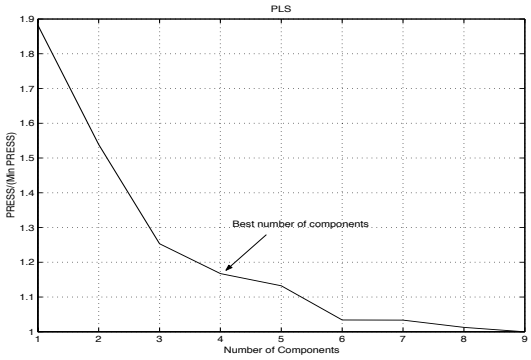


Figure 4: PLS-Component Selection

Therefore, we selected four(4) components for PLS and five(5) componentsfor PCR

5.2.2 Quantitative validation results

For the validation results PRBNS4 was used as validation data sets. Since The results are the same for **CCA1** and **LS**, just the **LS** results will be shown below.

In the figure (5) below are shown the results for the qualitative tests (see details in appendix A.1). There we can say that **PLS** is the best prediction model because it has the smallest prediction error (RMSEP), **LS** and **CCA** the best fitting model because they have the smallest fitting error (RMSEF). Spite of that and since we are more interested in prediction than fitting models, PLS is the best one, due to it is a much simpler model with the smallest prediction error (i.e. 4 PLS components vs. 5 PCR).

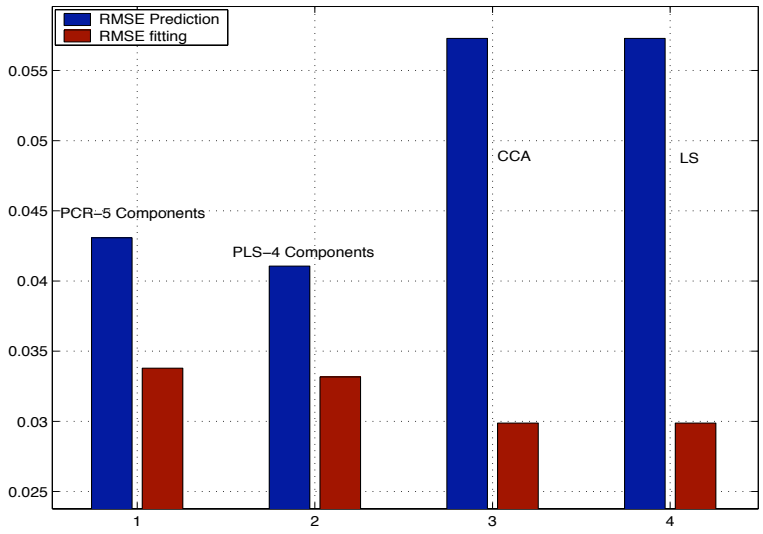


Figure 5: RMSE comparison

5.2.3 Qualitative validation results

Observe in the appendix (A.2.1), that the residual distribution is homogeneous into the whole range of the output values for the **LS-CCA1** case, while it is directly proportional to the outputs for the **PLS** and **PCR** case. It means that the **PLS**, **PCR** models predict better the values around the mean (operation point) than **LS-CCA1** . In appendix (A.2.2) we can observe the correpondence between the predicted and fitted output with the measured output. There the straight line is ideal model correspondence case. One can see that in the **LS-CCA1** the correspondence for the predicted values is very low in the whole range of the output, while in **PCR** and **PLS** the correspondence for predicted ouputs is higher for values smaller than $mathbf{3.1}$, it means that the process presents a nonlinearity around this point. Therefore, the biggest values are not well predicted maybe because those come from the non-linear characteristic of the process. Due to the low performance of the linear models, we can conclude that the process is very non-linear, in such case we recommend to use non-linear models in order to improve the performance of the prediction model.

Summarizing, The best method for **linear modeling** taking into account the quantitative and qualitative validation results, is Partial Least Squares **PLS**. Nevertheless, in order to improve the performance of the prediction model, it is better to use non-linear modeling like for instance; neural net-

works, support vector machines etc.

