CHALMERS

Implementation and Verification of a Robust PLS Regression Algorithm.

Master's Thesis in Engineering Mathematics and Computational Science

SIMON HALL

Department of Mathematical Sciences
Mathematical Statistics
Chalmers University of Technology
Gothenburg, Sweden 2014

Master's Thesis 2014:1

Abstract

Partial least squares regression (PLSR) based methods play an important role when dealing with collinear high dimensional regressors. In this thesis we introduce and implement RSIMPLS, a robust alternative to traditional PLSR methods such as non-linear iterative partial least squares (NIPALS) and SIMPLS. The RSIMPLS method has been used to analyse fluid properties using discretized acoustical spectrum data.

We find evidence that RSIMPLS performs better than SIMPLS, principal components regression (PCR) and canonical correlation analysis (CCA) for high-dimensional regressors (observation DOF p larger than number of observations n). However CCA performs slightly better for low dimensional regressors ($n > p$). We also find that RSIMPLS converges to the same limit as CCA for infinitely large, normally distributed sample sets.

Acknowledgements

I would like to thank my colleagues and friends at Acosense AB for their support, friendship and for all the things we have achieved together. It has truly been a pleasure and a very worthwhile experience. Furthermore I would like to direct a special thanks to Dr. Johan Karlsson at the Fraunhofer-Chalmers Centre for his help and expertise throughout the project and to Prof. Olle Nerman for his insightful comments and helpful advice.

Simon Hall, Göteborg 30/03/14

Abbreviations and Matrices

$\mathbf{B}_0$	Regression intercept.
$\mathbf{B}$	Regression coefficients. Used to predict $\mathbf{Y}$.
CCA	Canonical Correlation Analysis
EVD	Eigen Value Decomposition
MLR	Multiple Linear Regression
OD	Orthogonal Distance
PC	Principal Component
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
PLSR	Partial Least Squares Regression
RA	Regression Analysis
RSIMPLS	Robust implementation of PLS regression.
ROBPCA	Robust PCA implementation.
$\mathbf{P}$	Loadings matrix.
SIMPLS	Implementation of PLS regression.
SVD	Singular Value Decomposition. Matrix factorization. $\mathbf{X} = \mathbf{USV}^T$
$\mathbf{T}$	Score matrix. Variables transformed to the PCA subspace.
TD	Score Distance
$\mathbf{W}$	Weight matrix. Transforms $\mathbf{X}$ to $\mathbf{T}$.

Contents

1 Introduction

This thesis will cover the implementation and verification of RSIMPLS, a robust method for linear regression. RSIMPLS is a variant of partial least squares regression (PLSR). PLSR-based techniques are especially useful when dealing with highly-collinear multidimensional regression variables, cases where ordinary multiple linear regression (MLR) often breaks down.

1.1 Background

This thesis has been done in collaboration with Acosense AB. Acosense's main product is the Acospector Acoustic Chemometer, a clamp-on instrument measuring complex fluids in the process industry. The Acospector can analyse fluid properties previously thought difficult or impossible to measure.

The instrument emits acoustic stimulus and measures the acoustic response. A statistical model is then used to correlate the acoustic data with the desired fluid property. The technology is fluid independent and can be used to measure a wide range of properties such as viscosity, particle concentrations, fibre content and more.

The Acospector consists of two main parts, an emitter that generates an acoustic signal and a high performance accelerometer that records the transmitted signal essentially like a microphone. The emitter and accelerometer are typically placed on opposite sides of the pipe or tank containing the fluid. Figure 1.1 shows a simplified schematic view of the instrument setup.

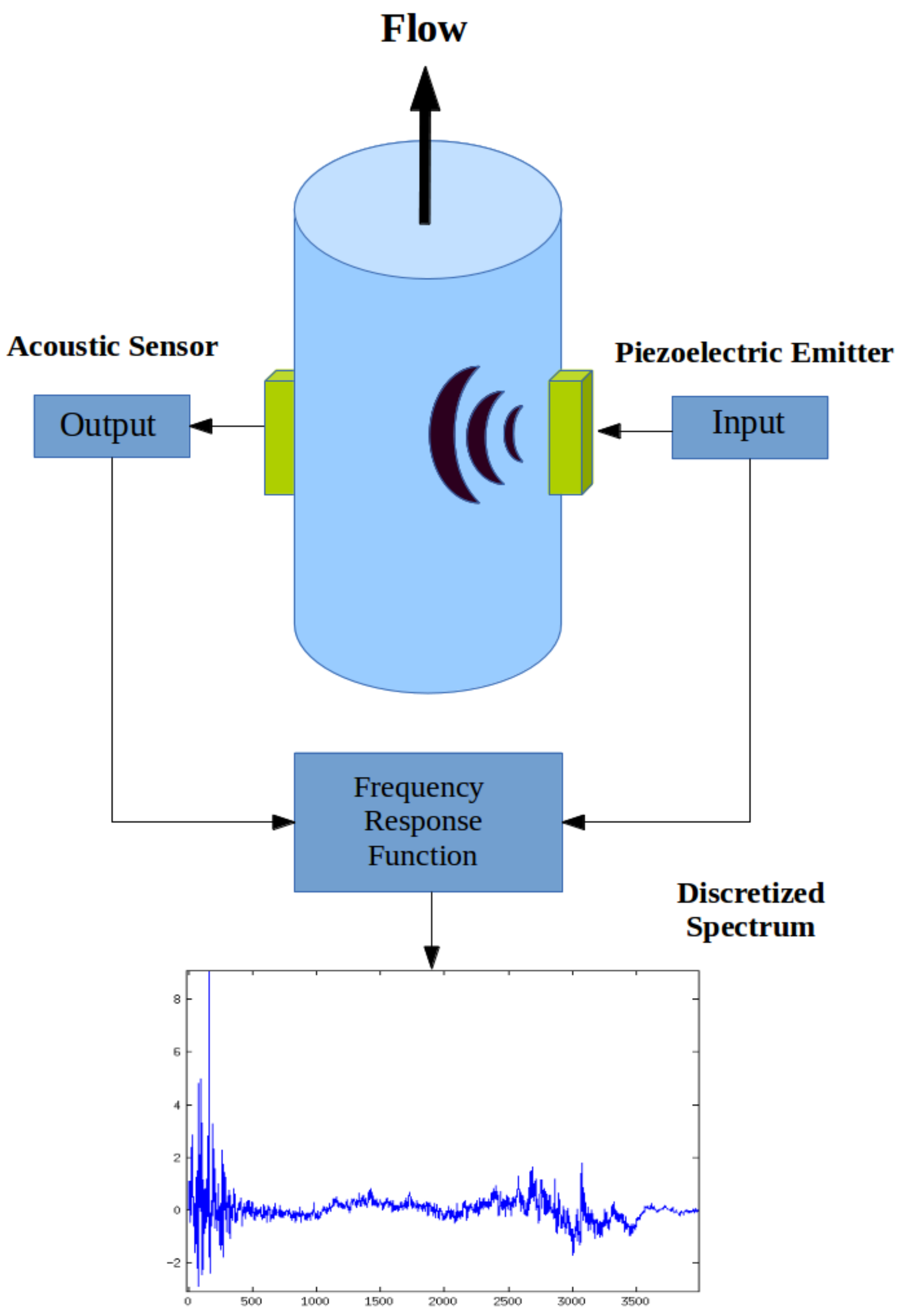


Figure 1.1: An overview of the Acospector setup.

The captured acoustic data is transformed to the frequency domain which results in a discretized spectrum. Figure 1.2 shows such a collection of spectra. It is these spectra that will be the predictor variables when estimating the fluid properties using a statistical model.

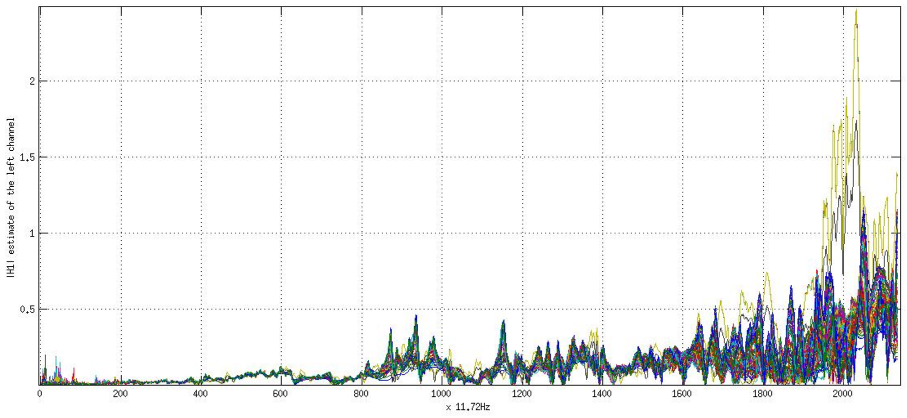


Figure 1.2: A collection of discretized acoustic spectra from an experiment carried out at Acosense. We can see that the spectra have the same general shape with a few possible outliers.

1.2 Problem Definition

The aim of the thesis is to evaluate and implement RSIMPLS to estimate fluid properties from discretized acoustical spectra. The implementation should be of sufficiently high quality for industrial use. Furthermore an evaluation of RSIMPLS as a method should be done to compare it with other similar solutions and statistical methods.

1.3 Notation

In this thesis matrices will be denoted by capitalized bold letters e.g. $\mathbf{T}$. Subscripts are added to denote the size of the matrix, e.g. $\mathbf{T}_{m,n}$ where m is the number of rows and n the number of columns in the matrix. A transposed matrix is denoted by a superscript, T e.g. $\mathbf{A}^T$. Matrix elements will be denoted by a non-bold capital letter with a double subscript e.g. $T_{i,j}$.

Vectors will be denoted by a bold non-capital letter e.g. $\mathbf{v}$ or with an appended subscript to denote the vector size e.g. $\mathbf{v}_n$. Individual vector elements will be denoted by a non-bold, non-capital letter with a single subscript e.g. v_i. Individual rows or columns of matrices will sometimes be denoted $\mathbf{v}^{(i)}$.

Scalars will be denoted by non-bold letters such as p.

1.4 Technical Introduction

PLSR can be seen as a combination of two techniques, partial least squares (PLS) and regression analysis (RA). The reader is assumed to be familiar with the basics of regression analysis, otherwise ”Mathematical Statistics and Data Analysis” by John A. Rice [14] is recommended as an introduction to the topic. The reader is however not expected to be familiar with PLS.

PLS is a method to reduce the dimensionality of a dataset while maximizing the covariance with another dataset. This is particularly useful in the case of regression analysis. Let $\mathbf{X}_{n,p}$ be our set of predictors and $\mathbf{Y}_{n,q}$ our set of responses. In regression analysis we want to find a mapping f describing the observations $\{\mathbf{Y}_{j,q}\}_{j=1}^{n}$ as a function of the predictors $\{\mathbf{X}_{j,p}\}_{j=1}^{n}$. However if p, the number of columns of $\mathbf{X}$, is too large we might have trouble finding a good mapping f. In the case of MLR p is required to be less than or equal to n in order for the matrix $\mathbf{X}^T\mathbf{X}$ to be invertible and for ordinary least squares regression to be possible.

