IMPROVEMENTS TO PLS METHODOLOGY

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2015

Alastair Campbell Bissett
School of Mathematics

Contents

Word count 46,435

List of Tables

List of Figures

The University of Manchester

Alastair Campbell Bissett
Doctor of Philosophy
Improvements to PLS Methodology
March 27, 2015

Partial Least Squares (PLS) is an important statistical technique with multiple and diverse applications, used as an effective regression method for correlated or collinear datasets or for datasets that are not full rank for other reasons. A short history of PLS is followed by a review of the publications where the issues with the application PLS that have been discussed. The theoretical basis of PLS is developed from the single value decomposition of the covariance, so that the strong links between principal components analysis and within the various PLS algorithms appear as a natural consequence.

Latent variable selection by crossvalidation, permutation and information criteria are examined. A method for plotting crossvalidation results is proposed that makes latent variable selection less ambiguous than conventional plots. Novel and practical methods are proposed to extend published methods for latent variable selection by both permutation and information criteria from univariate PLS1 models to PLS2 multivariate cases. The numerical method proposed for information criteria is also more general than the algebraic methods for PLS1 that have been recently published as it does not assume any particular form for the PLS regression coefficients. All of these methods have been critically assessed using a number of datasets, selected specifically to represent a diverse set of dimensions and covariance structures.

Methods for simulating multivariate datasets were developed that allow control of correlation and collinearity in both regressors and responses independently. This development also allows control over the variate distributions. Statistical design of experiments was used to generate plans for the simulation that allowed the factors that influence PLS model fit and latent variable selection. It was found that all the latent variable selection methods in the simulation tend to overfit and the feature in the simulation that causes overfitting has been identified.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

I would like to acknowledge the financial support from Federal-Mogul Friction Products Ltd for this study and to thank my colleagues for their support and encouragement. I would also like to thank Dr Mike Tso and Dr Alexander Donev for their supervision, discussions and help with this project as it evolved from industrial curiosity into this academic thesis. I must also thank Joan, for putting up with it all.

Chapter 1

Introduction

After using partial least squares analysis in a number of successful industrial projects it became clear to me that it is a powerful technique for analysing unstructured datasets. For studies where there the amount of available information is small, designed experiments with multiple regression and optimization are very efficient tools. Where there was already a collection of experimental data it was found that PLS could usually produce effective models for optimization without further experimentation even though the dataset contained some collinearity or partial correlations. So how was PLS achieving this? Even after some background reading I was left with no clear understanding of how the method actually worked. Also, the algorithmic nature of the fitting does tend to promote a "black box" view of the analysis. As the purpose of these industrial experiments was the product development of safety critical components, it was essential to have a thorough understanding of all the methods that are used. So rather than abandon such a successful technique, I started a systematic review of the PLS literature with the aim of identifying then adopting best practice. But instead of identifying an optimal method, the conclusions from this review were that as there was considerable variation in the way the PLS was being applied and interpreted. So an overall optimal method that we could call "best practice" probably does not exist. The desire for a more fundamental and rigorous approach to resolving these issues lead to continuing this work as an academic study. Understanding the mathematical and statistical framework behind PLS has clarified many aspects of its application. During the course of this work a number of improvements to PLS methods have been identified, which are the subject of this thesis. To set the context for

describing these improvements, this introduction continues with a brief history of PLS followed by some comments on the apparent problems and finishes with a statement of the specific objectives of this work.

1.1 A Brief History of Partial Least Squares

As statistical techniques evolve, it is often difficult to identify any single publication as the original source of a method. For such a wide field as PLS, any brief history cannot be comprehensive so the objective here is to trace the themes that lead to PLS and to identify only the key sources.

PLS was developed from principal components analysis (PCA) and principal components regression (PCR). Jolliffe[58] gives a review of the origins of PCA and puts the original source for PCA as a statistical method as Pearson[87] in 1901. But Cauchy's eigen analysis was published around 1829 and the singular value decomposition (SVD) was from Beltrami in 1873. The use of principal component scores as regressors was first published in 1957 independently by Hotelling[53] and Kendall[60]. This original form of PCA and PCR requires the complete evaluation of all eigenvectors and eigenvalues, but this was not feasible for high dimensional data until computing power became readily available. Geladi[37] describes development of iterative methods such as PCA NIPALS by Hermann Wold in the 1930s as a practical numerical method for evaluating the first few latent variables.

In 1961, Horst[49] describes an iterative method for orthogonal regression that would be described as a PLS method if published today. The basis for this paper is Hotelling's original paper from 1936[51] on canonical analysis that also contains the essential principles of PLS. In a review of the development of PLS methods, Geladi[37] describes the evolution of iterative PCA into PLS and gives the the first open publication of the PLS NIPALS method by Hermann Wold in 1975[111]. This is considered to be the source of PLS because the NIPALS algorithm was the first practical calculation method. The application for this 1975 publication was as a path modelling method for econometric models. The acronym NIPALS stands for "Non-linear Iterative Partial

Least Squares", with the "partial" to indicate that the score vectors are considered fixed at each iteration.

Geladi[37] points to a symposium in 1979 as the apex of the wide and rapid expansion of PLS applications in both natural and social sciences in the 1980s. During the 1980s, the first paper on PLS2 methods for multivariate responses by Svante Wold et al[115] was published. In particular, the 1983 publication of Wold, Martens and Wold[114] for the first application of PLS to multivariate calibration was very significant for analytical chemistry. Also during this period, two journals for "Chemometrics" appeared supported in the early days at least by the chemical applications of PLS. Herman Wold's son Svante Wold gives a interesting personal insight as a chemist into this development period of PLS methods[118]. In 1982, Herman Wold[113] published the application of PLS methods to multiblock analysis, which lead to methods that became known as path modelling and structural equation modelling. It is apparent in the literature that the applications of PLS in the chemical and other natural sciences use PLS as a predictive regression method diverged from path modelling and structural equation modelling which are nearly exclusive to econometrics and the other social sciences.

These early PLS publications are characterised by concentrating on the algorithms and only describing the fitting in terms of scores and loadings. The focus for the development of this method had been to develop a practical tool for analysing complex data, rather than for statistical rigour. At the end of the 1980s, papers examining the statistical aspects of PLS began to appear. In particular Helland's 1988 paper[47], Stone and Brooks in 1990[100] and Garthwaite in 1994[36] are key publications that identified objective functions, model forms and what was actually being optimized in PLS models. While considerable efforts have been made by the mathematical and statistical community to establish a rigorous foundation to PLS, this basic work has been nearly entirely on PLS1 univariate responses while basis for the important applications of PLS2 multivariate responses or multi-block PLS has been neglected.

The SIMPLS algorithm for PLS based on the direct deflation of the covariance

matrix was published by de Jong in 1993[16]. This algorithm is applicable to both PLS1 and PLS2 and is numerically more efficient for large datasets than NIPALS. The statistical aspects of PLS are generally clearer in the SIMPLS representation of PLS than in the earlier NIPALS method. Phatak and de Jong in 1997[90] made clear the geometrical aspects of PLS. A version of PLS based on bidiagonalization was first published by Manne[74] in 1987, but a later paper[122] in 2000 makes clear the computational advantage over NIPALS and SIMPLS for very large datasets.

Other developments of PLS during the 1990s and 2000s generally increased the range of applications for PLS, rather than extending the understanding. Quantitative Structure-Activity Relationships dominated the chemometrics literature in the 1990s. This is essentially PLS applied to identifying features of molecular structure related to chemical or pharmaceutical activity and became an important tool for drug discovery during this time. In 1998 Svante Wold et al[117] published orthogonal signal correction for spectroscopic data to remove systematic variation from the response matrix that is orthogonal and so unrelated to the property matrix. This filtering method was later incorporated directly into a PLS algorithm as orthogonal-pls ("O-PLS")[103]. This version of PLS is behind the emergence of genomics as a science in the 2000s as reviewed by Fonville et al in 2010[33]. Applications of PLS reach far beyond the original spectroscopic applications for analytical chemistry. PLS was used in almost one third of the structural path models reported in the top three management information systems journals between 2000 and 2003[42].

Although PLS was not developed as an analysis for categorical responses, it has become an important classification methods for collinear data. Barker and Rayen's paper in 2003[5] is perhaps typical of PLS publications in that it cites ten prior references to applications of PLS-discriminant analysis yet is the first publication to derive the theoretical basis behind the method. For the theoretical development of PLS during the 2000s, probably the key publication was by Pell, Ramos and Manne in 2007[88]. It's publication generated strong debate and comment by all the leading authors that clarified many subtle aspects of PLS methods.

A good explanation for the success of PLS in practice comes as some comments from Martens and Martens[79] on p385
"If $\mathbf{X}$ has clear structure then PLSR will use this to stabilise the regression model against noise in $\mathbf{Y}$. ...On the other hand, if $\mathbf{X}$ has no correlations between it's variables, but $\mathbf{Y}$ has, then the PLSR model will reveal this factor structure in $\mathbf{Y}$."