5.2.4 Table of Results

	PCR	**PLS**	**CCA1**	**LS**
Number of Inputs	9	9	9	9
Number of Components	5	4	-	-
RMSE Fitting	0.03379	0.03316	0.02988	0.02988
RMSE Prediction	0.04308	0.04106	0.05728	0.05728

Table 3: Comparative Results

A Figures

A.1 Quantitative test figures

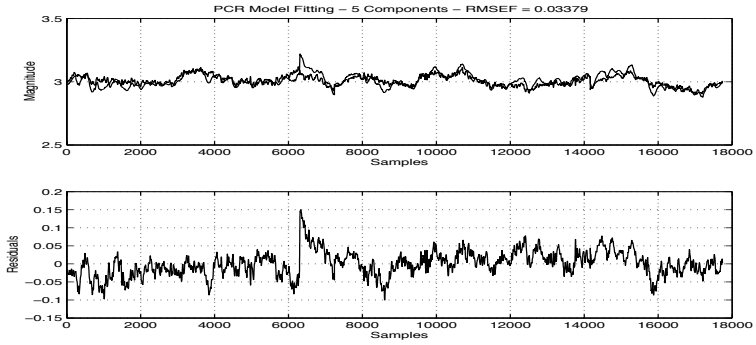


Figure 6: PCR-Fitting test

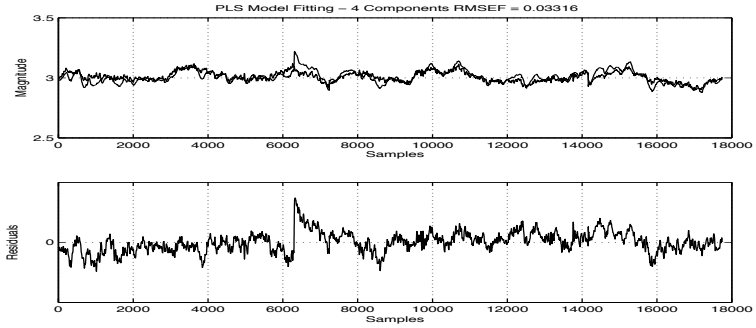


Figure 7: PCR-Fitting test

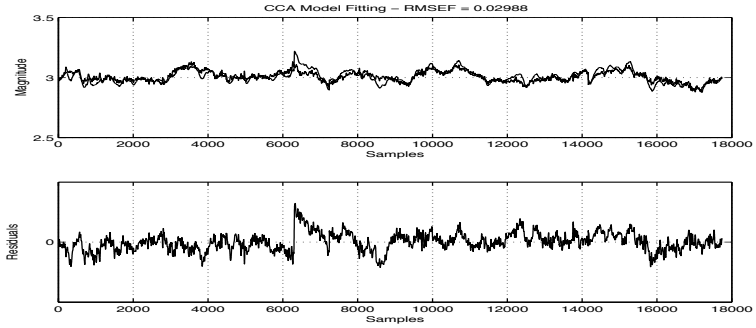


Figure 8: PCR-Fitting test

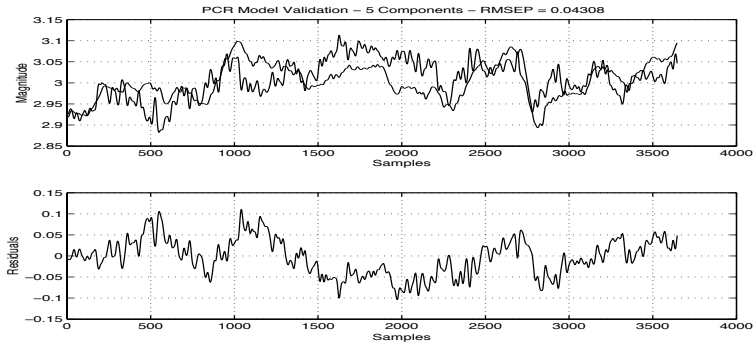


Figure 9: PCR-Validation test

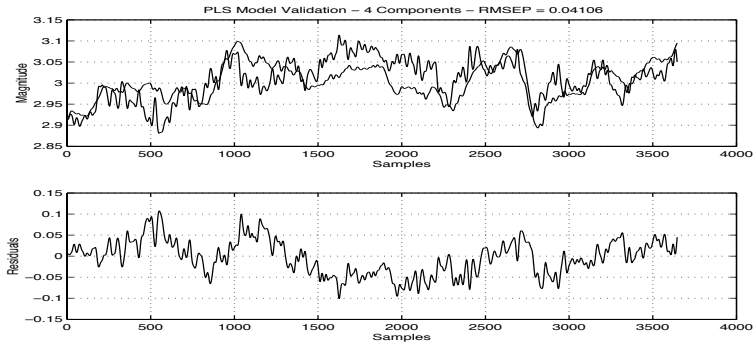


Figure 10: PCR-Validation test

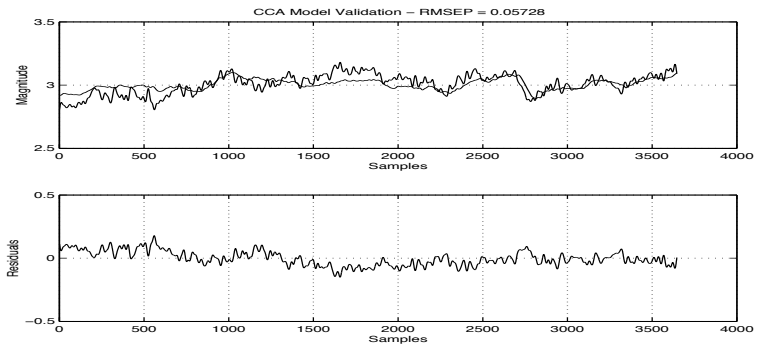


Figure 11: PCR-Validation test

A.2 Qualitative test figures

A.2.1 Residual Analysis (Validation test)