PLS provides a way to approximate $\mathbf{X}_{n,p}$ as $\mathbf{T}_{n,k}$ for any $1 \leq k \leq p$ while maximizing the covariance between $\mathbf{T}_{n,k}$ and $\mathbf{Y}_{n,q}$. The point of this is to be able to later use regression analysis to find a relationship between $\mathbf{T}_{n,k}$ and $\mathbf{Y}_{n,q}$. The theory behind PLS and PLSR will be covered briefly in the following chapter.

Herman Wold (1908 - 1992), a Swedish statistician, presented the NIPALS algorithm for PLSR in his article ”Causal flows with latent variables: Partings of the ways in the light of NIPALS modelling” in 1974 [8] and had presented the general ideas behind PLSR a few years earlier. Herman Wold continued to contribute to the field of statistics until the early 1980s, sometimes in collaboration with his son, Svante Wold (born 1941). For a general introduction to the subject the book ”Multi- and Megavariate Data Analysis Part I Basic Principles and Applications” by L. Eriksson, E. Johansson, N. Kettaneh-Wold, J. Tryggm C. Wikström and S. Wold [12] is recommended.

RSIMPLS is an extension of traditional PLSR that includes robustness in the form of resistance to outliers, both in the PLS-step and in the RA-step [1]. In theory a single outlier could influence a model (i.e. the function describing the relationship between $\mathbf{X}$ and $\mathbf{Y}$) to an arbitrarily large extent in the standard versions of PLSR. Figure 1.3 illustrates how outliers can influence the PLS step with and without robustness implemented. We will explain the figure later in the theory chapter.

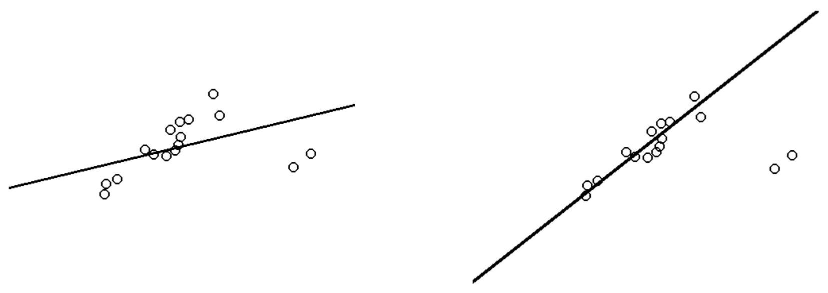


Figure 1.3: An illustration of the importance of robust methods in PLSR-like methods. In short the line in the graph represent the axis on which the points are projected on to in the dimension reduction step. The major difference between figure a) and b) is that the Robust PLS does not include the two outliers two the right in the model.

RSIMPLS was introduced in 2003 by Mia Hubert and Karlien Vanden Branden in their article ”Robust methods for partial least squares regression” [1].

2

Theory

This chapter contains the theoretical material that is base for this thesis. Note that the ROBPCA and RSIMPLS algorithms are extensive and will not be covered completely in this report.

Note that all methods presented in this chapter are applicable to many types of variables as long as they have certain properties such as well defined addition, subtraction, products, etc. However they should be regarded as ordinary vectors and matrices in this thesis.

Since the reader is not expected to have any prior knowledge of either PLS or PCA the theory chapter has been extended to include a brief introduction to them both.

2.1 PCA: Principal Component Analysis

The idea behind PCA is to reduce the dimension of a dataset while retaining as much of its information as possible. This is done by finding a new orthogonal basis for the data set. The base vectors of the new basis are ordered in descending order by how much variance of the original data they explain. By using a reduced number of base vectors a dimension reduction can be obtained. Since the higher order base vectors describes less structure of the original data set they can be discarded with a minimized loss of explained variance [13].

PCA assumes that $\mathbf{X}$ can be decomposed on the form [5]:

$$\mathbf{X} = \bar{\mathbf{X}} + \mathbf{T}\mathbf{W}^T + \mathbf{E} \tag{2.1}$$

Where $\bar{\mathbf{X}}$ is the center of $\mathbf{X}$. $\mathbf{T}$ and $\mathbf{W}$ are called latent variables, that is variables that describes $\mathbf{X}$ but are not directly observed. $\mathbf{T}$ is the score matrix, and can be seen as the original data transformed to a new, possibly smaller, basis. $\mathbf{W}$ is called the loadings or weight matrix and defines the basis of $\mathbf{T}$ in the coordinate system of $\mathbf{X}$. $\mathbf{W}$ will be described in greater detail below. The matrix $\mathbf{E}$ contains error terms.

The process of creating the PCA basis $\mathbf{W}$ could be seen as iteratively choosing base vectors such that they maximize the variance of the transformed variables (e.g rows or columns in a matrix) and are uncorrelated to all previously chosen base vectors. The fact that PCA maximizes variance [13] is important and will later turn out to be the major difference between PCA and PLS.

Let $\mathbf{X}_{n,p}$ be the matrix with the original data and let $\{\mathbf{w}^{(i)}\}_{i=1}^{k} = \{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, ..., \mathbf{w}^{(k)}\}$ be the (currently unknown) base vectors spanning the PCA-subspace. Note that k can be any integer $1 \leq k \leq p$. Assume $\mathbf{X}$ is centered, i.e that all of its columns are of zero-mean (otherwise center it by mean subtraction). Further assume that all base vectors are normalized, i.e $||\mathbf{w}^{(i)}|| = 1$.

Start by choosing $\mathbf{w}^{(1)}$ such that $\mathbf{w}^{(1)^T}\mathbf{X}^T\mathbf{X}\mathbf{w}^{(1)}$ is maximized under the condition of $\mathbf{w}^{(1)}$ being normalized. The next step is to deflate the matrix $\mathbf{X}$ by itself projected on all the previously found base vectors. Mathematically this can be expressed as:

$$\hat{\mathbf{X}}^{(i)} = \mathbf{X} - \sum_{j=1}^{i-1} \mathbf{X}\mathbf{w}^{(j)}\mathbf{w}^{(j)^T} \tag{2.2}$$

To find the i:th base vector redo the calculation above but with $\hat{\mathbf{X}}_i$ instead. The whole scheme can be summarize as iteratively solving the following equation for increasing i starting from 1.

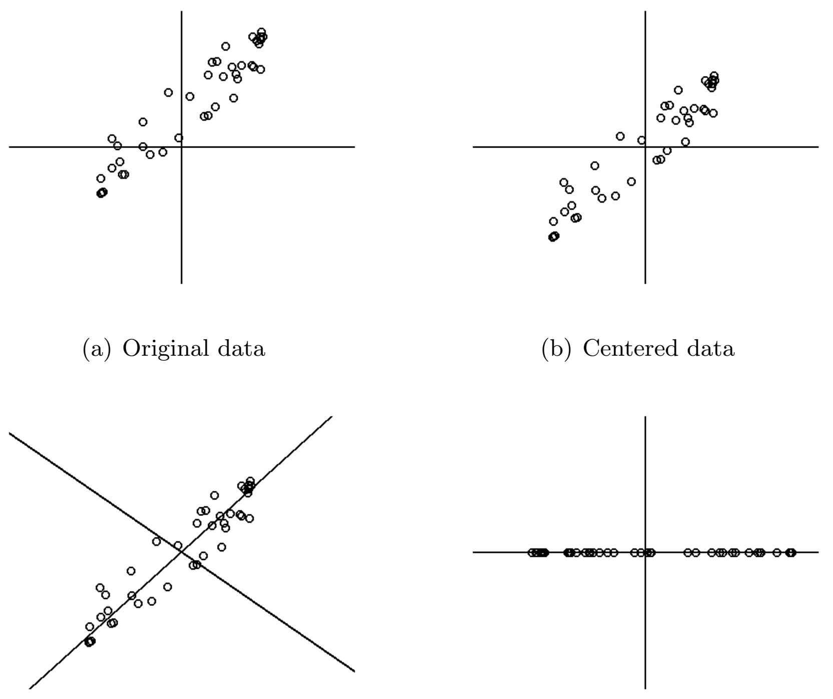
$$\mathbf{w}^{(i)} = \underset{\mathbf{w}}{\operatorname{argmax}} \left(\mathbf{w}^T \hat{\mathbf{X}}^{(i)^T} \hat{\mathbf{X}}^{(i)} \mathbf{w} \right) \quad s.t \quad |\mathbf{w}| = 1 \tag{2.3}$$

The process can be stopped at any non-negative integer k of base vectors less than or equal to n. k is the number of components in the PCA-subspace. The fewer components one keep the more data will be discarded in the transformed data set. It is not uncommon to only keep 1-15 components out of several thousand if the original data is highly collinear.

The k components can be combined as columns to create the $\mathbf{W}$ matrix, i.e $\mathbf{W} = \{\mathbf{w}^{(1)}, ..., \mathbf{w}^{(k)}\}$. We can, as mentioned in the beginning of this section, multiply $\mathbf{X}$ by $\mathbf{W}$ to transform $\mathbf{X}$ to the PCA subspace. We have that:

$$\mathbf{T}_{n,k} = \left(\mathbf{X}_{n,p} - \bar{\mathbf{X}} \right) \mathbf{W}_{p,k} \tag{2.4}$$

It is important to consider the scales of the input variables (e.g. columns of $\mathbf{X}$) when using techniques such as PCA or PLS. If the variables are of different scales relative to each other the variables of higher scale will be dominant in the final PCA [16]. This might be a desired behavior but care should be taken, especially when comparing data from different sources where no apparent relative scaling is available. Sometimes a standardization of input data prior to the PCA is recommended.



(a) Original data (b) Centered data

(c) Identification of principal components (d) Transformation of original data

Figure 2.1: Graphical representation of a PCA transformation of a 2-dimensional data set onto a single PC. The transformed data shown in plot (d) indicates that the variance along the PC is indeed maximized.

Another way of performing PCA is by singular value decomposition (SVD). SVD factorizes a matrix $\mathbf{X}$ into three parts $\mathbf{U}, \mathbf{S}, \mathbf{V}$. By first centering $\mathbf{X}$ we can write the SVD factorization of $\mathbf{X}$ as [5]:

$$\mathbf{X}_{n,p} = \bar{\mathbf{X}} + \mathbf{U}_{n,n}\mathbf{S}_{n,p}\mathbf{V}^T_{p,n} \tag{2.5}$$

Notice the similarities with equation (2.1). The singular-value matrix $\mathbf{S}$ only has non-negative diagonal elements sorted by size in descending order. $\mathbf{U}$ contains n orthonormal base vectors as columns. We can make an equivalent transformation as in (2.4) by truncating $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$. For a given k we have that $\mathbf{T}^* = \mathbf{U}_{n,n}\mathbf{S}_{n,k}$ and $\mathbf{W}^* = \mathbf{V}$ which turns equation (2.5) into the same form as (2.1). If not all singular values are unique neither PCA or SVD are uniquely defined. This is because all base vectors with a common singular value can be rotated freely in the spanned hyperplane. In most cases however $\mathbf{T}^* = \mathbf{T}$ and $\mathbf{W}^* = \mathbf{W}$ which means that we in many practical applications can use truncated SVD as a replacement for PCA [10].