1.2 How PLS Relates to Other Multivariate Regression Methods

There are many ways that the various multivariate methods are can be related to each other. The connections between PLS and other regressions fall into two sets, according to univariate responses or multivariate responses. So in this respect, PLS forms a key link between regressions methods.

As methods for univariate responses, the brief history of PLS has described how both PCA and PLS became practical statistical methods with the development of NIPALS as simple computational method for extracting the first few latent variables. Lorber, Wangen and Kowalski [71] was an early paper that showed the strong connection between PCR and PLS by relating both to the singular value decomposition that is at the core of each method. Stone and Brooks[100] linked PCR, PLS and multiple linear regression (MLR) together by proposing a generalised regression criterion they called continuum regression. Here a controlling parameter determines the nature of the regression, with PCR, PLS and MLR appearing at specific parameter values. This approach recognised that these were fundamentally similar regression techniques but with different objectives, where MLR maximized correlation, PLS maximized covariance and PCR maximized variance. Sundberg[101] extended continuum regression to include ridge regression.

For regressions with multivariate responses, Tso [102] showed the strong connection between canonical analysis and reduced rank regression, (RRR). Burnham, Viveros

and MacGregor[11] applied a similar strategy as Stone and Brooks to make a continuum regression for multivariate responses. This linked canonical coordinate regression(CCR) to reduced rank regression and PLS. Again the framework of continuum regression showed that these were fundamentally similar regressions with different objectives. CCR maximizes correlation, PLS maximizes covariance while RRR has an intermediate objective.

1.3 Some Problems with PLS

From the brief history of PLS, it is clear that it has become a key part of many existing and emerging technologies. Perhaps because of the range of algorithms and diversity of applications and algorithms, PLS does not appear to have a clear, consistent and reliable methodology for model fitting and diagnostics. The text by Martens and Martens from 2001[79] is probably the most complete methodology to date. But in another publication from 2001[78], the same authors identified some of the unresolved aspects of PLS as a regression method and stated

"It is important that the statistical properties of the PLSR method are studied from a theoretical point of view; otherwise, it will not be accepted in mainstream science."

From their chemometrics perspective, these authors gave a summary of the tools required at that time for PLS modelling, shown here as Table 1.1 which is taken directly from[78].

(a)	Estimates of the optimal model complexity (its ranks = # of PCs, $AOpt$).
(b)	Estimates of the model's predictive uncertainty its Root Mean Error of Prediction RMSEP in $\mathbf{X}$ and in $\mathbf{Y}$.
(c)	Automatic identification of outliers.
(d)	Estimates of the uncertainty of the RMSEP estimate itself.
(e)	Estimates of the reliability/statistical significance of the linear and bi-linear mode parameters $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{B}$, etc.
(f)	Simple identification and elimination of unreliable input variables.

Table 1.1: Validation and Optimization Tools Needed in PLS Modelling

The issue of latent variable selection is the first item on this list and is a critical step in developing any multivariate regression model. Selecting too few and the model is under fitted, so that information about the response remains in the residuals and model fit is reduced. Selecting too many latent variables and the model is over fitted, so that some portion of the random error is included in the model and predictive performance is reduced. Based on their practical experience Wold, Sjöström and Eriksson[119] warn that with numerous and correlated regressors there is substantial risk of over-fitting so a reliable test for the significance of each consecutive latent variable is necessary. Clark and Cramer[14] have shown that PLS is capable of identifying correlations between scores even for random data, particularly when the number of latent variables approaches the number of trials, as is typical for small datasets. Consequently, rigorous methods for latent variable selection are essential as the primary guard against this type of over-fitting.

Since 2001 there have been many important developments in PLS. Due to the commercial significance of many of the PLS applications, it could well be that a lot more is really known about PLS than has been openly published. But comments in publications since 2001 show that these fundamental issues with PLS modelling and diagnostics are not being resolved. In 2006, Marcoulides and Saunders[75] used the Editor's Comments in an information science journal to warn authors about the dangers of applying PLS to small sample size datasets. This warning was repeated by the same authors in 2009[76].

The theoretical advances in PLS before and after 2001 have mainly concerned PLS1, not the multivariate PLS2 that is important for practical applications. In 2010 and from a statistical perspective, Chun and Keles[13] stated

"There are limited or virtually no results on the theoretical properties of PLS regression within the context of a multivariate response."

In 2010 as part of the Springer Handbooks of Computational Statistics series Vinzi et al edited a comprehensive "Handbook of Partial Least Squares"[107] . Even though 140 pages are dedicated to PLS tutorials, one reviewer[121] stated that

"... the book lacks sufficient guidelines to alert readers what "not to do" when using PLS in their research projects."

A paper by Kvalheim[66] on chemometrics also from 2010 starts with the words "Interpretation of PLS regression models has become a major task over the past decade".

So even though PLS methods has been has been used extensively, it is clear that many of the practical aspects of fitting PLS models, fit diagnostics and model interpretation have been uncertain for many years.

1.4 The Objective and Structure

The objective of this thesis is to propose some improvements of the current methodology for PLS regression in model fitting and model diagnostics. In particular to identify reliable methods for latent variable selection that are applicable to datasets with either univariate or multivariate responses and that are also robust across a wide range of dataset dimensions and characteristics. In order to do this thoroughly, path modelling, multi-block PLS and other extensions to the core form of PLS such as orthogonal filters for O-PLS have been excluded. Six example datasets have been used to provide evidence for the effectiveness of the methods proposed as improvements. Particular attention is given to how the nature of the dataset may influence the selection of PLS methods. Original work has only been included where it may be relevant as best practice.

The following chapter on the PLS Algorithms develops PLS from the point of view of single value decomposition of the covariance. This approach makes clear how the various algorithms are related, rather than trying to infer the relationships between algorithms from their common properties. After the example datasets have been described, latent variable selection methods are examined in detail. The traditional method for PLS latent variable selection is crossvalidation, which has had many ad hoc variations and test methods often assuming multivariate normality. Permutation

tests have been proposed as improvements as these hypothesis tests do not assume any specific underlying distribution. More recently, developments in calculating the degrees of freedom involved in PLS regressions has lead to the use of information criteria for latent variable selection. These three different methods are examined and compared. An extensive simulation study is reported that shows how dataset structure is related to PLS model performance. In particular, how PLS model coefficients and predictions are related to latent variable selection. The latent variable selection methods are also compared as part of this simulation. As a conclusion, the final chapter proposes a method of latent variable selection that may be considered as "best practice". It also contains a summary of the original work and concludes with some comments on how this work might be extended.

For consistency and to ensure that any part of this work may be easily reproduced, all the results presented here have used PLS calculations from the pls R library by Mevik, Wehrens and Liland[82]. All datasets, R code and results tables are available on request. During the development of this work, most the analysis was also made with code written in MATLAB direct from the original sources. The function of this code was verified against example datasets with some commercial software packages and against the pls R library. Where other R libraries have been used for specific calculations this is made clear in the text.

PLS publications can be confusing because they use a variety of notations that appear at first sight to be very similar, but actually contain subtle differences. I have tried to make the notation shown in Table A the simplest form that does not require a lot of unnecessary transposes. This is very close to the style used by Burnham, Viveros and MacGregor[11].

Chapter 2

The PLS Algorithms

Before examining the methodology of fitting and interpreting PLS models, it is necessary to establish the mathematical and statistical foundations for the method. The purpose of this chapter is to show what PLS is actually doing, to prove the relevant properties and show the implications for the data and model spaces that are referred to later. The PLS style has been used where scores are column vectors and loadings are row vectors. All notation used is summarised as an appendix.

As PLS methods have evolved over the years, the theoretical basis now forms a collection of related algorithms. Consequently, the literature contains a lot of fairly complex proofs that show equivalence or establish relation between the various algorithms. With the great benefit of hindsight, this review starts from the basic definition and simplest form of PLS, then proceeds to show how the main algorithms are related to this. The development of PCA/PCR into PLS was mentioned in the introduction. The dimension reduction aspect of PCA is examined first as an introduction to the basic form of PLS. To simplify the notation in this chapter, all observation/regressor matrices $\mathbf{X}$ and responses $\mathbf{Y}$ are assumed to be centred. They may or may not be scaled without loss of generality. The standard forms of covariance matrices are shown here to avoid ambiguity with so specific terms. The notation here implies that these variances and covariances refer to sample statistics not populations as this is more relevant to the PLS algorithms.

Let $\mathbf{S_X}$ be the variance-covariance of a matrix $\mathbf{X}$ defined by

$$\mathbf{S_X} = E\left[(\mathbf{X} - E[\mathbf{X}])(\mathbf{X} - E[\mathbf{X}])^T\right] \tag{2.1}$$

If $\mathbf{X} \in \mathbb{R}^{n\times k}$ then $\mathbf{S_X} \in \mathbb{R}^{k\times k}$.

Let $\mathbf{S_{X,Y}}$ be the cross-covariance matrix between two matrices $\mathbf{X} \in \mathbb{R}^{n\times k}$ and $\mathbf{Y} \in \mathbb{R}^{n\times m}$ be defined by

$$\mathbf{S_{X,Y}} = E\left[(\mathbf{X} - E[\mathbf{X}])(\mathbf{Y} - E[\mathbf{Y}])^T\right] \tag{2.2}$$

and $\mathbf{S_{X,Y}} \in (k \times m)$.