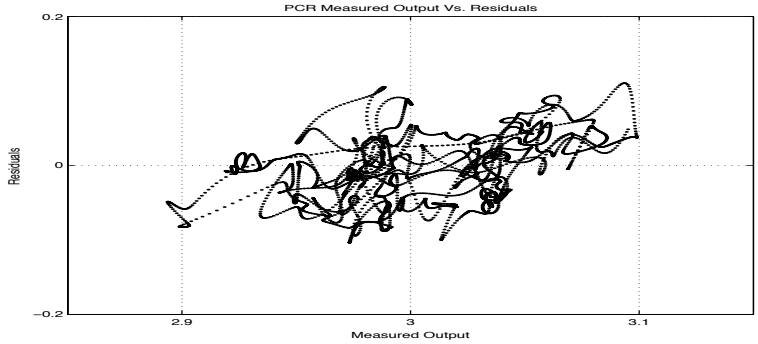


Figure 12: PCR Case

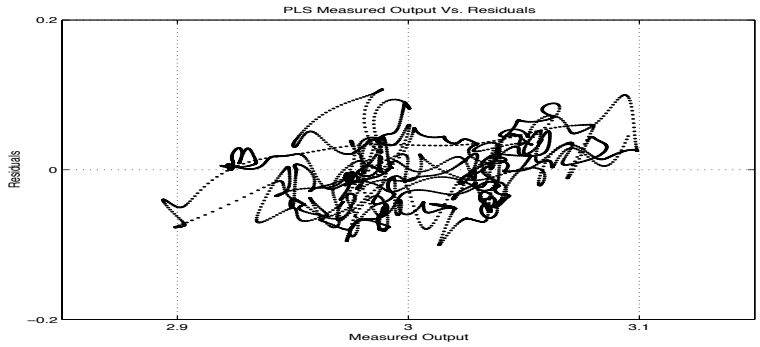


Figure 13: PLS Case

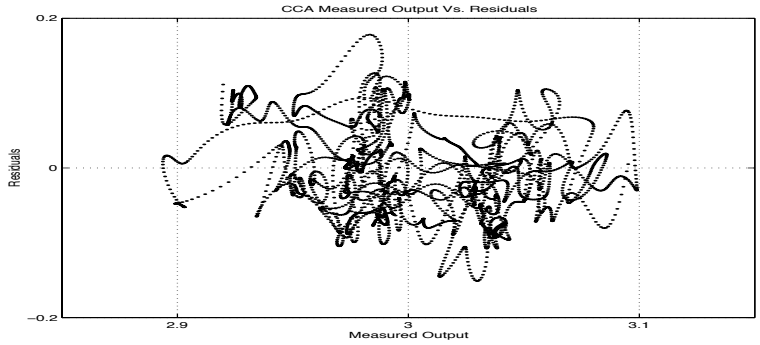


Figure 14: LS-CCA1 Case

A.2.2 Model Correspondence

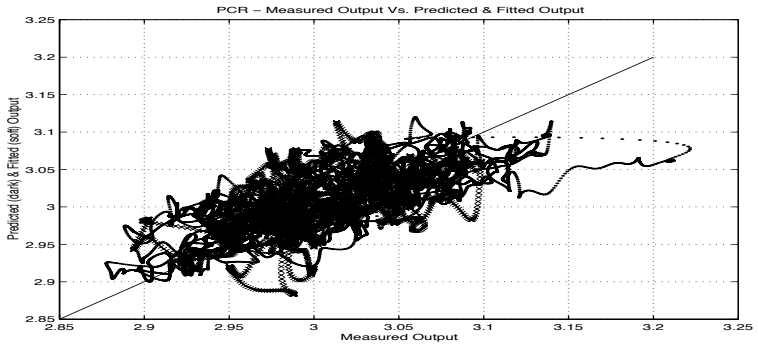


Figure 15: PCR Case

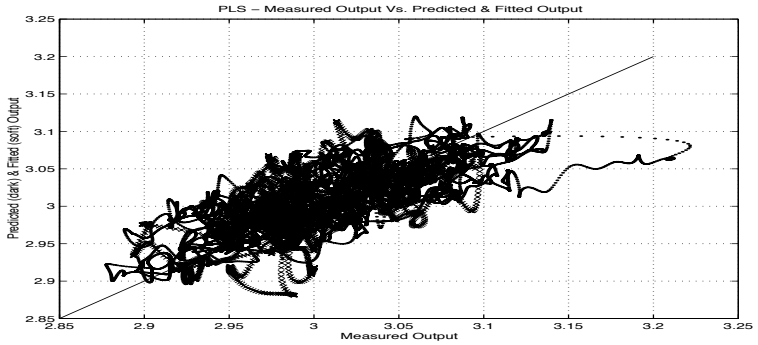


Figure 16: PLS Case

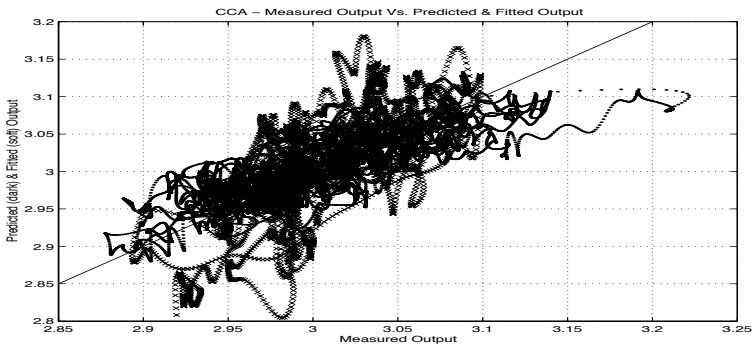


Figure 17: LS-CCA1 Case

B Matlab Algorithms

B.1 PCR-CCA1 Matlab algorithm

```
function [ye,x]=PCR(A,y,NC);

Sxy=A'*y;
[n,p]=size(A);
[V,D,V]=svd(A'*A,0);    % Eigenvalue Decomposition
InvSxx=V(:,1:NC)*inv(D(1:NC,1:NC))*V(:,1:NC)';  % Taking the first NC components
x=InvSxx*S12;      % Computing regression vector
ye=A*x;            % Computing fitted output
```

B.2 PLS1 Matlab algorithm

```
function [ye,x]=PLS1(A,y,NC);
% This function computes the Partial Least Squares (PLS)
% for one output several inputs (PLS1).
% Given y=Ax, where  y(nx1), A(nxp), x(px1),and NC= Number of Components
% find a matrix V such that, U=A*V, y=U*Q', A=U*P' x=V*Q'
% V=[v1 v2...vk], U=[u1 u2...uk], Q=[q1 q2...qk],P=[p1 p2...pk]
% where vk maximizes [cov(uk,y)]^2 subject to vk'*vk=1
%                                             uk'*uj=0 j<>u
%                                             j<k ; 1<k<p
Say=A'*y;
Saa=A'*A;
[n,p]=size(A);
```

```
v=Say;          % Initial value for vk
for i=1:NC
   V(:,i)=v/norm(A*v); % Normalization of vk
   P(:,i)=Saa*V(:,i);  % Computing pk
   Q(:,i)=Say'*V(:,i); % Computing qk
   Saa=Saa-P(:,i)*P(:,i)'; % Subtracting p(k-1) component from A
   Sxy=Sxy-P(:,i)*Q(i); %Subtracting q(k-1) component from y
   v=(eye(p,p)-P*inv(P'*P)*P')*Say; %Computing vk+1
end
x=V*pinv(P'*V)*Q';   % Computing regressor vector
ye=A*x;   % Computing fitted Output
```