2.2 PLS: Partial Least Squares

PLS is a dimension reduction technique with some similarities to PCA. While PCA reduces the dimensionality of a single data set by iteratively projecting the data onto components of maximum variance, PLS reduces the dimensionality of a data set by projecting the data onto components of maximum covariance with a second data set. PLS assumes the following fundamental relationship between the two sets.

$$\begin{cases} \mathbf{X} = \bar{\mathbf{X}} + \mathbf{T}\mathbf{P}^T + \mathbf{E}_X \\ \mathbf{Y} = \bar{\mathbf{Y}} + \mathbf{U}\mathbf{Q}^T + \mathbf{E}_Y \end{cases} \tag{2.6}$$

Where $\mathbf{T}$ is the $\mathbf{X}$ scores, $\mathbf{P}$ the $\mathbf{X}$ loadings (similar to $\mathbf{W}$ in PCA), $\mathbf{U}$ the $\mathbf{Y}$ scores and $\mathbf{Q}$ the $\mathbf{Y}$ loadings. Notice the similarities with eq (2.1). In PLS the covariance between score vectors is maximized in each iteration (similar to how the variance was maximized for each iteration in PCA). This covariance criterion can be expressed by the following equation [3].

$$\begin{aligned} \mathbf{w}^{(i)}, \mathbf{q}^{(i)} = \underset{\mathbf{w},\mathbf{q}}{\operatorname{argmax}}\ \ \mathbf{t}^t\mathbf{u} = \underset{\mathbf{w},\mathbf{q}}{\operatorname{argmax}}\ \ \mathbf{q}^T\left(\mathbf{Y}_0^T\mathbf{X}_0\right)\mathbf{w} \\ s.t.\quad |\mathbf{w}| = 1,\ \ |\mathbf{q}| = 1,\ \ \mathbf{t}^{(i)} \perp \mathbf{t}^{(j)}\ \ \forall\ j < i \end{aligned} \tag{2.7}$$

Let $\mathbf{X} = \mathbf{X}_{n,p}$ and $\mathbf{Y} = \mathbf{Y}_{n,q}$ be two data sets. Furthermore let $PCA(\mathbf{X},k)$ be the function that maps a data set $\mathbf{X}$ onto the PCA space with k components. Also let $PLS(\mathbf{X},\mathbf{Y},k)$ be the function that maps the data set $\mathbf{X}$ onto the PLS space of k components with respect to $\mathbf{Y}$.

For a fixed k_0, $\mathbf{T}^{PCA} = PCA(\mathbf{X},k_0)$ is the optimal projection of $\mathbf{X}$ with respect of explained variance. Similarly $\mathbf{T}^{PLS} = PLS(\mathbf{X},\mathbf{Y},k_0)$ is the optimal projection of $\mathbf{X}$ with respect of explained covariance between $\mathbf{X}$ and $\mathbf{Y}$. One can say that the covariance information lost in the transformation is minimized for that particular k_0.

There are two major ways of implementing PLS: NIPALS and SIMPLS, both yielding equivalent results for univariate $\mathbf{Y}$ but might differ if $\mathbf{Y}$ is multivariate [3]. This difference is typically small and is due to the fact that the data matrix deflation differs slightly [3]. However in both NIPALS and SIMPLS the covariance criterion, i.e. $cov(t^{(i)},u^{(i)})$ is maximized under the conditions of normalized weights and standardized scores [3]. In this thesis we will only cover SIMPLS since it is the base for RSIMPLS. A description of NIPALS can be found in [8] and [4]. A description of SIMPLS is found in the following chapter or in Sijmen de Jong's article ”SIMPLS: an alternative approach to partial least squares regression” [3].

2.3 PLSR: Partial Least Squares Regression

PLS can be, and commonly is, combined with a regression step. The resulting method is called partial least squares regression (PLSR). PLSR yields two variables $\mathbf{B}$ and $\mathbf{B}_0$ such that $\hat{\mathbf{Y}} = \mathbf{B}_0 + \mathbf{XB}$ is a good approximation of $\mathbf{Y}$. The PLSR algorithm can be described in terms of a PLS step follow by regression of $\mathbf{Y}$ w.r.t $\mathbf{X}$.

```
Input: Predictors $\mathbf{X}_{n,p}$, Responses $\mathbf{Y}_{n,q}$, integer $k_0$
Result: Regression parameters $\mathbf{B}$, Intercept $\mathbf{B}_0$
Algorithm:
$\mathbf{T}_{n,k_0} = PLS(\mathbf{X},\mathbf{Y},k_0)$              // Perform PLS step
$\mathbf{B}_0,\mathbf{B} = MLR(\mathbf{T},\mathbf{Y})$            // Regression step with MLR
```

Algorithm 1: General idea of PLSR

Both NIPALS and SIMPLS provides an even simpler way of performing PLSR. Instead of using MLR to perform the regression step, $\mathbf{B}$ and $\mathbf{B}_0$ can be found directly through the variables generated in the earlier stages of the algorithm. The SIMPLS algorithm as formulated by Jong [3] is as follows:

```
**Input**:
Predictors $\mathbf{X}_{n,p}$
Responses $\mathbf{Y}_{n,q}$
Non-negative integer $k_0 \leq n$
**Result**:
Regression parameters $\mathbf{B}$
Regression Intercept $\mathbf{B}_0$
Weight Matrix $\mathbf{W}$
Score Matrix $\mathbf{T}$
**Algorithm:**
$\mathbf{Y}_0 = \mathbf{Y} - \bar{\mathbf{Y}}$                                   // Center $\mathbf{Y}$
$\mathbf{S} = \mathbf{X}^T\mathbf{Y}_0$                          // $\mathbf{S}$ similar to covariance matrix
for $k \leftarrow 1$ to $k_0$ do
    $\mathbf{q}$ = dominant eigenvector of $\mathbf{S}^T\mathbf{S}$
    $\mathbf{w} = \mathbf{S}\mathbf{q}$
    $\mathbf{t} = \mathbf{X}\mathbf{w}$
    $\mathbf{t} = \mathbf{t} - \bar{\mathbf{t}}$
    $\mathbf{w} = \mathbf{w}/||\mathbf{t}||$
    $\mathbf{t} = \mathbf{t}/||\mathbf{t}||$
    $\mathbf{p} = \mathbf{X}^T\mathbf{t}$
    $\mathbf{q} = \mathbf{Y}_0^T\mathbf{t}$
    $\mathbf{u} = \mathbf{Y}_0\mathbf{q}$
    $\mathbf{v} = \mathbf{p}$
    if $k > 1$ then
        $\mathbf{v} = \mathbf{v} - \mathbf{V}(\mathbf{V}^T\mathbf{p})$                    // Deflate $\mathbf{v}$
        $\mathbf{u} = \mathbf{u} - \mathbf{T}(\mathbf{T}^T\mathbf{u})$                    // Deflate $\mathbf{u}$
    end
    $\mathbf{v} = \mathbf{v}/||\mathbf{v}||$
    $\mathbf{S} = \mathbf{S} - \mathbf{v}(\mathbf{v}^T\mathbf{S})$                        // Deflate $\mathbf{S}$
    $\mathbf{W} = [\mathbf{W},\mathbf{w}]$                           // Append $\mathbf{w}$ to $\mathbf{W}$
    Append $\mathbf{t}$, $\mathbf{p}$, $\mathbf{q}$, $\mathbf{u}$, $\mathbf{v}$ to $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{U}$, $\mathbf{V}$ respectively
end
$\mathbf{B} = \mathbf{W}\mathbf{Q}^T$                                  // Regression step
$\mathbf{B}_0 = \bar{\mathbf{Y}}$                                      // Intercept
```

Algorithm 2: SIMPLS algorithm as formulated by Jong [3]

Notice the difference between **W** in PLS and **W** in PCA. For both PLS and PCA **W** defines the transformation of **X** to **T** however the score matrix **T** have different properties. In PCA **T** explains the variance of **X** but in PLS **T** explains covariance between **X** and **Y**. Thus **W** in PCA would be closer to the matrix **P** in PLS. This can be confusing at first and sometimes different notation is used to avoid confusion, e.g. **W** could sometimes be called **P** in PCA and **R** in PLS.

2.4 Uncertainty Measures

Both PLS and PCA will transform an observation $\mathbf{x}^{(i)} \in \mathbf{X}$ to the score space resulting in the transformed variable $\mathbf{t}^{(i)} \in \mathbf{T}$. The scores are commonly chosen s.t they are centred around origo. In other words:

$$\bar{\mathbf{t}} \approx \mathbf{0} \tag{2.8}$$

The score distance (TD), can be seen as the leverage of an observation. The score distance gives us information about how influential that observation was during the building of the model and in a sense also how similar that observation is to the other observations in the set. The score distance for a score matrix can be computed by [2]:

$$TD_i = \sqrt{\sum_{i=1}^{k} \frac{T_{i,j}^2}{l_j}} \tag{2.9}$$

Where $\mathbf{l}$ is the vector of eigenvalues of the scatter matrix $\mathbf{S}$ sorted in descending order [2]. The reason for dividing with the eigenvalue is to make the higher order principal components more influential to the total score distance.

Another interesting property that can be found for each observation is the orthogonal distance (OD). The orthogonal distance quantifies how well an observation fits to a certain model. It can also be seen as the information lost in the dimension reduction from the PCA or PLS or the distance between the original data point and the projection onto the model in the $\mathbf{X}$ space. For an observation $\mathbf{x}^{(i)}$ with scores $\mathbf{t}^{(i)}$ the orthogonal distance is calculated by [2]:

$$OD_i = ||\mathbf{x}^{(i)} - \bar{\mathbf{X}} - \mathbf{t}^{(i)}\mathbf{P}^T||_{euclidean} \tag{2.10}$$

For the ROBPCA algorithm $\bar{\mathbf{X}}$ is replaced by the robust center $\boldsymbol{\mu}_x$ [2]. Figure 2.2 visualizes the meaning of the score and orthogonal distances in a dimension reduction from 2D to 1D.