The covariance matrix for the regression between the two matrices $\mathbf{X}$ and $\mathbf{Y}$ is defined as

$$\mathbf{S} = \begin{vmatrix} \mathbf{S_X} & \mathbf{S_{X,Y}} \\ \mathbf{S}^T_{\mathbf{X,Y}} & \mathbf{S_Y} \end{vmatrix} \tag{2.3}$$

so $\mathbf{S} \in \mathbb{R}^{(k+m)\times(k+m)}$ and whose elements are the scalar variances and covariances within and between the columns of $\mathbf{X}$ and $\mathbf{Y}$.

2.1 Principal Components Analysis

At the core of both PCA and the PLS algorithms is the same orthogonal projection of the original data into a reduced rank subspace. The standard derivation of PCA is well known, for example Jolliffe[58]. This starts with a column centred observation matrix $\mathbf{X} \in \mathbb{R}^{n\times k}$. The projection of $\mathbf{X}$ onto a line by any vector $\mathbf{u} \in \mathbb{R}^{k\times 1}$ is the scores vector $\mathbf{Xu}$. If $\mathbf{X}$ is mean centred, the variance of the projection along the line is $\mathbf{u}^T\mathbf{S}_X\mathbf{u}$ where $\mathbf{S}_X$ is the variance-covariance of $\mathbf{X}$. Constraining the projecting vector $\mathbf{u}$ to unit length so that it only represents the projecting direction forms the method as an optimization problem. This projection is illustrated as Figure 2.1.
Maximize

$$f(\mathbf{u}) = \mathbf{u}^T\mathbf{S}_X\mathbf{u} \tag{2.4}$$

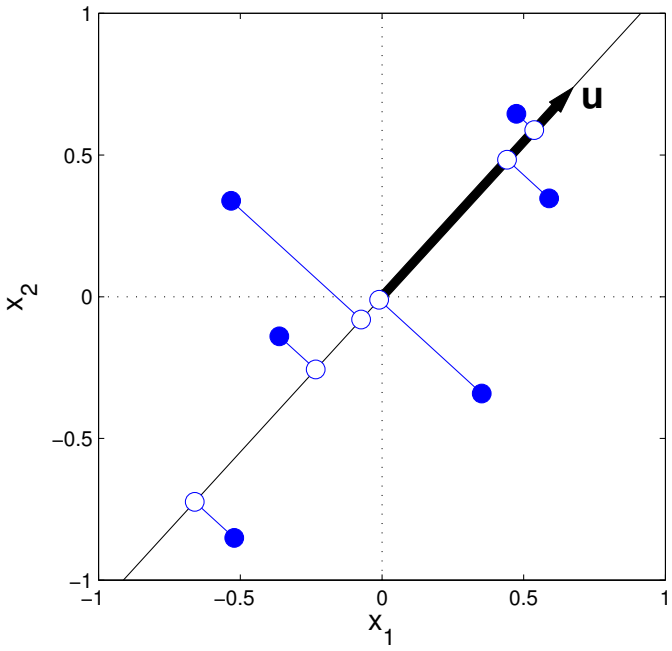

Figure 2.1: Orthogonal Projection onto a Line

With respect to $\mathbf{u}$, subject to

$$\mathbf{u}^T\mathbf{u} = 1 \tag{2.5}$$

As there is a single constraint, the multiplier is a single scalar. So the Lagrangian function with a single multiplier is

$$L(\mathbf{u}, \lambda) = \mathbf{u}^T\mathbf{S}_X\mathbf{u} - \lambda(\mathbf{u}^T\mathbf{u} - 1) \tag{2.6}$$

Setting

$$\frac{\partial L(\mathbf{u}, \lambda)}{\partial \mathbf{u}} = 0 \tag{2.7}$$

gives

$$\mathbf{S}_X\mathbf{u} = \lambda\mathbf{u} \tag{2.8}$$

or

$$\mathbf{u}^T\mathbf{S}_X\mathbf{u} = \lambda \tag{2.9}$$

The objective $f(\mathbf{u})$ is a scalar function with a vector argument $\mathbf{u} \in \mathbb{R}^{n\times 1}$. So the partial derivative in equation (2.7) is a vector differential. This final form as equation

(2.9) shows that the maximum variance along the projection line must be equal to the largest eigenvalue of $\mathbf{S}_X$, so the optimal value of $\mathbf{u}$ for maximizing the projection variance must be the corresponding eigenvector.

Another important consequence of the orthogonal projection is that the projection residuals are also minimized. So this projection simultaneously optimizes the sums of squares along the line and the sums of squares between the data points and the line. This shows how maximizing the variance of projections, Lagrange's functions, eigenvalues and eigenvectors are all related within PCA.

The links between eigenvalues and singular value decomposition is well known, see for example Gentle[38]. An efficient way of computing the scores vector $\mathbf{u}$ from equation (2.9) is by a SVD on the variance-covariance as $\mathcal{U}\mathcal{S}\mathcal{V}^T = \mathbf{S}_X$ so that the scores $\mathbf{u}$ associated with the maximum eigenvalue and singular value is the first column of $\mathcal{U}$. Further, the maximum number of principal components is equal to k, the number of columns in $\mathbf{X}$ and the number of non-zero principal components is equal to the rank of $\mathbf{S}_X$.

2.2 Basic PLS

By definition, PLS seeks to maximize the covariance between the regressor matrix and response. As this covariance is the cross-covariance matrix, a projection onto a lower dimension subspace is implied so that the scalar objective function of the cross-covariance that is be maximized. So the primary difference between PCA and PLS is that PCA finds vectors to project the $\mathbf{X}$ matrix, while PLS finds vectors to project the cross-covariance $\mathbf{S}_{XY}$ between regressor matrix $\mathbf{X}$ and the response matrix $\mathbf{Y}$. The general case for multivariate responses is presented here and is equally applicable to univariate responses.

In order to make the cross-covariance function scalar, the PLS algorithm starts by using weights vectors to project the (centred) regressor and response matrices into scores vectors, $\mathbf{t} = \mathbf{X}\mathbf{w}^T$ and $\mathbf{u} = \mathbf{Y}\mathbf{c}$ where the weights vectors $\mathbf{w}$ and $\mathbf{c}$ are unit

length. The PLS algorithm can now be stated as an optimization problem, just as PCA in equation (2.4). This approach to PLS is from Phatak and de Jong[90].

That is, maximize

$$f(\mathbf{w}, \mathbf{c}) = \mathbf{t}^T\mathbf{u} \tag{2.10}$$

with respect to $\mathbf{w}$ and $\mathbf{c}$, where

$$\mathbf{t} = \mathbf{X}\mathbf{w}^T \tag{2.11}$$

$$\mathbf{u} = \mathbf{Y}\mathbf{c} \tag{2.12}$$

subject to

$$\mathbf{w}^T\mathbf{w} = 1 \tag{2.13}$$

$$\mathbf{c}^T\mathbf{c} = 1 \tag{2.14}$$

Since $\mathbf{t}^T\mathbf{u} = \mathbf{w}^T\mathbf{X}^T\mathbf{Y}\mathbf{c}$, the Langangian function is

$$L(\mathbf{w}, \mathbf{c}) = \mathbf{w}^T\mathbf{X}^T\mathbf{Y}\mathbf{c} + \lambda_1(1 - \mathbf{w}^T\mathbf{w}) + \lambda_2(1 - \mathbf{c}^T\mathbf{c}) \tag{2.15}$$

The solution to the maximum cross-covariance function problem is the point where all four differentials are zero.

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{X}^T\mathbf{Y}\mathbf{c} - 2\lambda_1\mathbf{w} = 0 \tag{2.16}$$

$$\frac{\partial L}{\partial \mathbf{c}} = \mathbf{Y}^T\mathbf{X}\mathbf{w} - 2\lambda_2\mathbf{c} = 0 \tag{2.17}$$

$$\frac{\partial L}{\partial \lambda_1} = 1 - \mathbf{w}^T\mathbf{w} = 0 \tag{2.18}$$

$$\frac{\partial L}{\partial \lambda_2} = 1 - \mathbf{c}^T\mathbf{c} = 0 \tag{2.19}$$

The partial differentials equations (2.16) and (2.17) are vector differentials. Comparing these equations shows that apart from the transpose, they are the same and so the multipliers are equal $\lambda_1 = \lambda_2$. This turns the solution into a SVD form.

$$\mathbf{c}\mathbf{Y}^T\mathbf{X}\mathbf{w} = \mathbf{w}\mathbf{X}^T\mathbf{Y}\mathbf{c} = s \tag{2.20}$$

where s is singular value associated with $\mathbf{Y}^T\mathbf{X}$ or equivalently $\mathbf{X}^T\mathbf{Y}$. Since the optimization function from equation (2.10) to be maximized is $f(\mathbf{w}, \mathbf{c}) = \mathbf{t}^T\mathbf{u} = \mathbf{w}^T\mathbf{X}^T\mathbf{Y}\mathbf{c}$,

the unique solution for s is the maximal singular value of $\mathbf{X}^T\mathbf{Y}$ or $\mathbf{Y}^T\mathbf{X}$. Further, the $\mathbf{w}, \mathbf{c}$ vectors that are the solution to the optimization are the left and right vectors of the SVD form.