B.3 PLS2 Matlab algorithm

See [5]

```
function [Ye,X]=pls(A,Y,NC);

Say=A'*Y;
Saa=A'*A;
C=eye(length(Saa));
for i=1:NC
   [u,s,v]=svd(Say,0);
   V(:,i)=u(:,1)/norm(A*u(:,1));
   P(:,i)=Saa*V(:,i);
   Q(:,i)=Say'*V(:,i);
   Saa=Saa-P(:,i)*P(:,i)';
   h=C*P(:,i)/norm(C*P(:,i));
   C=C-h*h';
   Say=C*Say;
end
X=V*pinv(P'*V)*Q';
Ye=A*X;
```

B.4 CCA2 Matlab algorithm

see [10]

```
function [Ye,X]=CCA(A,Y);

[QA,RA]=qr(A,0);  % QR decomposition of A
[QY,RY]=qr(Y,0);  % QR decomposition of Y
K=QA'*QY;         % Projection of A basis over Y basis
[W,D,Z]=svd(K,0); % Singular Value Decomposition of the projection
U=QA*W;           % Transformed variable computation (Canonical variable)
P=U\Y;            % Least square projection of U in Y
Ye=U*P;           % Estimated Output
C=inv(RA)*W;
X=C*P;            % Regression vector computation
```

C CCA Several Outputs - Several Inputs

C.0.1 Mathematical Description

Given p variables in x $(\alpha_1, \alpha_2, \ldots, \alpha_p)$ and q variables in y $(y_1, y_2, \ldots, y_q)$. Suppose that $\mathbf{v^T}\alpha$ is a linear combination in the x variables, where $\mathbf{v}$ is a $(p \times 1)$ vector of coefficients. Suppose that $\mathbf{w^T y}$ is a linear combination in the y variables, where $\mathbf{w}$ is a $(q \times 1)$ vector of coefficients.

In CCA one solves the following optimization problem:

$$\begin{array}{ll} \max & cor(\mathbf{v^T}\alpha, \mathbf{w^T y}) = \{\mathbf{v_k}, \mathbf{w_k}\} \\ \mathbf{v} \in \mathbb{R}^{\mathbf{p}} & \\ \mathbf{w} \in \mathbf{R}^{\mathbf{l}} & \end{array} \tag{104}$$

subject to

$$\begin{array}{ll} \|\mathbf{v^T}\alpha\| = 1, & \mathbf{v^T S_{AA} V_{k-1}} = \mathbf{0^T}, \\ \|\mathbf{w^T}\alpha\| = 1, & \mathbf{w^T S_{yy} W_{k-1}} = \mathbf{0^T}, \\ & \mathbf{v^T S_{Ay} W_{k-1}} = \mathbf{0^T}, \end{array} \tag{105}$$

where

$$\begin{array}{c} 1 \leq k \leq s = min(p, q), \\ \mathbf{V_{k-1}} = [\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{k-1}}] \text{with dimension } p \times (k-1) \\ \mathbf{W_{k-1}} = [\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_{k-1}}] \text{with dimension } q \times (k-1). \end{array} \tag{106}$$

Note that the correlation reduces to

$$\begin{aligned} cor(\mathbf{v^T}\alpha, \mathbf{w^T y}) &= \frac{cov(\mathbf{v^T}\alpha, w^T y)}{\sqrt{\Big(var(\mathbf{v^T}\alpha)\ var(\mathbf{w^T y})\Big)}} \\ &= \frac{\mathbf{v^T S_{Ay} w}}{\sqrt{\Big((\mathbf{v^T S_{AA} v})\ (\mathbf{w^T S_{yy} w})\Big)}} \\ &= \mathbf{v^T S_{Ay} w}. \end{aligned} \tag{107}$$

under the scaling constraints

$$\|\mathbf{v^T}\alpha\| = var(\mathbf{v^T}\alpha) = \mathbf{v^T S_{AA} v} = 1 \tag{108}$$

$$\|\mathbf{w^T}\alpha\| = var(\mathbf{w^T y}) = \mathbf{w^T S_{yy} w} = 1. \tag{109}$$