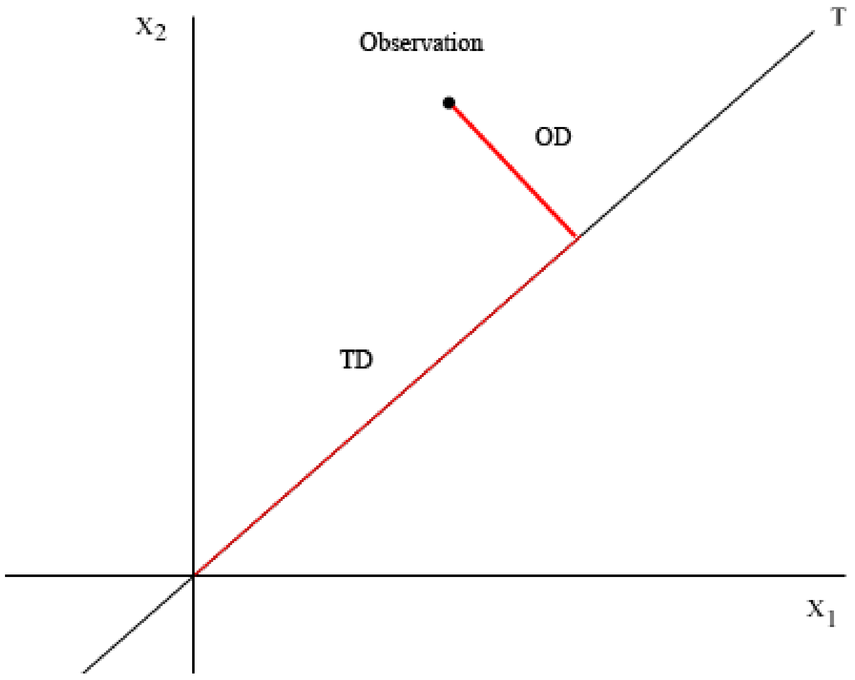
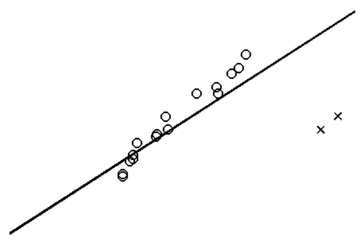


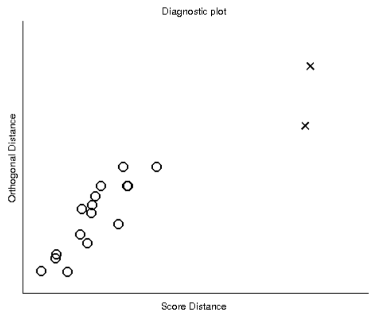
Figure 2.2: A visualization of the uncertainty measures score and orthogonal distance.

The score and orthogonal distance gives valuable information about the outlyingness of observations. Observations with a large TD but small OD are called ”good leverage points” [2]. Points with a small TD but large OD are called ”orthogonal outliers” and points with both high TD and OD are called ”bad leverage points” [2]. The identified bad leverage points can be due to several reasons: observation outlyingness, a poor model or if a linear model is not suitable for the data. The bad leverage points due to observation outlyingness are particularly destructive to the quality of the model and a pre-step removing those is the key feature of robust PCA and PLS methods.

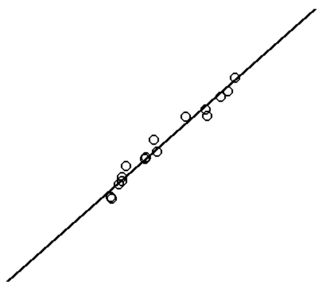
Finally figure 2.3 illustrates a model created with and without 2 outliers (marked by crosses). As we can see the outliers are identifiable in the diagnostic plot (a plot with observation TD and OD on separate axis). The removal of the outliers greatly increases the model performance.



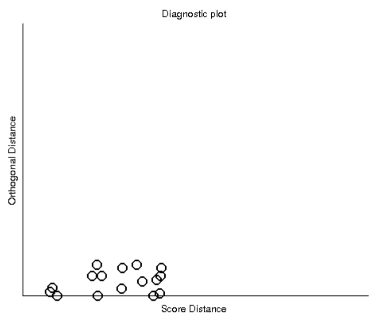
(a) Model created from data including outliers (marked by x).



(b) Diagnostic plot of model including the outliers. Notice that the two outliers separates from the other observations.



(c) Model created from data without outliers.



(d) Diagnostic plot in same scale as above.

Figure 2.3: The figure illustrates how outliers effect model performance and how they could be identified in a diagnostic plot.

2.5 ROBPCA: Robust PCA

Since the classical PCA algorithm is based on finding components of maximum variance it is sensitive to outliers in the data, something that was illustrated in figure 1.3. In fact a single observation could have an arbitrarily large impact on the model. The ROBPCA method aims to add robustness to the classical PCA algorithm while keeping most of its properties intact.

The ROBPCA algorithm is extensive and only the basics will be covered in this thesis. For the complete version I refer to "ROBPCA: A New Approach to Robust Principal Component Analysis" by Mia Hubert, Peter J. Rousseeuw and Karlien Vanden Branden [2].

The first step of the ROBPCA algorithm is to transform the data $\mathbf{X}_{n,p}$ to a subspace of dimension $r_0 = rank(\mathbf{X}_{n,p} - \bar{\mathbf{X}})$ [2]. This can be done by a matrix factoring technique such as SVD or EVD. Note that $\mathbf{X}_{n,p}$ is assumed to be any generic data at this point and should not be equated with the predictors in the RA problem. In fact the RSIMPLS methods calls ROBPCA with the concatenated matrix $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]$ as will be shown later. In the case of SVD we would have the following operations [2]:

$$\mathbf{X}_{n,p} - \bar{\mathbf{X}} = \mathbf{U}_{n,r_0} S_{r_0,r_0} \mathbf{V}^T_{r_0,k} \tag{2.11}$$

$$\mathbf{Z}_{n,r_0} = \mathbf{U}_{n,r_0} S_{r_0,r_0} \tag{2.12}$$

We will use the transformed data **Z** in the rest of the algorithm. Note that **Z** contains the same information as **X** but is expressed in an orthogonal basis.

The ROBPCA algorithm requires the user to specify a parameter α corresponding to the fraction of the total number of observations that can be considered non-outliers. For example if we have 100 observations and expect roughly 15 of them to be outliers an alpha less than or equal to 0.85 is required for optimal performance. The user could also equivalently specify h, the total number of non-outliers, given that $h = n\alpha$.

The second step of the ROBPCA algorithm is to find the h least outlying observations out of the n total [2]. To do this we first choose 250 pairs of points from **Z** at random and let **B** be the set of directions determined by each pair [2]. Each observation $\mathbf{z^{(i)}}$ is then assigned an outlyingness measure according to the formula [2]:

$$outl(\mathbf{z}^{(i)}) = \max_{\mathbf{v}\in\mathbf{B}} \left(\frac{|\mathbf{z}^{(i)^T}\mathbf{v} - t_{MCD}(\mathbf{z}^{(i)^T}\mathbf{v})|}{s_{MCD}(\mathbf{z}^{(i)^T}\mathbf{v})} \right) \tag{2.13}$$

Where $t_{MCD}(\mathbf{z}_i^T\mathbf{v})$ and $s_{MCD}(\mathbf{z}_i^T\mathbf{v})$ are the univariate MCD location and scale estimators [2] [9]. If we order $\mathbf{z}_i^T\mathbf{v}$ s.t $Var(\mathbf{z}_j^T\mathbf{v}) \geq Var(\mathbf{z}_i^T\mathbf{v}) \ \forall \ j > i$, that is in increasing order w.r.t. variance, we have that:

$$t_{MCD} = Mean(\{\mathbf{z}^{(i)^T}\mathbf{v}\}_{i=1}^{h}) \tag{2.14}$$

$$s_{MCD} = SD(\{\mathbf{z}^{(i)^T}\mathbf{v}\}_{i=1}^{h}) \tag{2.15}$$

Note that the sorting of $\mathbf{z}^{(i)^T}\mathbf{v}$ will have to be reiterated for every choice of v. For more details on this step please refer to [2] and [9].

We now pick the h observations with the smallest outlyingness measures $outl(\mathbf{z}^{(i)^T})$ and call this matrix $\hat{\mathbf{Z}} = \hat{\mathbf{Z}}_{h,r_0}$.

We now let $S = Cov(\hat{\mathbf{Z}})$ be the robust covariance matrix and $\hat{\mu}_X = Mean(\hat{\mathbf{Z}})$ be the robust centre [2]. By performing eigen decomposition of **S** we can write it on the form [2].

$$\mathbf{S} = \mathbf{PLP}^T \tag{2.16}$$

Where **L** is a diagonal matrix of eigenvalues sorted in descending order. The matrix **P** is orthonormal and contains the eigenvectors. We can now use the truncated matrix $\mathbf{P}^* = \mathbf{P}_{r_0,k}$. $\mathbf{P}^*$ will have a similar function to **W** in the classical PCA and we can use it to make the transformation:

$$\mathbf{T}_{n,k} = \left(\mathbf{X}_{n,p} - \bar{\mathbf{X}}\right) \mathbf{P}^*_{p,k} \tag{2.17}$$

The algorithm can be terminated here but another round of improvements can be made. Please refer to [2] for the final step of the algorithm.

2.6 RSIMPLS: Robust SIMPLS

The RSIMPLS algorithm is a robust extension of SIMPLS. A major difference between the two is the fact that RSIMPLS uses a robust mean and scatter matrix instead of the empirical correlation and arithmetic mean used in the SIMPLS algorithm.

We use the ROBPCA algorithm to produce a robust scatter matrix. Given a concatenated matrix $\mathbf{Z}_{n,p+q} = [\mathbf{X}_{n,p}, \mathbf{Y}_{n,q}]$ the scatter matrix $\mathbf{S}_{p+q,p+q}$ of $\mathbf{Z}$ can be explained in terms of the scatter matrix of $\mathbf{X}$, $\mathbf{Y}$ and their joint scatter matrix.

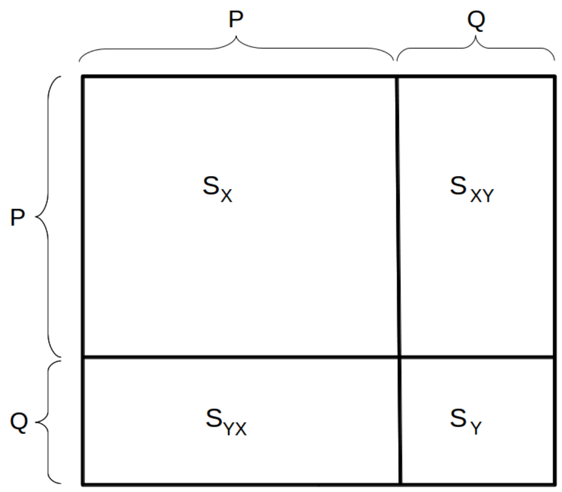


Figure 2.4: An illustration of how the robust scatter matrix $\mathbf{S}_Z = \mathbf{S}_{[X,Y]}$ of a joint matrix can be explained in terms of the original matrices.