So to calculate the weights vectors, compute the SVD of the cross-covariance matrix

$$\mathcal{U}\mathcal{S}\mathcal{V}^T = SVD(\mathbf{X}^T\mathbf{Y}) \tag{2.21}$$

The weights vectors $\mathbf{w}$ and $\mathbf{c}$ are then the first columns of $\mathcal{U}$ and $\mathcal{V}$. The other optimization constraints of unit length on both weights vectors $\mathbf{w}$ and $\mathbf{c}$ are also consistent with this SVD solution as the columns of $\mathcal{U}$ and $\mathcal{V}$ from a singular value decomposition are all orthonormal by definition.

For PCA, the variance-covariance matrix decomposed in equation (2.9) is square, so only a single score vector is required. By comparison, the cross-covariance decomposed for the general form of PLS for multivariate responses as equation (2.20) is generally not square, so two weights vectors and consequently two scores vectors are required. Many of the properties of the scores and loadings vectors in both PCA and PLS are related to the properties of the SVD that is behind both methods.

Deflation is a common feature to all the PLS algorithms. Here, deflation means that the variance explained for each latent variable is subtracted from the $\mathbf{X}$ regressor matrix, $\mathbf{Y}$ responses or their cross-covariances. After A latent variable iterations and for the general case of PLS2 multivariate responses, the fitted values of $\mathbf{X}$ and $\mathbf{Y}$ are

$$\hat{\mathbf{X}}_A = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T \tag{2.22}$$

$$\hat{\mathbf{Y}}_A = \sum_{a=1}^{A} \mathbf{u}_a \mathbf{q}_a^T \tag{2.23}$$

where the $\mathbf{X}$ and $\mathbf{Y}$ loadings $\mathbf{p}$ and $\mathbf{q}$ are row vectors defined by

$$\mathbf{p}_a = \frac{\mathbf{t}_a^T \mathbf{X}_a}{\mathbf{t}_a^T \mathbf{t}_a} \tag{2.24}$$

$$\mathbf{q}_a = \frac{\mathbf{t}_a^T \mathbf{Y}_a}{\mathbf{t}_a^T \mathbf{t}_a} \tag{2.25}$$

so the deflation for this Basic PLS is

$$\mathbf{X}_{a+1} = \mathbf{X}_a - \hat{\mathbf{X}}_a = \mathbf{X}_a - \mathbf{t}_a\mathbf{p}_a^T \tag{2.26}$$

$$\mathbf{Y}_{a+1} = \mathbf{Y}_a - \hat{\mathbf{Y}}_a = \mathbf{Y}_a - \mathbf{t}_a\mathbf{q}_a^T \tag{2.27}$$

The same deflation scheme also applies to PLS1 univariate responses, except that the $\mathbf{y}$ loadings $\mathbf{q}$ is a scalar and the $\mathbf{y}$ weights $\mathbf{u}$ in equation(2.23) is replaced by the $\mathbf{X}$ scores $\mathbf{t}$, that is $\hat{\mathbf{Y}}_a = \mathbf{t}_a\mathbf{q}_a^T$. After deflation, the calculation starts again using the deflated $\mathbf{X}$ and $\mathbf{Y}$ matrices.

For the the PLS algorithm, deflating both X and Y matrices is not required[50], since

$$\mathbf{X}_{i+1}^T\mathbf{Y}_{i+1} = \mathbf{X}_i^T\mathbf{Y}_{i+1} = \mathbf{X}_{i+1}^T\mathbf{Y}_i \tag{2.28}$$

Further, since $\mathbf{X}$ can be updated as a function of the scores $\mathbf{t}$, the cross-covariance matrix deflation can also be expressed as a function of $\mathbf{t}$. Hence, alternative PLS algorithms by deflating X, Y or the cross-covariance would give identical results[20].

2.2.1 Basic PLS and PLS1 NIPALS

The standard form of Wold's[111] PLS1 NIPALS algorithm, as shown in Martens and Næs[80] for example is

$$\mathbf{w}_1 = \mathbf{X}_0^T\mathbf{y}/||\mathbf{X}_0^T\mathbf{y}|| \tag{2.29}$$

$$\mathbf{t}_1 = \mathbf{X}_0\mathbf{w}_1 \tag{2.30}$$

$$\mathbf{p}_1 = \mathbf{X}_0^T\mathbf{t}_1/\mathbf{t}_1^T\mathbf{t}_1 \tag{2.31}$$

$$\mathbf{q}_1 = \mathbf{y}_0^T\mathbf{t}_1/\mathbf{t}_1^T\mathbf{t}_1 \tag{2.32}$$

$$\mathbf{X}_1 = \mathbf{X}_0 - \mathbf{t}_1\mathbf{p}_1^T \tag{2.33}$$

$$\mathbf{y}_1 = \mathbf{y}_0 - \mathbf{t}_1\mathbf{q}_1 \tag{2.34}$$

then repeated for the next latent variable.

Since $\mathcal{U}\mathcal{S}\mathcal{V}^T = \mathbf{X}^T\mathbf{y}$, the maximum singular value of the cross-covariance decomposition is s_1, the 1st diagonal element in $\mathcal{S}$, then

$$||\mathbf{X}^T\mathbf{y}|| = s_1 \tag{2.35}$$

So

$$(\mathbf{X}^T\mathbf{Y})\mathcal{V} = \mathcal{U}\mathcal{S}\mathcal{V}^T\mathcal{V} = \mathcal{U}\mathcal{S} \tag{2.36}$$

since $\mathcal{V}$ is orthogonal from the SVD. For PLS1, $\mathbf{X}^T\mathbf{Y}$ is a vector, so $\mathcal{V}$ is scalar and always ± 1, so the $\mathbf{Y}$ weights $\mathbf{c}$ in the original Lagrangian equation (2.15) are not required for PLS1. Given the first left hand vector of $\mathcal{U}$ is $\boldsymbol{\upsilon}_1$ then

$$\boldsymbol{\upsilon}_1 = \mathbf{X}^T\mathbf{y}/\mathcal{S}_1 \tag{2.37}$$

$$= \mathbf{X}^T\mathbf{y}/||\mathbf{X}^T\mathbf{y}|| = \mathbf{w} \tag{2.38}$$

which is the way the weights are calculated in the first step of the NIPALS algorithm. Since each iteration of NIPALS starts with the calculation of the weights vector $\mathbf{w}$ then calculates all the other scores and loadings in sequence from the weights, the PLS1 NIPALS algorithm must be equivalent to this Basic form.

2.2.2 Basic PLS and PLS2 NIPALS

The original PLS2 NIPALS algorithm by Wold, Martens and Wold[114] calculates the weights, scores and loadings vectors iteratively. The scores vector $\mathbf{u}$ is initialised to some arbitrary value, usually the first column of the response matrix.

Then

$$\mathbf{w} = \mathbf{X}^T\mathbf{u}/||\mathbf{X}^T\mathbf{u}|| \tag{2.39}$$

$$\mathbf{t} = \mathbf{X}\mathbf{w} \tag{2.40}$$

$$\mathbf{q} = \mathbf{t}^T\mathbf{Y}/||(\mathbf{t}^T\mathbf{Y}) \tag{2.41}$$

$$\mathbf{u} = \mathbf{Y}\mathbf{q}^T \tag{2.42}$$

Equations (2.39) to (2.42) are iterated until convergence as tested by the weights $\mathbf{w}$ is achieved. At the a^{th} latent variable, the scalar inner regression coefficient c_a is then calculated followed by the deflation steps as

$$c_a = \mathbf{u}^T\mathbf{t}/\mathbf{t}^T\mathbf{t} \tag{2.43}$$

$$\mathbf{X}_{a+1} = \mathbf{X}_a - \mathbf{t}_a\mathbf{p}_a^T \tag{2.44}$$

$$\mathbf{Y}_{a+1} = \mathbf{Y}_a - c_a\mathbf{t}_a\mathbf{q} \tag{2.45}$$

Manne[74] noted that when this algorithm is applied to PLS1 univariate responses, $\mathbf{q}$ in equation (2.41) is one, $\mathbf{u}$ becomes $\mathbf{y}$ and convergence is always achieved in one step. In practice, convergence is usually obtained with PLS2 multivariate responses but may be slow. Lyttkens[72] gives a general proof of convergence for NIPALS algorithms. Even if this is proved algebraically, Hensler[48] makes some valuable comments on how convergence can be made robust numerically.