Furthermore, the constraint that the between-sets correlations should be zero for different directions in both spaces,

$$cor(\mathbf{v_i}, \mathbf{w_j}) = 0 \qquad \forall\ i \neq j \tag{110}$$

is implied by the constraint

$$\mathbf{v^T S_{Ay} W_{k-1} = 0^T}, \text{ or equivalently, } \mathbf{w^T S_{yA} V_{k-1} = 0^T}. \tag{111}$$

Remark that this constraint is fulfilled automatically when the above scaling constraints are chosen.

C.0.2 Algorithm

Using the technique of Lagrange multipliers, one maximizes

$$\mathbf{v^T S_{Ay} w} \quad \text{subject to} \quad \mathbf{v^T S_{AA} v} = 1 \text{ and } \mathbf{w^T S_{yy} w} = 1 \tag{112}$$

by maximizing

$$\phi(\mathbf{v}, \mathbf{w}) = \mathbf{v^T S_{Ay} w} - \frac{1}{2}\lambda(\mathbf{v^T S_{AA} v} - 1) - \frac{1}{2}\mu(\mathbf{w^T S_{yy} w} - 1). \tag{113}$$

Taking the partial derivatives of ϕ with respect to $\mathbf{v}$ and $\mathbf{w}$ and equating to the null vector, we obtain

$$\frac{\partial \phi}{\partial \mathbf{v}} = \mathbf{S_{Ay} w} - \lambda \mathbf{S_{AA} v} = 0 \tag{114}$$

$$\frac{\partial \phi}{\partial \mathbf{w}} = \mathbf{S_{Ay}^T v} - \mu \mathbf{S_{yy} w} = 0 \tag{115}$$

Premultiplying equation (114) by $\mathbf{v^T}$ and equation (115) by $\mathbf{w^T}$ gives

$$\mathbf{v^T S_{Ay} w} - \lambda \mathbf{v^T S_{AA} v} = 0 \tag{116}$$

$$\mathbf{w^T S_{Ay}^T v} - \mu \mathbf{w^T S_{AA} w} = 0 \tag{117}$$

As $\mathbf{v^T S_{AA} v} = 1$ and $\mathbf{w^T S_{yy} w} = 1$ we derive that

$$\mathbf{v^T S_{Ay} w} = \lambda = \mu \tag{118}$$

From (115) and noting that $\lambda = \mu$ we have

$$\mathbf{S_{Ay}^T v} = \lambda \mathbf{S_{yy} w} \tag{119}$$

Multiplying (114) on the left by $\mathbf{S_{yA} S_{AA}^{-1}}$ gives

$$\mathbf{S_{yA} S_{AA}^{-1} S_{Ay}}\ \mathbf{w} = \lambda \mathbf{S_{Ay}^T}\ v. \tag{120}$$

Substituting (114) into (120) yields the equation in $\mathbf{w}$

$$(\mathbf{S_{yA} S_{AA}^{-1} S_{Ay}} - \lambda^2 \mathbf{S_{yy}^T})\mathbf{w} = 0. \tag{121}$$

A similar manipulation also gives the equation in v

$$(\mathbf{S_{Ay}S_{yy}^{-1}S_{yA}} - \lambda^2\mathbf{S_{AA}^T})\mathbf{v} = 0. \tag{122}$$

The problem is now reformulated as a generalized eigenproblem. If A and Y are of full rank, then $\mathbf{S_{AA}}$ and $\mathbf{S_{yy}}$ are invertible (?) so that (121) and (122) reduce to regular eigenproblems. Hence we can conclude that in CCA one seeks for the pc's of matrices

$$\mathbf{S_{yy}^{-1}S_{yA}S_{AA}^{-1}S_{Ay}} \tag{123}$$

$$\mathbf{S_{AA}^{-1}S_{Ay}S_{yy}^{-1}S_{yA}}. \tag{124}$$

The coefficients $\mathbf{v_1, v_2, ..., v_s}$ and $\mathbf{w_1, w_2, ..., w_s}$ are called the canonical weights, the eigenvalues $\sqrt{\lambda_k^2}$ are called the canonical coefficients, and the linear combinations u and z are called canonical variates.