The RSIMPLS also includes robust regression unlike SIMPLS. Algorithm 3 shows an outline of the RSIMPLS algorithm, for more details consult ”Robust methods for partial least squares regression” by Mia Hubert and Karlien Vanden Branden [1].

```
Input:
Predictors $\mathbf{X}_{n,p}$
Responses $\mathbf{Y}_{n,q}$
Non-negative integer $k_0 \leq n$
Result:
Regression parameters $\mathbf{B}$
Regression Intercept $\mathbf{B}_0$
Weight Matrix $\mathbf{W}$
Score Matrix $\mathbf{T}$
Algorithm:
$\mathbf{Z}_{n,p+q} = [\mathbf{X}_{n,p},\mathbf{Y}_{n,q}]$                 // Concatenation of $\mathbf{Y}$ and $\mathbf{Y}$
$\mathbf{S}_Z, \boldsymbol{\mu}_z \leftarrow ROBPCA(\mathbf{Z})$             // Robust scatter matrix and mean
$\mathbf{S}_{XY} = SUBMATRIX(\mathbf{S}_z,p+1,p+q,1,p)$
$\mathbf{S}_X = SUBMATRIX(\mathbf{S}_z,1,p,1,p)$
$\boldsymbol{\mu}_x = SUBVECTOR(\boldsymbol{\mu}_z,1,p)$
for $k \leftarrow 1$ to $k_0$ do
    $\mathbf{q}$ = dominant eigenvector of $\mathbf{S}_{XY}^T\mathbf{S}_{XY}$
    $\mathbf{w} = \mathbf{S}_{XY}\mathbf{q}$
    $\mathbf{t} = (\mathbf{X} - \boldsymbol{\mu}_x)\mathbf{w}$
    $\mathbf{t} = \mathbf{t} - \bar{\mathbf{t}}$
    $\mathbf{w} = \mathbf{w}/||\mathbf{t}||$
    $\mathbf{t} = \mathbf{t}/||\mathbf{t}||$
    $\mathbf{p} = (\mathbf{w}'\mathbf{S}_X\mathbf{r})^{-1}\mathbf{S}_X\mathbf{w}$
    $\mathbf{q} = \mathbf{Y}_0^T\mathbf{t}$
    $\mathbf{u} = \mathbf{Y}_0\mathbf{q}$
    $\mathbf{v} = \mathbf{p}$
    if $k > 1$ then
        $\mathbf{v} = \mathbf{v} - \mathbf{V}(\mathbf{V}^T\mathbf{p})$                              // Deflate $\mathbf{v}$
        $\mathbf{u} = \mathbf{u} - \mathbf{T}(\mathbf{T}^T\mathbf{u})$                              // Deflate $\mathbf{u}$
    end
    $\mathbf{v} = \mathbf{v}/||\mathbf{v}||$
    $\mathbf{S}_{XY} = \mathbf{S}_{XY} - \mathbf{v}(\mathbf{v}^T\mathbf{S}_{XY})$                          // Deflate $\mathbf{S}$
    $\mathbf{W} = [\mathbf{W},\mathbf{w}]$                                        // Append $\mathbf{w}$ to $\mathbf{W}$
    Append $\mathbf{t}$, $\mathbf{p}$, $\mathbf{q}$, $\mathbf{u}$, $\mathbf{v}$ to $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{U}$, $\mathbf{V}$ respectively
end
```

Algorithm 3: PLS step of the RSIMPLS algorithm as formulated by M. Hubert and K. Vanden Branden [1]

The second part of the algorithm performs robust regression. The can be done in a number of ways, in particular least trimmed squares (LTS) regression might be a good alternative if $\mathbf{Y}$ is univariate [1] [9]. If $\mathbf{Y}$ is multivariate another approach such as ROBPCA regression is needed. They all essentially select a subset of points in the score space for which ordinary MLR is applied and the results corrected. Since ROBPCA regression is fairly technical we refer to the article "Robust methods for partial least squares regression" by M. Hubert and K. Vanden Braden [1] for further details.

2.7 Model Quality Measures: R^2 and Q^2

To evaluate the performance of a regression model it is common to use the scalar measures R^2 and Q^2. R^2, sometimes called coefficient of determination, is a measure of the models descriptive quality or in other words how well it fits the set it was built on. R^2 is usually defined as follows [14]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_i\right)^2} \tag{2.18}$$

Since R^2 is a measure of descriptive quality, models can often be created with near perfect R^2 by increasing the internal degrees of freedom. In the PLS case a high number of components has this effect. However if the internal degrees of freedom is too high the model might be over fitted and lose predictive power on new data not included in the test set.

Q^2 is a measure of the models predictive performance on a new, for the model unseen, test set. Q^2 is defined in the same way as R^2 but only applies to observations not included in the model set.

$$Q^2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_i\right)^2} \tag{2.19}$$

A typical behavior of R^2 and Q^2 is illustrated in figure 2.5. Q^2 typically has a parabolic shape and there exists am optimum with regards to predictive power. In general $Q^2 < R^2$ but is often a better measure of the true model performance.

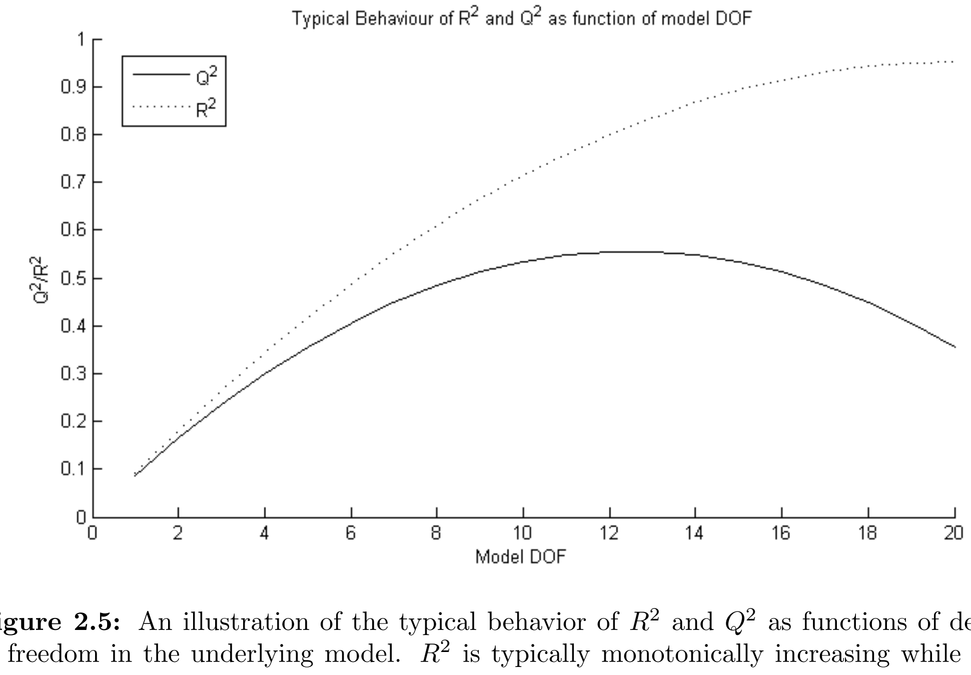


Figure 2.5: An illustration of the typical behavior of R^2 and Q^2 as functions of degrees of freedom in the underlying model. R^2 is typically monotonically increasing while Q^2 is parabolic and in this case has a maximum around 13 DOF.

A common way to estimate Q^2 is to use cross-validation. When cross-validating one repeatedly splits the data set in two sets. One part is used to build a statistical model and the other to evaluate the predictive power of the newly created model according to the formula above. Cross-validation is particularly useful when determining the RSIMPLS coefficients k and α.

2.8 Statistical Properties of RSIMPLS

RSIMPLS is a complex algorithm, which makes it hard to quantify specific statistical properties. RSIMPLS primarily filters two kinds of outliers, bad leverage points and residual outliers. The bad leverage points are characterized by their large TD and OD i.e. the points that does not fit the model and would have a large impact if included. The residual outliers are those points which are characterized by a large residual error in the preliminary regression step.

The removal of the bad leverage points might actually result in a performance loss of the model if the preliminary robust model is faulty. If the so called bad leverage points are falsely identified outliers their correcting effect will be removed if the points are discarded. The same argument holds for the residual outliers. Thus if the parameter α is set too high the RSIMPLS method is at risk of losing averaging benefits of falsely identified outliers.

2.9 The Underlying Physics

The physics behind the relationship between the stimulus and response in the Acospector can be partly explained by wave-length dependent attenuation. In general we have the following formula describing the amplitude of a decaying wave.

$$A(\omega) = A_0(\omega)e^{-\alpha(x,\omega)D} \tag{2.20}$$

Where $\alpha(x,\omega)$ is the attenuation coefficient, D the distance traveled (i.e. the diameter of the pipe) and A_0 the undamped amplitude. Note that α is dependent of some parameter x and the wavelength ω.

For sound traveling in a fluid the attenuation coefficient can be derived from 6 different physical mechanisms [6].

- **Electrokinetic Properties:** Oscillating dipoles generate an electric current which results in loss of wave energy
- **Intrinsic Properties:** This mechanism is due to the acoustical wave interaction with the fluid particles. This will typically result in loss of acoustical energy.
- **Scattering:** Scattering of acoustical waves will typically result in a loss of transmitted energy. This statement is particularly true for a point source and a single accelerometer.
- **Structural Properties:** The acoustic wave will generally propagate and excite the molecules in the whole system.
- **Thermodynamics:** The combination of temperature- and pressure gradients will lead to a loss of energy in the acoustic signal.
- **Viscosity:** Viscosity is essentially fluid friction. Since an acoustical wave is mechanical of nature this will lead to a loss of energy.

All mechanisms will contribute to the total attenuation constant but for wavelengths much larger than the particle diameter the effect of the structural- and electrokinetic properties become negligible and the remaining properties can be approximated as additive [6]. We then have:

$$\alpha(x,\omega) \approx \alpha_{intrinsic}(x,\omega) + \alpha_{scatter}(x,\omega) + \alpha_{thermo}(x,\omega) + \alpha_{viscosity}(x,\omega) \tag{2.21}$$

This means that we can re-write equation (2.20) as:

$$A(\omega) \approx A_0(\omega)e^{-\alpha_{intrinsic}(x,\omega)D}e^{-\alpha_{scatter}(x,\omega)D}e^{-\alpha_{thermo}(x,\omega)D}e^{-\alpha_{viscosity}(x,\omega)D} \tag{2.22}$$

Each of the attenuation constant will have a unique dependence of the measured fluid property x. When we create a linear model we make the assumption that the attenuation can be approximated as a linear function. This is a fairly bold claim and the linear approximation is not always good enough.

There are a few tricks one can do to increase the linearity of the system, however they will not be covered in this thesis. Great care should always be used when modeling complex systems with linear models.

When creating a statistical model of the acoustic response of a fluid one makes the implicit assumption that the system is stable of time. If the system is changing behavior over time (i.e. $\alpha = \alpha(t)$) the quality of a statistical model might decay. This is a general problem in statistics, a model can only describe a system as it was when the data was collected. In practice one would sample the system over an extended period of time to capture as much of the system variation as possible, implicitly assuming that the system might change over time but only in certain intervals.

3

Methods

This chapter covers the various methods used in this project including the software implementation and improvements/extensions made. Since the project was carried out in collaboration with the industry, implementation specific details and code has been left out of this report.

3.1 Implementation

The project was implemented in C++ using OpenBlas as a base library for numerical calculations. The software was divided into two parts, one library containing the base functionality and one user interface connecting to the library.