On convergence, the weights $\mathbf{w}$ are equivalent to left singular vector from the SVD of $\mathbf{X}^T\mathbf{Y}$, which is proved by Manne[74]. The $\mathbf{Y}$ loadings $\mathbf{q}$ on convergence are related to the $\mathbf{Y}$ weights $\mathbf{c}$ by

$$\mathbf{q} = s_1\mathbf{c}^T/(\mathbf{t}^T\mathbf{t}) \tag{2.46}$$

where s_1 is the largest singular value of $\mathbf{X}^T\mathbf{Y}$. This is proved in Di Ruscio[20]. As the weights and the loadings are the same, this shows that this Basic PLS and PLS2 NIPALS are equivalent.

2.2.3 Basic PLS and PLS NIPALS Kernel Methods

These algorithms were originally developed by Lindgren, Geladi and Wold[69] just to improve the computational speed of the NIPALS algorithms for datasets with large numbers of observations. The improvement in speed comes from avoiding computing the large scores matrices $\mathbf{T} \in \mathbb{R}^{n\times A}$ by using the smaller "kernel" matrix $\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X} \in \mathbb{R}^{k\times k}$ combined with $\mathbf{X}^T\mathbf{X}$, $\mathbf{X}^T\mathbf{Y}$ and $\mathbf{Y}^T\mathbf{Y}$. These are the algorithms that are often used for calculation when NIPALS algorithms are specifically required. The original paper by Lindgren, Geladi and Wold[69] proves that these kernel algorithms give identical results to PLS1 NIPALS and PLS2 NIPALS, so these algorithms not discussed specifically any further here.

2.2.4 Basic PLS and SIMPLS

Unlike NIPALS, de Jong[16] derived SIMPLS specifically to maximize a covariance measure. Each iteration of the SIMPLS algorithm starts with the SVD decomposition

of the (deflated) covariance matrix. In outline, the algorithm starts with

$$\mathbf{S} = \mathbf{X}^T\mathbf{Y} \tag{2.47}$$

$$\mathcal{U}\mathcal{S}\mathcal{V}^T = \mathbf{S} \tag{2.48}$$

$$\mathbf{r} = \boldsymbol{\upsilon}_1 \tag{2.49}$$

$$\mathbf{t} = \mathbf{X}\mathbf{r}/||\mathbf{X}\mathbf{r}|| \tag{2.50}$$

$$\mathbf{p} = \mathbf{X}^T\mathbf{t} \tag{2.51}$$

$$\mathbf{q} = \mathbf{t}^T\mathbf{Y} \tag{2.52}$$

where equation(2.48) represents the SVD decomposition of the covariance matrix $\mathbf{S}$ and $\boldsymbol{\upsilon}_1 \in \mathbb{R}^{k\times 1}$ in equation(2.49) is the left singular vector of $\mathcal{U}$, not the response first scores vector $\mathbf{u}_1 \in \mathbb{R}^{n\times 1}$. Up to this point, this is the same method as Basic PLS expect for the normalisation. The PLS method has been presented here as constrained optimization. In NIPALS and Basic PLS the constraint is that the regressor weights $\mathbf{w}$ are scaled to unit length. SIMPLS does not require weight vectors directly, so it is the regressor scores vectors $\mathbf{t}$ that is scaled to unit length. Since the regressor and loadings $\mathbf{p}$ are calculated from the scores, their products as fitted values $\hat{\mathbf{X}}_a = \mathbf{t}\mathbf{p}^T$ are independent of whether it is the weights of scores that have been normalised.

The principal difference between Basic PLS or NIPALS and SIMPLS is that the deflation for each latent variable in SIMPLS is performed on the covariance matrix $S_{\mathbf{XY}}$, not the regressor and response matrices $\mathbf{X}$ and $\mathbf{Y}$. To update the covariance, we need an orthonormal basis of all the $\mathbf{X}$-block loadings $\mathbf{P}$, say $\mathbf{V}$. For the second and subsequent latent variables at say a latent variables, initialise $\mathbf{v}_a$ as $\mathbf{u}_1$,

$$\mathbf{v}_a = \mathbf{v}_a - \mathbf{V}_{1..a}(\mathbf{V}_{1..a}^T\mathbf{p}) \tag{2.53}$$

$$\mathbf{u}_a = \mathbf{u}_a - \mathbf{T}_{1..a}(\mathbf{T}_{1..a}^T\mathbf{u}) \tag{2.54}$$

This Gram-Schmidt procedure forces the loadings $\mathbf{v}$ to be orthogonal to all the previous loadings $\mathbf{V}_{1..a}$ and the scores $\mathbf{u}$ to be orthogonal to all the previous scores $\mathbf{T}_{1..a}$.

The next step is to deflate the covariance by first removing projections along the

current basis vector

$$\mathbf{v}_a = \mathbf{v}_a / ||\mathbf{v}_a|| \tag{2.55}$$

$$\mathbf{S_{XY}} = \mathbf{S_{XY}} - \mathbf{v}_a(\mathbf{v}_a^T \mathbf{S_{XY}}) \tag{2.56}$$

The next latent variable is calculated in turn by calculating the SVD as in equation (2.48).

SIMPLS does not require the calculation of the weights $\mathbf{w}_i$. For comparison with other PLS algorithms or diagnostic purposes, these can be calculated by

$$\mathbf{w}_i = \mathbf{r}_i / ||\mathbf{X}\mathbf{r}_i|| \tag{2.57}$$

so $\mathbf{w}_i \in \mathbb{R}^{k \times 1}$.

But there is a difference between SIMPLS and NIPALS due to the way the orthogonality constraint is applied. Phatak and de Jong[90] showed that for NIPALS, the deflated $\mathbf{X}$ matrix after A latent variables have been extracted is

$$\hat{\mathbf{X}}_A = [\mathbf{I}_A - \mathbf{T}_A \mathbf{T}^T]\mathbf{X}_0 \tag{2.58}$$

so the projection of the columns of $\mathbf{X}$ onto the space orthogonal to the scores already extracted. For SIMPLS they showed that

$$\hat{\mathbf{X}}_A = [\mathbf{I}_A - \mathbf{T}_A (\mathbf{T}_A^T \mathbf{T})^{-1} \mathbf{T}^T]\mathbf{X}_0 \tag{2.59}$$

In the original SIMPLS paper by de Jong[16], there is a proof that there is no difference for PLS1 univariate responses. For PLS2 multivariate responses, only the scores and loadings vectors from the first latent variable will be identical for NIPALS and SIMPLS, all the other vectors will be slightly different. Phatak and de Jong[90] state "Only in pathological cases will the difference be of any practical consequence."

As NIPALS and SIMPLS use different sets of vectors internally, the simplest direct comparison is through the model fits in the regressor and response spaces. As an example, these fits are shown for the OliveOil PLS2 dataset for scaled regressors and responses as Figure 2.2, where the differences in the fit are minor.

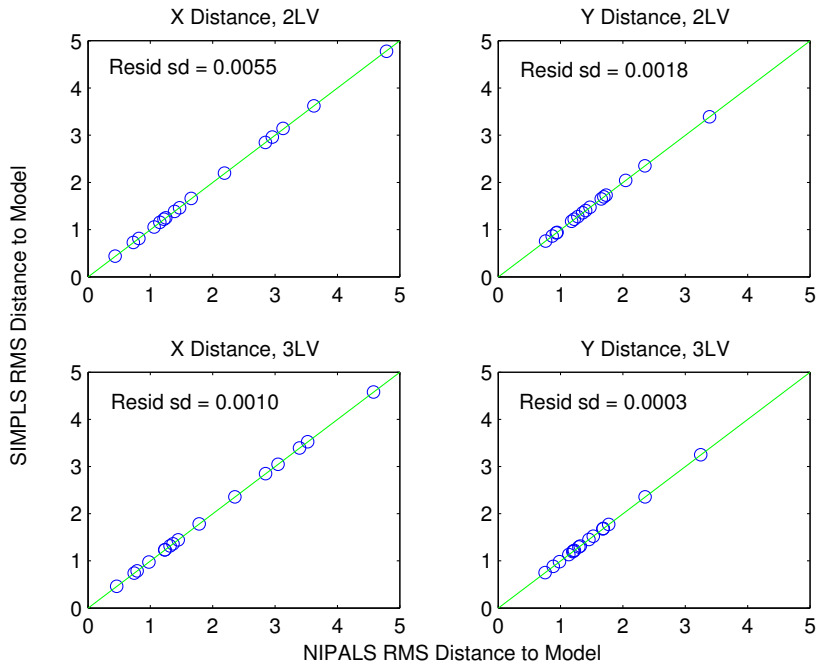

Figure 2.2: Example Comparing NIPALS and SIMPLS Model Fits

2.2.5 Basic PLS and BIDIAG

The 1987 paper by Manne[74] also introduced a diagonalization method for univariate PLS1, based on a numerical method originally from Golub and Kahan [40]. This version of bidiagonalization is an iterative algorithm that uses a Lancos process and was called BIDIAG2 by Paige and Saunders[86]. Originally proposed as a fast calculation method for large datasets. Also, as small eigenvalues due to noise may give large contributions to the regression coefficients in all forms of latent variable regression, the diagonalization approach uses a limiting threshold so that these small eigenvalues are excluded. Consequently, it should also improve the stability of PLS models.