Remark the projection matrices in the matrices above

$$\mathbf{(A^TA)^{-1}\ A^TY(Y^TY)^{-1}Y^TA} \tag{125}$$

$$\mathbf{(Y^TY)^{-1}\ Y^TA(A^TA)^{-1}A^TY}. \tag{126}$$

See appendix (B.4) for a matlab code.

Remarks

CCA2 is only of interest when both sets have more or less the same dimensionality or $p \geq 2$ and $q \geq 2$, since we need the first few linear combinations that account for most of the correlation.

D Bibliography

- A. J. Miller, Subset Selection in Regression, Chapman & Hall, 1990.
- E. L. Russell, L. H. Chiang, R.D. Braatz, Data-driventechniques for fault detection and diagnosisin chemical processes, Springer Verlag London, 2000.
- J.D. Jobson, Applied Multivariate Data Analysis, Vol 1&2, Springer, 1992.
- I.T. Jolliffe, Principal Component Analysis, Springer Verlag, 1986.
- Helland, I. (1988). On the structure of partial least squares regression. Communications in Statistics- Simulation and Computations, 17, 581-607.
- Wold, H. (1966). Estimation of principal components and related models by iterative least squares, in Multivariate Analysis,P.R. Krishnaiah, ed. Academic Press, New York, 391-420.
- Lorber, A., Wangen, L, and Kowalski, B. (1987). A theoretical foundation for the PLS algorithm. Journal of Chemometrics, 1, 19-31. Frank, I and Friedman, J. (1993). A statistical view of some chemometrics regression tools. Technomet- rics, 35, 109-148.
- Stone, M., and Brooks, R. (1990). Continuum regression: cross-validated sequentially constructed pre- diction embracing ordinary least squares, partial least squares and principal components regression. Journal of the Royal Statistical Society, Series B, 52, 237-269.
- W.F. Massy (1965), Principal Component Regression in Exploratory Statistical Research, Journal of the American Statistical Association, 60,pp. 234-246.
- Martens, H., Wold, S., and Martens, M. (1983), a laymans guide to multivariate data analysis. In Food Research and Data Analysis, H. Martens and H. Russworm, Jr. (eds) 473-492. London : Applied Science Publications (about CCA).
- Hoerl, A.E.and Kennard, R.W. (1970), Ridge regression biased estimation for nonorthogonal problems, Technometrics, 8, 27-51.

References

[1] D.J.C.Mackay, Probable networks and plausible predictions, a review of practical Bayesian methods for supervised neural networks

[2] D.J.C.MacKay, 1992, Bayesian interpolation, Neural Computation 4(3) 415-447

[3] D.E.Rumelhart, G.E.Hinton, R.J.Williams, 1986, "learning represenations by backpropagation errors", Nature vol. 323, pp. 533-536

[4] M. Borga, "Learning Multidimensional Signal Processing", phd thesis, Department of Electrical Engineering, Linkoping University,1998.

[5] S. De Jong, 1993, Simpls: An alternative approach to partial least squares regression. Chemometrics and Intelligent lab. systems, 18, 251-263.

[6] B.R. Kowalski and M.B. Seasholtz, 1991, Recent developments in multivariate calibration. Journal of Chemometrics, vol. 5, 129-145.

[7] C.M.Bishop (1995), Neural NEtworks for Pattern recognition, Oxford University Press.

[8] Rayens W. S., the art of maximizing covariance, unpublished notes, 6/2000, university of kentucky, dept. of statistics, tecnical reports. (http://www.ms.uky.edu/ statinfo/techreports/techreports.html)

[9] Hinkle and Rayens, Partial Least Squares for Compositional Data - Problems and Alternatives , 1995; Chemometrics and Intelligent Laboratory Systems. (http://www.ms.uky.edu/ jhinkle/JEHdoc/JEHdoc.html) (http://www.ms.uky.edu/ rayens/research/archives/)

[10] R. Gittins, Canonical analysis, Springer-Verlag, 1985.

[11] Van Huffel S., Vandewalle J.(1988), The total least squares technique: computation, properties and applications, in SVD and Signal Processing: Algorithms, Applications and Architectures, (Deprettere E.,ed.), Elsevier, 1998 pp. 189-207.