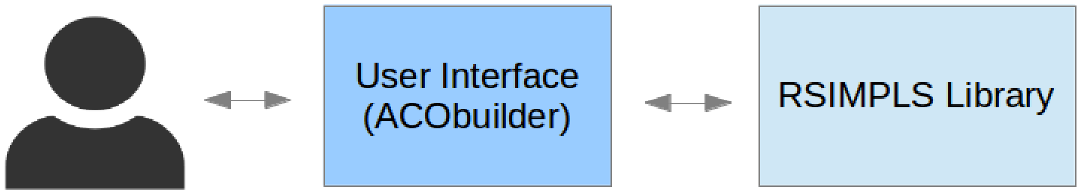


Figure 3.1: A general overview of the structure of the program. The underlying library is indirectly accessed via the user interface.

3.1.1 The RSIMPLS Library

The RSIMPLS library contains the base functionality of the application with the two most important functions being `rsimpls(x,y,k,alpha)` and `robpca(x,k,alpha)`. In addition, the library also contains methods for various matrix and vector operations and decompositions.

3.1.2 The User Interface

The user interface provides a way for the user to access the RSIMPLS and ROBPCA functionality. A basic command line interface was first implemented and later a graphical user interface was built to provide easy and intuitive access.

3.2 Sample Selection

The RSIMPLS algorithm contains several computationally heavy elements with the most time and memory consuming ones being the SVD and EVD matrix decompositions. SVD for example has a time complexity of $\mathcal{O}(min\{m^2n, mn^2\})$ for an $m \times n$ matrix [11].

To reduce the number of observations in the model building set one might want to do sample selection to reduce the number of samples and make the model building possible.

How do we find a subset of all samples to optimally build the model and represent all available data? Together with Johan Karlsson from Fraunhofer Chalmers-Centre we came up with the idea to first do a feature extraction of the observations and then choose a subset based on the features.

We decided to let the user choose an evenness parameter between 0 and 1 representing the fraction of the desired observations to be chosen with maximum internal distance while the remaining observations are chosen at random. Figure 3.2 shows some subsets for a few different subset sizes and evenness parameters.

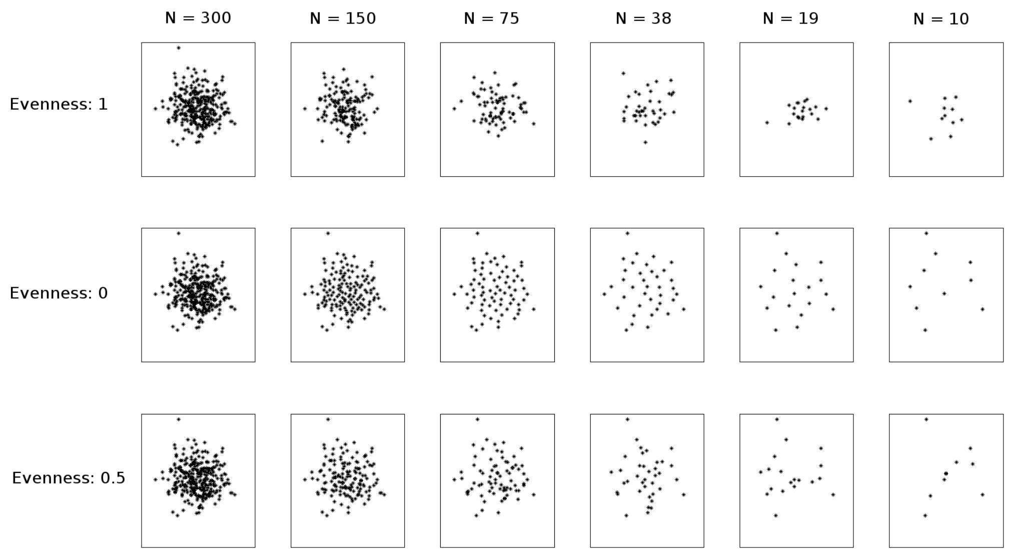


Figure 3.2: The figure illustrates how the evenness parameters effects choosing subsets of different sizes. The evenness parameter 1 represent the case where all points are chosen at random and the evenness parameter 0 represent the case where all point are selected to maximize the variation in the selected set.

To find features of our observations we first generate a few PCs by doing a PCA transform of a small number of randomly chosen observations. These PCs will form our feature space and all observations are projected onto it. The whole data set projected to the feature space is then passed to the sample selection algorithm together with an evenness parameter and the desired size of the final number of selected observations.

3.3 Pre-removal of distant points

In some cases there is a regular process abnormality that influence the observations. If too common they will not be regarded as outliers in the RSIMPLS algorithm by any reasonable choice of the parameter α. Including those observation results in decreased model performance and increased time and memory usage.

Figure 3.3 illustrate such a case where the points at the left in the picture are due to a regular disturbance of the system.

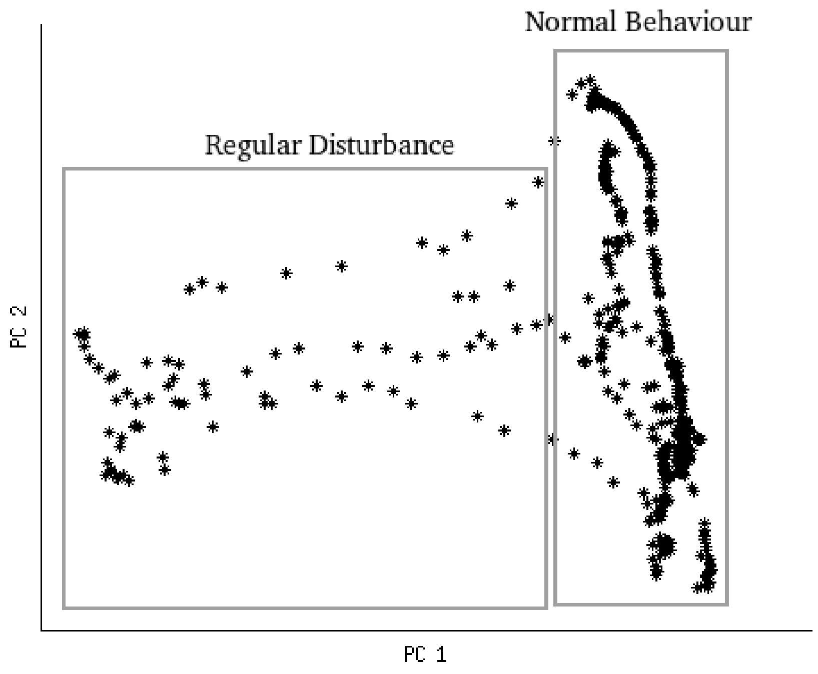


Figure 3.3: The figure illustrates how a recurring disturbance of a system could look in the PCA space. The points to the left in the picture are due to abnormal behaviour and should ideally be disregarded in the model building step.

To provide an additional tool for dealing with regular disturbances a pre-removal step was introduced. The pre-removal algorithm works by iteratively finding the observation with the largest combined distance to all other points and then remove it. It is required by the user to provide a fixed number of points to remove.

Pre-removal of distant points is a feature that should be used with great care and preferably not without prior knowledge about the underlying process. If used incorrectly it might remove key observations from the model set which might result in a decrease in model quality. It was implemented to add flexibility and additional tools for the end user of the software.

3.4 Verification

To verify the RSIMPLS algorithm we chose to compare it to a few other similar algorithms. Among these were SIMPLS, PCA-Regression (PCR) and canonical-correlation (CCA). SIMPLS was chosen for its close relation with RSIMPLS, PCA-Regression to show the importance of using PLS instead of PCA in the dimension reduction step and CCA for its historical importance and strength in certain applications.

4

Results

The software implementation of RSIMPLS was completed successfully. The RSIMPLS algorithm performance was comparable, but slightly worse, than CCA for low-dimensional regressors but better than CCA, SIMPLS and PCR for high-dimensional regressors. For simplicity the results were generated with univariate response variables, i.e. $\mathbf{Y} = \mathbf{Y}_{n,1}$ however there is no algorithmic limit regarding the dimensionality of the response variables.

4.1 Idealized Test Case

Consider a set of predictors $\mathbf{X}_{n,p}$ with $X_{i,j} \in \mathcal{N}(0,\sigma_x)$, that is all entries of $\mathbf{X}$ are normally distributed and centered around 0. Furthermore assume that the responses are univariate, $\mathbf{Y} = \mathbf{Y}_n$, and can be partially expressed in one term linearly dependent of $\mathbf{X}$ and one that is normally distributed.

$$Y_i = \sum_{j=1}^{n} (X_{i,j} v_j) + \mathcal{N}(0,\sigma_y) \tag{4.1}$$

Where $\mathbf{v}$ is a normalized vector describing the fundamental linear relationship between the two sets. Since both the $\mathbf{X}$ and $\mathbf{Y}$ matrices are assumed to be centered we state that the predictions made by the canonical correlation models are:

$$\hat{Y}_i = \sum_{j=1}^{p} X_{i,j} v_j^* \tag{4.2}$$

Where $\mathbf{v}^*$ is the empirical linear relationship found by the canonical correlation method. Consider the case where $n \to \infty$, that is the case of unlimited samples. Since the errors are normally distributed $\mathbf{v}^* \to \mathbf{v}$ when $n \to \infty$.

As a quality measure of the model quality we use the R^2 measure presented in (2.18) in the theory chapter. We have that:

$$\begin{aligned}
R^2 &= \lim_{n\to\infty} 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_i\right)^2} & (4.3)\\
&= \lim_{n\to\infty} 1 - \frac{\sum_{i=1}^{n}\mathcal{N}(0,\sigma_y)^2}{\sum_{i=1}^{n}\left(\sum_{j=1}^{p}(X_{i,j}v_j) + \mathcal{N}(0,\sigma_y) - \bar{Y}_i\right)^2} & (4.4)\\
&= \lim_{n\to\infty} 1 - \frac{\sum_{i=1}^{n}\mathcal{N}(0,\sigma_y)^2}{\sum_{i=1}^{n}\left(\sum_{j=1}^{p}(X_{i,j}v_j) + \mathcal{N}(0,\sigma_y)\right)^2} & (4.5)\\
&= \lim_{n\to\infty} 1 - \frac{\sum_{i=1}^{n}\mathcal{N}(0,\sigma_y)^2}{\sum_{i=1}^{n}\left(\left(\sum_{j=1}^{p}X_{i,j}v_j\right)^2 + \mathcal{N}(0,\sigma_y)^2\right)} & (4.6)\\
&= \lim_{n\to\infty} 1 - \frac{\sum_{i=1}^{n}\mathcal{N}(0,\sigma_y)^2}{\sum_{i=1}^{n}\left(\mathcal{N}(0,\sigma_x)^2 + \mathcal{N}(0,\sigma_y)^2\right)} & (4.7)\\
&= 1 - E\left(\frac{\mathcal{N}(0,\sigma_y)^2}{\mathcal{N}(0,\sigma_x)^2 + \mathcal{N}(0,\sigma_y)^2}\right) & (4.8)\\
&= 1 - \frac{\sigma_y^2}{\sigma_x^2 + \sigma_y^2} & (4.9)\\
&= \frac{\sigma_x^2}{\sigma_x^2 + \sigma_y^2} & (4.10)
\end{aligned}$$

Where we in eq (4.4) used that $\bar{\mathbf{Y}}$ is centered. In eq (4.6) we used the fact that $\mathbf{v}$ is normalized and that $\mathbf{X}$ is centered and normally distributed.