The BIDIAG algorithm decomposes the regressor matrix according to

$$\mathbf{X} = \mathbf{U} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T \tag{2.60}$$

where $\mathbf{X} \in \mathbb{R}^{n \times k}$. $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_n)$ and $\mathbf{V} = (\mathbf{v}_1, ..., \mathbf{v}_n)$ are square orthogonal matrices and $\mathbf{B} \in \mathbb{R}^{(k+1) \times k}$ is a bidiagonal matrix. Since $\mathbf{V}$ is orthogonal then

$$\mathbf{XV} = \mathbf{U} \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \tag{2.61}$$

For PLS, the BIDIAG algorithm for A latent variables is

$$\mathbf{v}_1 = \mathbf{X}_0^T\mathbf{y}/||\mathbf{X}^T\mathbf{y}|| \tag{2.62}$$

$$\alpha_1\mathbf{u}_1 = \mathbf{X}_0^T\mathbf{v}_1 \tag{2.63}$$

for $a = 2$ to A

$$\gamma_{a-1}\mathbf{v}_a = \mathbf{X}_0^T\mathbf{u}_{a-1} - \alpha_{a-1}\mathbf{v}_{a-1} \tag{2.64}$$

$$\alpha_a\mathbf{u}_a = \mathbf{X}_0\mathbf{v}_a - \gamma_{a-1}\mathbf{u}_{a-1} \tag{2.65}$$

All of the coefficients α and γ are calculated so that all of the vectors $\mathbf{u}$ and $\mathbf{v}$ are unit length. For A latent variables, the bidiagonal matrix $\mathbf{B}$ is

$$\mathbf{B} = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ & \alpha_2 & \gamma_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{A-1} & \gamma_{A-1} \\ & & & & \alpha_A \end{bmatrix} \tag{2.66}$$

From equations (2.62) and (2.63) at the start of the PLS BIDIAG algorithm, it is clear that $\mathbf{v}_1$ and $\mathbf{u}_1$ are the first weights and scores vectors $\mathbf{w}_1$ and $\mathbf{t}_1$ from NIPALS. By considering the span of the weights and scores as Krylov sequences, Eldén[24] proved equality between $\mathbf{u}$ and $\mathbf{t}$ and between $\mathbf{v}$ and $\mathbf{w}$ for all latent variables. So from equation(2.60) and using the weights and scores notation for clarity, after A latent variables the regressor matrix is approximated by

$$\mathbf{X} \approx \mathbf{T}_A\mathbf{B}_A\mathbf{W}_A^T \tag{2.67}$$

As an example comparing NIPALS and BIDIAG pls, regressor and response space distances for the Wine Aroma dataset with scaled variates is shown as Figure 2.3. This shows that there is no difference in the response fit and so response residuals but there is a difference in the regressor fitted values.

The paper by Pell, Ramos and Manne[88] in 2007 concerning the validity of the NIPALS compared to the BIDIAG approach caused some controversy in the literature. These issues were resolved by Bro and Eldén[9] who give a good explanation of this,

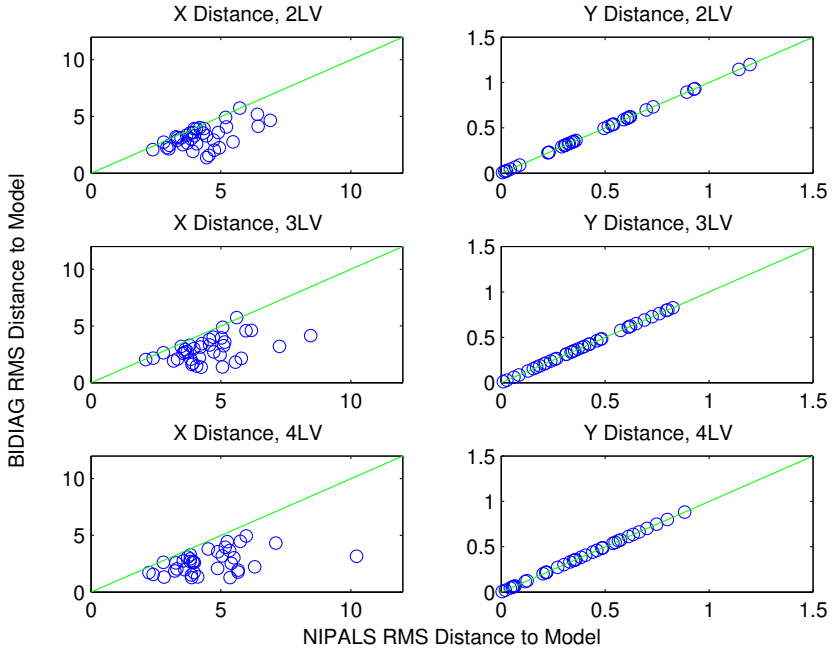

Figure 2.3: Example Comparing NIPALS and BIDIAG Model Fits

in that both approaches are equally valid but represent different projections. This difference in projections is apparent in the regressor distance plots shown as Figure 2.3. As this explanation and contributions by others [25],[26],[32],[120] revealed a lot about the nature of PLS methods these details are examined next.

PLS NIPALS Regressor Residuals

In the standard form of PLS NIPALS, the loadings vector $\mathbf{p}_a$ at a latent variables are calculated from the scores $\mathbf{t}_a$,

$$\mathbf{p}_a = \mathbf{X}_{a-1}^T \mathbf{t}_a \tag{2.68}$$

or in a cumulative matrix form for the 1^{st} to the A^{th} latent variable

$$\mathbf{P}_A = \mathbf{X}_0^T \mathbf{T}_A \tag{2.69}$$

$$\mathbf{P}_A^T = \mathbf{T}_A^T \mathbf{X}_0 \tag{2.70}$$

where $\mathbf{P}_A = [\mathbf{p}_1, ..., \mathbf{p}_A]$ and $\mathbf{T}_A = [\mathbf{t}_1, ..., \mathbf{t}_A]$ are the matrices formed by combining the column vectors and $\mathbf{X}_0$ is the original undeflated regressor matrix.

With the notation $\mathbf{X}_a$ for the regressor matrix and $\hat{\mathbf{X}}_a$ for the fitted matrix after a latent variables, the residuals used as the regressor matrix in the next iteration are

$$\mathbf{X}_{a+1} = \hat{\mathbf{X}}_a - \mathbf{t}_a \mathbf{p}_a^T \tag{2.71}$$

or in a cumulative matrix form for A latent variable is

$$\hat{\mathbf{X}}_A = \mathbf{X}_0 - \mathbf{T}_A \mathbf{P}_A^T \tag{2.72}$$

So

$$\mathbf{X}_0 - \hat{\mathbf{X}}_A = \mathbf{T}_A \mathbf{T}_A^T \mathbf{X}_0 \tag{2.73}$$

From equation (2.73) it is apparent that the the regressor residuals are in the column space of the scores. Further proof of this is in Höskuldsson[50], where it is also proved that the row space of the regressor residuals are also orthogonal to the loadings $\mathbf{p}$. Equation (2.71) shows that at each iteration, the variation in $\mathbf{X}_a$ in the direction of the scores vector $\mathbf{t}_a$ is removed entirely during the deflation. So an important consequence is that for PLS NIPALS, regressor space outliers can be detected through their scores and response fit outliers from their residuals, because the scores and residuals are orthogonal and so independent.

PLS BIDIAG Regressor Residuals

So after A latent variable, the regressor residuals corresponding to equation(2.73) are

$$\hat{\mathbf{X}}_A = \mathbf{X}_0 - \mathbf{T}_A \mathbf{B}_A \mathbf{W}_A^T \tag{2.74}$$

From equation (2.61) $\mathbf{XW} = \mathbf{TB}$ then

$$\hat{\mathbf{X}}_A = \mathbf{X}_0 - \mathbf{X}_0 \mathbf{W}_A \mathbf{W}_A^T \tag{2.75}$$

$$\hat{\mathbf{X}}_A = (\mathbf{I}_A - \mathbf{W}_A \mathbf{W}_A^T)\mathbf{X}_0 \tag{2.76}$$

Comparing equations (2.71) and (2.76) shows that the regressor residuals for PLS BIDIAG are in the column space of the weights $\mathbf{W}$ and orthogonal to them, but are not orthogonal to the scores $\mathbf{T}$ and loadings $\mathbf{P}$. This is opposite to the properties of the regressor residuals for the PLS1 NIPALS derived from equation (2.73). So the important model diagnostics are weaker for PLS BIDIAG, because the the model space

part of the model is not orthogonal to the residuals[120]. In a study of PLS regressor residuals in the context of a process control study, Ergon, Halstensen and Esbensen[26] continue to support PLS BIDIAG even though there is little practical difference between the analyses of their datasets. Wise[110] showed that for PLS BIDIAG the correlation was always between the regressor errors and last scores vector and that this level of correlation is variable but can be large. A further conclusion was that the regressor residuals from PLS BIDIAG will always be greater than those from PLS NIPALS. Consequently, the BIDIAG version of PLS will not be considered further here.