We use the result above as a performance measure when assessing the performance of RSIMPLS and canonical correlation. Figure 4.1 illustrates a simulation of the two regression methods for a number of different sample sizes.

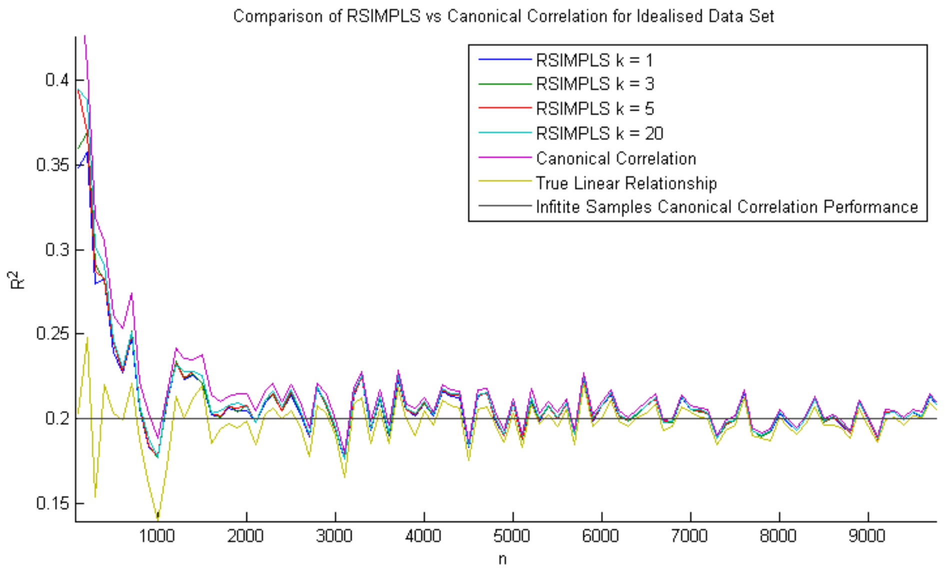


Figure 4.1: The figure illustrates the generated model quality for CCA and RSIMPLS with a few different number of components. As we can see both methods outperform the fundamental linear relationship and converges to the theoretical performance of canonical correlation for an infinite number of samples as n increases.

The simulation above was carried out with $\sigma_x = 1$ and $\sigma_y = 2$. As we can see canonical correlation outperforms RSIMPLS slightly in this scenario. This is expected since canonical correlation is particularly suited to be used in cases of idealized data. It is however more sensitive to noise and might underperform in other cases.

4.2 Low Dimensional Predictors

To assess the performance of RSIMPLS in a realistic setting the algorithm was tested on house pricing data from Ames. After removing all categorical data from the original data set 33 variables remained in the predictor set. The response set consisted of the univariate final selling price of the house. The total available amount of data was 2 930 entries.

A model set of $n = 150$ observations was chosen together with a separate, non-overlapping test set with 1 000 observations used to evaluate the model performance. Several models were generated by calling the RSIMPLS methods with different values of k and α. All models were evaluated using cross-validation. The result can be found in table 4.1.

$\mathbf{Q}^2(k,\alpha)$	$\alpha = 0.6$	$\alpha = 0.75$	$\alpha = 0.85$	$\alpha = 0.95$	$\alpha = 1$
k = 1	-0.0434	-0.0354	-0.0273	-0.0180	-0.0180
k = 2	0.4770	0.4815	0.4865	0.5156	0.5184
k = 3	0.5077	0.5137	0.5484	0.5324	0.5265
k = 4	0.5597	0.6069	0.6280	0.6529	0.6396
k = 5	0.6275	0.6058	0.6381	0.6763	0.6372
k = 6	0.6739	0.6722	0.6855	0.6767	0.6699
k = 7	0.5949	0.6690	0.6835	0.6687	0.6717
k = 8	0.6597	0.6939	0.6958	0.6852	0.6806
k = 9	-0.0322	0.6970	0.6862	0.6745	0.6559
k = 10	-0.0766	0.6761	0.6796	0.6696	0.6559
k = 11	-6.2805	-0.0695	0.6695	0.6624	0.6559
k = 12	0.0477	0.4898	0.6858	0.6734	0.6559

Table 4.1: The table shows the quality of a number of different models generated by the RSIMPLS method with different values of k and α. The table shows that if k is taken to be too large the model loses its predictive power and the quality decreases. There is also a trade off when choosing α. A small α gives a higher resistance towards outliers but will reduce the effective number of observations used in the model and can result in the discarding key observations.

In addition to the RSIMPLS models a canonical correlation model was created using the same model set. The canonical correlation model performed slightly better than all RSIMPLS models with an estimated Q^2 of 0.8251 by cross-validation. Figure 4.2 shows the first 140 entries in the test set.

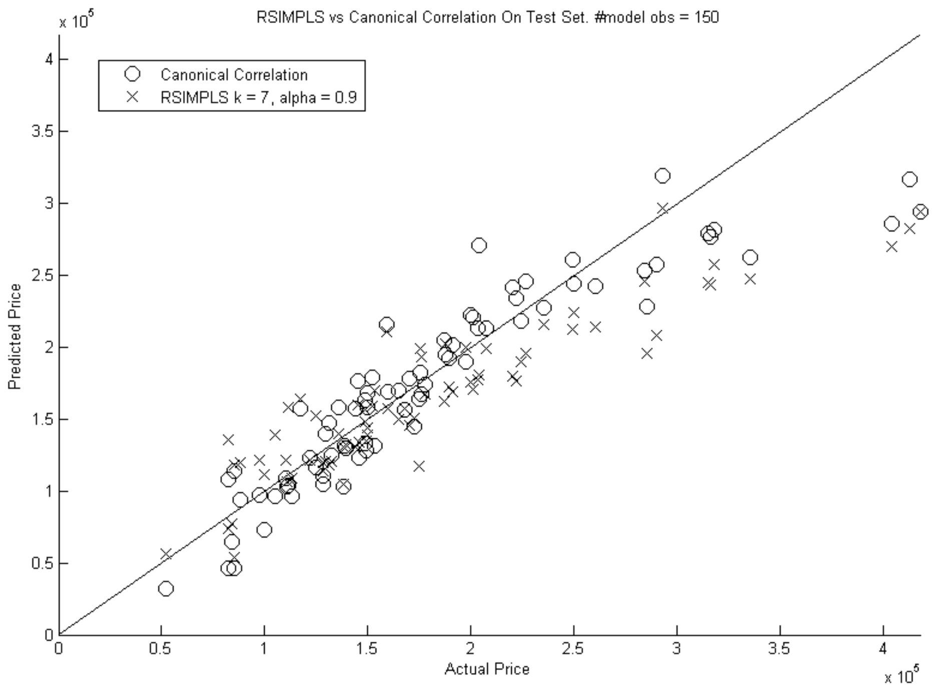


Figure 4.2: The figure shows the predictions from a well performing RSIMPLS model and a canonical correlation model on the y axis with the reference value from the data set on the x axis. This scatter plot might indicate a slight non-linearity in the data.

4.3 High Dimensional Predictors

To compare RSIMPLS to other similar methods when dealing with high dimensional predictors a data set from an experiment carried out by Acosense was examined. The data set contained 286 observations, 1 univariate response variable and > 4000 predictor variables. The methods PCR, SIMPLS, RSIMPLS and CCA were all used for 20 different permutations of 5 model set sizes. R^2 and Q^2 were calculated for each permutation and model size and averaged, using cross-validation for Q^2. The results are presented in table 4.2

	$\mathbf{R^2/Q^2}$	**PCR**	**SIMPLS**	**RSIMPLS**	**RSIMPLS**	**CCA**
		$k = 8$	$k = 8$	$k = 8, \alpha = 0.8$	$k = 8, \alpha = 0.9$	-
$n = 50$	R^2	0.4347	0.2598	0.4807	0.4646	1.0000
	Q^2	0.2407	0.1301	0.3131	0.2864	-3.5064
$n = 100$	R^2	0.3613	0.3646	0.5126	0.4924	1.0000
	Q^2	0.2342	0.2138	0.3644	0.3437	-5.1853
$n = 150$	R^2	0.2982	0.3804	0.4817	0.4771	1.0000
	Q^2	0.2199	0.2850	0.4054	0.3880	-6.9541
$n = 200$	R^2	0.2793	0.4104	0.5055	0.4951	1.0000
	Q^2	0.1770	0.2393	0.3910	0.3758	-6.4495
$n = 250$	R^2	0.2654	0.4234	0.5079	0.4865	1.0000
	Q^2	0.1816	0.3459	0.4454	0.4183	-7.5125

Table 4.2: Performance of PCR, SIMPLS, RSIMPLS and CCA for various model set sizes. In this experiment RSIMPLS outperforms all other methods and CCA breaks down.

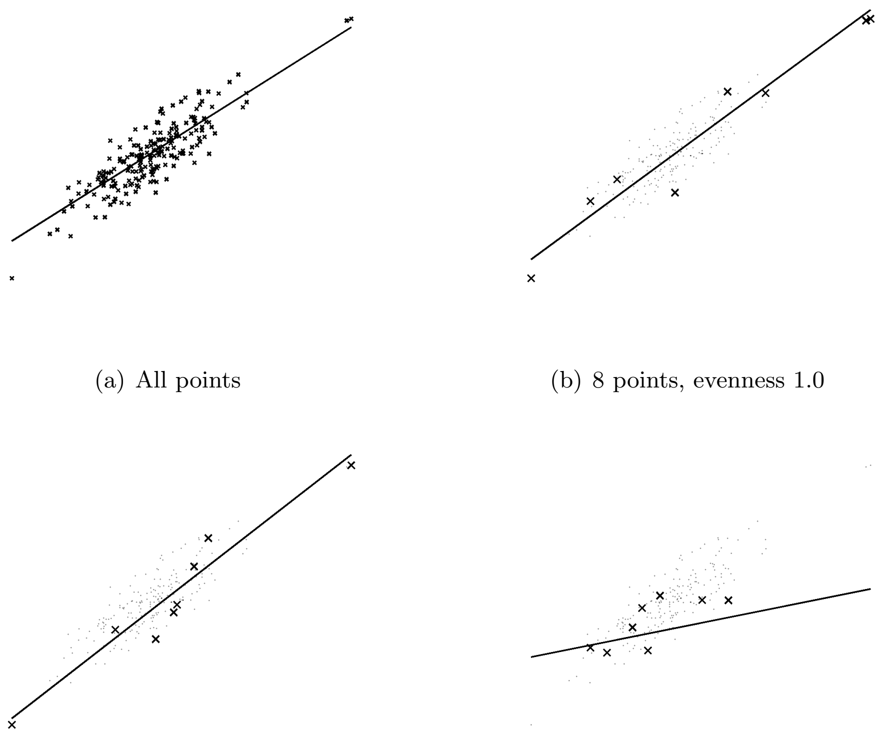
4.4 Sample Selection

We evaluate the sample selection algorithm presented in the methods chapter on two data sets, one linear and one slightly non-linear. In order for the results to be illustrative two dimensional regressors with distinct extreme points were used. In total 200 observations were generated, 196 of which being centered normally distributed of unit variance and 6 of which being of variance 4.