2.3 The PLS Regression Coefficients

It is quite possible to fit PLS models and examine the characteristics of the fit, then go on to use this for prediction without calculating any coefficients for the regression. But these coefficients are important for interpreting PLS models and can simplify the statistical aspects of model fitting and diagnostics. This section shows how coefficients β can be calculated from the scores and loadings vectors which can then be used for equivalent prediction calculations. As might be anticipated for PLS, these coefficients can be defined and derived in a number of different ways.

In this section the "hat" on $\hat{\beta}$ implies that this is in some way an estimated best fit value. Similarly, $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ are the fitted $\mathbf{X}$ and $\mathbf{Y}$ residuals after A latent variables have been extracted. The $_A$ subscript is implied throughout but is omitted for clarity. The other scores, weights and loadings vectors are considered as internal model parameters and so are not best fit in any way. The coefficients and their derivatives are important calculations, used in degrees of freedom calculations, estimating the effects of variables and prediction intervals.

2.3.1 Martens and Naes Original Derivation

The most accessible source for the coefficients for PLS1 univariate regression is Martens and Næs[80], where it is stated without proof

$$\hat{\beta} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{q} \tag{2.77}$$

Helland[47] said that this could be proved in a number of ways and provides this as an example

$$\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^T \tag{2.78}$$

$$\hat{\mathbf{X}}\mathbf{W} = \mathbf{T}\mathbf{P}^T\mathbf{W} \tag{2.79}$$

$$\mathbf{T} = \hat{\mathbf{X}}\mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1} \tag{2.80}$$

$$\hat{\mathbf{Y}} = \mathbf{T}\mathbf{q} \tag{2.81}$$

$$= \hat{\mathbf{X}}\mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{q} \tag{2.82}$$

As $\hat{\mathbf{Y}} = \hat{\mathbf{X}}\hat{\beta}$ then

$$\hat{\beta} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{q} \tag{2.83}$$

Manne[74] proved that $\mathbf{P}^T\mathbf{W}$ is lower triangular with ones on the diagonal. As such it is easily invertible and Manne gives an efficient formula for this. Equation (2.81) is specific to PLS1 univariate regressions. The equivalent equation in matrix form for PLS2 multivariate responses from equation (2.45) is

$$\hat{\mathbf{Y}} = \mathbf{T}\mathbf{C}\mathbf{Q} \tag{2.84}$$

where $\mathbf{T} \in \mathbb{R}^{n\times A}$ is the matrix of column vector scores, $\mathbf{C} \in \mathbb{R}^{A\times A}$ is a of diagonal matrix of the inner regression coefficients from equation (2.43) and $\mathbf{Q} \in \mathbb{R}^{A\times m}$ is the matrix of row vector loadings. So the equivalent form for the PLS2 regression coefficients is

$$\hat{\beta} = \mathbf{W}(\mathbf{P}^T\mathbf{W})^{-1}\mathbf{C}\mathbf{Q} \tag{2.85}$$

2.3.2 Pell, Ramos and Manne Pseudoinverse Derivation

This alternative derivation from Pell, Ramos and Manne[88] is interesting as it shows a pseudoinverse form direct from scores and loadings so is applicable to NIPALS and

SIMPLS algorithms with PLS1 univariate responses. While the construction of pseudoinverses normally uses SVD decompositions, this alternative way is shown below using results from Barnet[6]

Assume $\mathbf{A}$ does not have full rank, i.e. $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $rank(\mathbf{A}) = r < min(n, m)$. There does always exists two matrices $\mathbf{C} \in \mathbb{R}^{n \times r}$ and $\mathbf{D} \in \mathbb{R}^{r \times m}$ of rank r such that $\mathbf{A} = \mathbf{CD}$. Then

$$\mathbf{A}^+ = \mathbf{D}^+\mathbf{C}^+ \tag{2.86}$$

$$\mathbf{A}^+ = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \tag{2.87}$$

For PLS, $\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^T$ which leads directly to the form used in Pell, Ramos and Manne

$$\hat{\mathbf{X}}^+ = \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T \tag{2.88}$$

Since $\hat{\mathbf{Y}} = \mathbf{Tq}$ then

$$\hat{\beta} = \hat{\mathbf{X}}^+\hat{\mathbf{Y}} \tag{2.89}$$

$$= \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Tq} \tag{2.90}$$

$$= \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{q} \tag{2.91}$$

which is not equivalent to equation (2.83). While this coefficient derivation using pseudoinverses is from the Pell,Ramos and Manne paper[88] that mainly concerned the BIDIAG algorithm, this derivation is not specific to any particular algorithm. Figure 2.4 compares the coefficients for the WineAroma example dataset, which shows that the differences are not negligible. This issue is discussed by Ergon et al[25],[26] which showed that this difference in the coefficients calculation is another aspect of the model space issue.

2.4 Conclusions on the PLS Algorithms

The conventional presentation of PLS presents the development as the evolution of the various algorithms. This requires fairly complex proofs to show equivalence or to reveal where the algorithms differ. Starting from the underlying singular value decomposition makes the relation between the methods clear while avoiding the necessity for

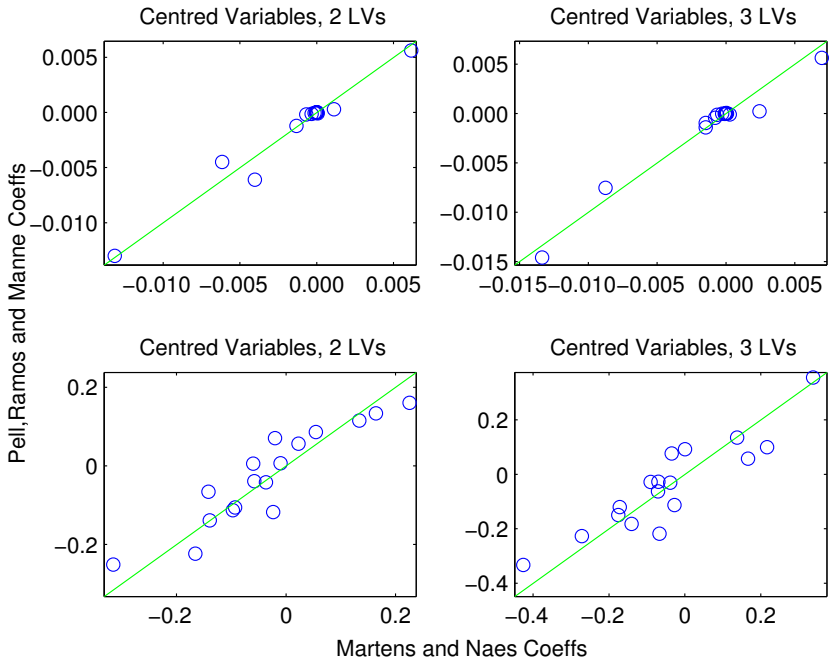

Figure 2.4: Example Comparing Alternative Coefficient Calculations

most of the proofs.

This controversy concerning the regressor space residuals started by Pell, Ramos and Manne[88] has made the relations between the PLS vectors and their spaces a lot clearer. The overall conclusion from Wold et al [120] is that the NIPALS and BIDIAG approaches represent different models so slightly different results should be anticipated. While it is now clear that NIPALS should support stronger diagnostic tests than BIDIAG, the issues raised concerning the appropriate column space of the regression coefficients by Ergon, Halstensen and Esbensen[26] remain unresolved. Apart from a short comment by Ergon[25] the PLS2 multivariate case has not been considered in these discussions. How these properties for the SIMPLS algorithm relate to NIPALS and BIDIAG has also not been discussed in the literature. The regressor residual space has been proved in equation 2.73. Similarly, rearranging equation (2.59) for SIMPLS gives the residuals as

$$\hat{\mathbf{X}}_A = [\mathbf{I}_A - \mathbf{T}_A(\mathbf{T}_A^T\mathbf{T})^{-1}\mathbf{T}^T]\mathbf{X}_0 \tag{2.92}$$

$$\mathbf{X}_0 - \hat{\mathbf{X}}_A = \mathbf{T}_A(\mathbf{T}_A^T\mathbf{T})^{-1}\mathbf{T}^T \tag{2.93}$$

which shows that the NIPALS and SIMPLS regressor fitted values residuals projections and the residuals are within the same column space. The equivalence of NIPALS and SIMPLS for univariate responses yet the differences although generally minor for multivariate responses was clear from the original paper by de Jong[16], so any assumptions about PLS1 univariate models spaces may not be valid for PLS2 multivariate case. To date, nothing further specific to this controversy has been published by the main protagonists since Ergon, Halstensen and Esbensen[26] in 2011. But it may well be that our understanding of PLS spaces is not yet complete.

For the purposes of the rest of this work, the standard NIPALS algorithm has been used as it's properties are at least well established. PLS coefficients and their derivatives appear later in the discussion on latent variable selection by information criteria. In view of the issues raised here, methods for calculating information criteria are presented in Chapter 6 that avoid the use of coefficients.