The linear response variable $\mathbf{Y}$ was taken as the sum of the regressors with an added noise term of unit variance i.e:

$$Y_i = \sum_{j=1}^{2} X_{i,j} + \mathcal{N}(0,1) \tag{4.11}$$

From the randomly generated $\mathbf{X}$ and $\mathbf{Y}$ described above, 8 samples were chosen by the sample selection algorithm using three different evenness parameters (1.0, 0.5 and 0.0). Note that this selection is random or partly random for all evenness parameters not equal to 1. The resulting models are presented in figure 4.3 together with a model built using all 200 samples. The x axis is $\sum_{j=1}^{2} X_{i,j}$ in all plots.


(a) All points (b) 8 points, evenness 1.0

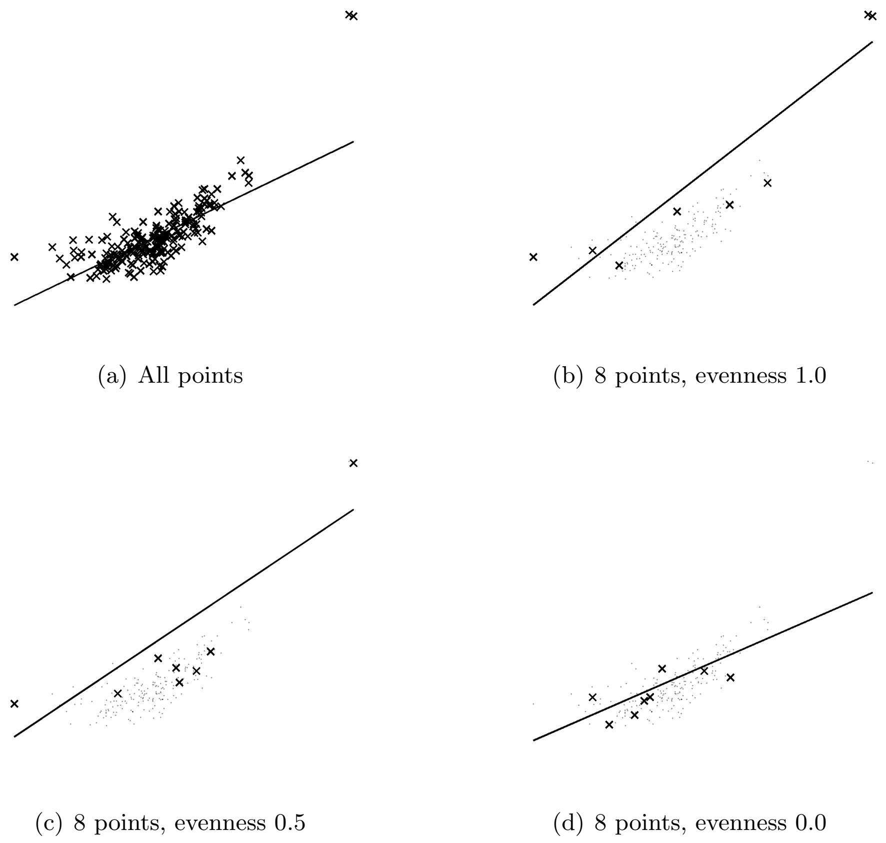
(c) 8 points, evenness 0.5 (d) 8 points, evenness 0.0

Figure 4.3: The sample selection algorithm helps identify the extreme points which bring stability to the model. Since the extreme points are few they are missed in the lower right plot where the evenness parameter 0 was used (random selection). Note that the x axis is $\sum_{j=1}^{2} X_{i,j}$ in all plots and the y axis the y values.

Similarly to the linear case presented above a non-linear response variable was generated using the square of the sum instead.

$$Y_i = \left(\sum_{j=1}^{2} X_{i,j} \right)^2 + \mathcal{N}(0,1) \tag{4.12}$$

Figure 4.4 presents the results for the non-linear response variable. Notice how the outliers greatly influences the model and how they are selected by the sample selection algorithm.


(a) All points

(b) 8 points, evenness 1.0

(c) 8 points, evenness 0.5

(d) 8 points, evenness 0.0

Figure 4.4: In this model we are creating a linear model of a non-linear response variable. The response variable is fairly linear around 0 but grows quadratically. The sample selection algorithm will in general select extreme points given a high evenness parameter. If the response variable is non-linear in terms of the given regressors a high evenness parameter will have a great impact on the resulting model.

5

Discussion

In this thesis we have seen how we can extend the traditional PLS algorithms to include resistance towards outliers by using outlying measures. This resistance is particularly useful when dealing with a few contaminated samples in a set which otherwise might have a large negative impact on the model.

RSIMPLS is a powerful method with several performance benefits over other comparable methods in certain situations. In this thesis we have seen that RSIMPLS is a good alternative when dealing with high dimensional, highly collinear, predictor sets where CCA breaks down completely and PCR performs poorly.

For the real data set with few predictors and the idealized data set RSIMPLS successfully created linear models but was slightly outperformed by CCA. They however both converged to the same theoretical performance in the idealized test case when the sample size grew bigger and outperformed the model based on the true underlying linear model. This indicates that both methods successfully captured the actual structure of the data rather than the fundamental structure.

Looking back at the problem definition formulated in section 1.2 we have successfully implemented and evaluated the RSIMPLS method. The RSIMPLS algorithm has been successfully applied to numerous test cases, many of which not presented in the result section of this thesis. The implementation has been included into Acosense's toolbox and is now frequently used. Furthermore RSIMPLS as an algorithm has been evaluated and compared to similar methods.

There are several possible extensions to this project. One would be to use elastic net regularization when determining the loadings to let the components be more frequency specific (in the case of acoustic spectra). Another possible extension would be to replace the ROBPCA method with ROBPCA-kmax [15]. This would eliminate the need of finding a suitable k when building a statistical model.

5.1 Typical Use Case

How does this ultimately comes together as a useful tool to create statistical models? The first step is data acquisition and preparation. Collect all available predictor data (e.g. spectrum data) and compile it into matrix form with each observation being one row. Next, we apply the same procedure to the response variables we want to model and relate to our predictors. This will result in the two matrices $\mathbf{X}$ and $\mathbf{Y}$.

The next step is to build a model with strong predictive qualities i.e a high Q^2. Since the models with high k tend to be over fitted it is recommended to design a test procedure based on cross-validation. A strong recommendation is to separate the model set and verification set in time to avoid to fully test the models predictive powers. Also note that the size of the model set and verification set is highly dependent of the amount of available data but a good rule of thumb is to use approximately $^1/_3$ of the total data for the verification set. It is also recommended to generate a few different pairs of model sets and verification sets and average the results to minimize the impact of a lucky choice.

With a test procedure in place, the search of a good model can begin. The RSIMPLS has k and α as intrinsic parameters for which the function RSIMPLS(k,α) could be optimized over for maximum Q^2. Sample selection and pre removal of distant points also brings in parameters that can be used in the search of a good model. Basically any generic optimization routine could be used or an extensive search if the problem size/computing performance is good enough.

6

Conclusion

When comparing RSIMPLS with similar methods we found that RSIMPLS is particularly strong dealing with high dimensional regressors, a case where CCA broke down and PCR performed poorly with respect to predictive power. For data sets with fewer predictor variables CCA and RSIMPLS performed about equally with a slightly better performance for CCA.

Besides performing slightly better for lower dimensional predictor sets CCA is in general a simpler and faster algorithm. These facts combined makes CCA a good first alternative to try when faced with the problem of creating a linear model for a data set with few predictors. However if the data set contains several outliers RSIMPLS might have a performance edge.

For high dimensional predictor sets RSIMPLS performed noticeable better than comparable methods. PCR might work in some cases but since it maximizes variance in the dimension reduction step no guaranties can be made about the scores predictive power of the responses. It's not uncommon that noisy predictors with high variance are also being the ones that are the least correlated with the response variables.

The downside to RSIMPLS is that it is a fairly complex algorithm to implement and is computationally heavy while not necessarily performing better than SIMPLS or CCA for certain data sets. Therefore I recommend a thorough pre-study of the properties of the problem at hand before implementing RSIMPLS. One could also start by implementing SIMPLS for a first evaluation and then, if necessary, extend the algorithm to RSIMPLS.

Bibliography

[1] M. Hubert and K. V. Branden (2003) ”Robust Methods for Partial Least Squares Regression”, *Journal of Chemometrics*, 17, 537-549.

[2] M. Hubert, P. J. Rousseuw and K. V. Branden (2005) ”ROBPCA: A New Approach to Robust Principal Component Analysis”, *Technometrics*, 47:1, 64-79.

[3] Sijmen de Jong (1992) ”SIMPLS: an alternative approach to partial least squares regression”, *Chemometrics and Intelligent Laboratory Systems*, 18, 251-263.

[4] S. Wold, M. Sjöström and L. Eriksson (2001) ”PLS-regression: a basic tool of chemometrics”, *Chemometrics and Intelligent Laboratory Systems*, 58, 109-130.

[5] S. Wold, k. Esbensen and P. Geladi (1987) ”Principal Component Analysis”, *Chemometrics and Intelligent Laboratory Systems*, 2, 37-52.

[6] F. Törner (2010) ”Analys av Svartlut med Aktiv Akustisk Spektrometri”, Linköping: Linköpings Universitet. (Examensarbete inom Institutionen för Teknisk Biologi)

[7] S. Wold, H. Martens and H. Wold (1983) ”The Multivariate Calibration Problem in Chemistry Solved by the PLS Method”, *Matrix Pencils*, Springer Berlin Heidelberg, 286-293.

[8] H. Wold (1974) ”Causal flows with latent variables: Partings of the ways in the light of NIPALS modelling”, *European Economic Review*, North-Holland Publishing Company, 5, 67-86.

[9] Peter J. Rousseuw (1984) ”Least Median of Squares Regression”, *Journal of the American Statistical Association*, 79-388

[10] Jan J. Gerbrands (1980) ”On The relationships betsween SVD, KLT and PCA”, *Pattern Recognition*, Pergamon Press Ltd, 14 375-381

[11] M. Holmes, A. Gray and C. Isbell (2007) ”Fast SVD for large-scale matrices”, *In Workshop on Efficient Machine Learning at NIPS.*

[12] L. Eriksson, E. Johansson, N. Kettaneh-Wold, J. Tryggm C. Wikström, S. Wold (2006) ”Multi- and Megavariate Data Analysis Part I Basic Principles and Applications, 2:nd edition”, Umetrics.

[13] I.T. Jolliffe (2002) "Principal Component Analysis (Springer Series in Statistics) 2nd Edition", Springer.

[14] J.A. Rice (2007) "Mathematical Statistics and Data Analysis", Cengage Learning.

[15] S. Engelen, M. Hubert, and K. Vanden Branden (2005) "A Comparison of Three Procedures for Robust PCA in High Dimensions", *Austrian Journal of Statistics*, 32:2, 117-126.

[16] Donald F. Morrison (1990) "Multivariate Statistical Methods 3rd edition", McGraw-Hill.