Chapter 3

The Example Datasets

Publications of PLS developments generally contain examples to illustrate the technique. In this work six datasets have been used to compare and develop PLS methods. Some of these datasets have already been used in the previous chapter on PLS algorithms, In this chapter the example datasets are examined in detail and their structure and characteristics discussed. Most of these are from the chemometrics literature, but have been selected so that their characteristics form a factorial plan. These features are "portrait", that is more trials than regressor variables and with correlated or collinear regressor variables, but full rank. "Landscape" as more regressor variables than trials so not full rank. These two landscape datasets are both spectroscopy data, which is a typical application for PLS. This data is analysed in the "calibration" sense where the absorptions at each of many wavelengths are the regressors and the chemical assays are the responses. A characteristic of this type of data is the strong correlation across small subsets of wavelengths against a low background of random noise for all of the other wavelengths. The third pair of datasets are physical mixtures where the regressor variables represent the composition of a physical mixture, so the sum of the variables in each data row is 1.0. Consequently, the portrait correlated or collinear not full rank. Each of these three types of dataset is examined in both univariate response PLS1 and multivariate response PLS2 forms, making six datasets in total.

3.1 Wine Aroma : "PLS1 & Portrait"

Univariate response with more data rows than regressor variables. This example is the WineAroma dataset from Kowalski[63], a study of the effects of trace elements in pinot noir wine on the aroma. The response is the scores on the wine aroma from a panel of judges. The original paper has 40 observations, but 3 are strong outliers so the reduced dataset of 37 observations by 17 variables is used here. This reduced dataset is also used as a MINITAB example.

Even though the regressor variables represent the same physical quantity, the values range from 0.094 to 990 so are very different. the largest range is 990, the next 5 variables range from 36 to 36 with the other 11 between 0.1 and 61. This extreme variation is scale suggests that scaling may be a the only rational approach. The first principal component accounts for only 24% of the regressor variation and it requires 11 principal components to account for 95% of the variation. The response is approximately normally distributed.

The correlation between the 7th and 9th variables is high and these are also both highly correlated with the response. These variables are Barium and Strontium, as alkali metals they have similar chemical properties and are often occur together in natural deposits which explains their correlation. As their salts have no appreciable odour they probably have an effect on wine aroma by neutralising sugar acid astringency.

As a dataset for a model with linear terms estimated by OLS regression, the regressor design is rather poor with G-efficiency of 13% and an average prediction variance of 7.1. The condition number is 14.4 and the mean absolute correlation coefficient is 0.2179, which indicates a medium level of correlation.

In the summary of the collinearity diagnostics for the WineAroma dataset as Table 3.1, the high VIFs indicate that OLS MLR is not appropriate due to collinearity. The appearance of pairs rather than triples shows that the problem here is simply

correlation and not more complex collinear structures. The strongest dependency is between X_{15} and X_{17} where this dependecy accounts for 78% and 93% of the coefficients variation. But the correlation between X_2 and X_{10} accounts for the highest variance inflation factors. The third correlation between X_9 and X_{12} explains where the high variance inflation on the β_9 coefficient is coming from.

Cond'Idx	X2	X9	X10	X12	X15	X17
40		[0.34]		0.64		
59	0.72		0.89			
84				4	0.78	0.93
VIF	21.0	14.8	24.4	5.2	9.2	9.8

Table 3.1: WineAroma Dataset : Collinearity Diagnostics

3.2 Gasoline : "PLS1 & Landscape"

Univariate response with far more regressor variables then data rows. Near infrared spectroscopy of gasoline is a classical application for PLS. This dataset recurs in the literature, but was originally described by Kalivas[59]. The regressors are 60 observations by 401 variables and is massively correlated. The regressor variables range from 0.012 to 0.281 which is not extreme, so does not indicate if scaling is required. The first principal component alone accounts for 71% of the variation. The response if far from normally distributed. With more regressor variables than data rows, the experimental design characteristics of "Landscape" datasets cannot be assessed with respect to OLS regression. The condition number is 1.3×10^8 and the mean absolute correlation coefficient is 0.6905, which indicates a high level of correlation.

The regressor variables represent the same physical measurement - the absorption of infrared radiation at a characteristic set of wavelengths. Scaling the regressor variables here would tend to sink the relevant responses into the inflated background noise. For this dataset, scaling the regressor variables is not a sensible approach.

3.3 Waste Glass : "PLS1 & Mixture"

Univariate response with more data rows than regressor variables. Each variable row sums to 1.0. This WasteGlass dataset represents the composition of a glass designed

for nuclear waste encapsulation, the single response is the spinel liquidus temperature which is the highest temperature that crystals can exist in the melt. This dataset is from Piepel et al[93]. As a mixtures dataset, the overall negative correlation between variables is inevitable even though this is a statistically designed experiment.

The regressor variables range in scale from 0.00097 to 0.22, which is a ratio of 227:1 and so may indicate that scaling is appropriate before analysis. The first principal component accounts for only 17% of the regressor variance and it requires 10 principal components to accumulate 95% of the regressor variance. The response is approximately normally distributed.

The condition number is 1.42 and the mean absolute correlation coefficient is 0.0892, which indicates a low level of correlation. The four regressor variables are highly correlated with the response. As a experimental design for a linear term OLS mixture model, G-efficiency is 23%, the prediction variance is 2.3 average and 5.0 maximum. These poor design statistics are similar to the PLS2 & Mixture example and are both typical of mixture experiments in practice, where the design parameters are degraded by individual component constraints and the overall component sum.

3.4 Olive Oil : "PLS2 & Portrait"

Multivariate response with more data rows than regressor variables. The five regressor variables are all quality measures, the six responses are all sensory measures. This dataset is an example in the R pls library[82], originally from Massart et al[81]. The condition number is 1.68 and the mean absolute correlation coefficient is 0.4633, which indicates a medium to low level of correlation. As an experimental design for an OLS regression with linear terms, the G-efficiency is 15%, with an average prediction variance around 2.

The regressor variables range in scale from 0.008 to 1.25. This ratio of 1400:1 may indicate that scaling the regressors may be appropriate before analysis. The response variable ranges are from 10.05 to 63.7, which is not a concern. The first principal

component accounts for 59% of the regressor variance and only 3 principal componets are required to accumulate 95% of the variance. The response variables have similar strong structure with 3 principal components also accumulating 95% of the variance.

The collinearity diagnostics for the OliveOil dataset in Table 3.2 show no apparent strong collinearity between these regressors. But the diagnostics from the multivariate responses shown in Table 3.3 show that these are highly collinear. This indicates that PLS2 may be the most appropriate analysis.

Cond'Idx	X2	X3	X4
20	0.55		0.52
30	0.43	0.98	0.48
VIF	4.4	10.1	5.3

Table 3.2: OliveOil Dataset : Regressor Collinearity Diagnostics

Cond'Idx	Y1	Y2	Y4	Y5	Y6
81	0.93	0.97			0.55
203				0.98	0.97
VIF	125.2	117.7	25.8	28.1	4.7

Table 3.3: OliveOil Dataset : Response Collinearity Diagnostics

3.5 Biscuits : "PLS2 & Landscape"

Multivariate response with far more regressor variables then data rows. This is another near infra-red spectroscopy dataset, this time of biscuit dough. This is a well-known PLS dataset originally from Osbourne et al.[85]. Other analysis with this dataset are Brown et al[10], Goutis and Fearn[43] and KondylisWhittaker2012[62]. The regressors are 40 observations as data rows by 600 regressor variables as columns. The responses are measures of fat, sucrose, dry flour and water content.

The range of the regressor variable is from 0.0028 to 0.1285, which as a ratio of 46:1 is not excessive. Both data blocks show some correlation in that the first principal component explains 45% and 71% of the regressor and response variation respectively. The condition number is 16.17 and the mean absolute correlation coefficient is 0.4248, which indicates a medium level of correlation. As the observations are at sequential

infra-red wavelengths, the regressor correlation has a strong blocked structure. The regressor variables represent spectroscopic measurements, just as the Gasoline dataset. So for the same reasons, scaling the regressor variables would destroy the structure in the variations and so is not a sensible approach.

3.6 Abrasives : "PLS2 & Mixture"

Multivariate response with more data rows than regressor variables. This dataset represents a study of effects of various abrasives in friction material formulations. The regressors comprise 28 observations as rows by 9 component variables as columns. The 14 responses represent friction coefficients taken during a dynamometer performance test running an industry standard test procedure. For this study, only these 9 components were variable components, with the rest of the formulation fixed for each trial. Consequently, these component variables have been rescaled to 100% for this analysis, which is mainly why the design characteristics are better than the PLS1 & Mixtures WasteGlass dataset.

The regressor variables range from 0.03 to 0.10 and the response variables range from 0.08 to 0.28 neither of which is excessive. The regressors require 7 principal components to accumulate 95% of the variation while the reponses reqiure 8 principal components.The condition number is 8.3×10^7 and the mean absolute correlation coefficient is 0.1798, which is unusual in that it shows high collinearity combined with a low level of correlation. As a linear term ordinary least squares model, the G-efficiency is 60%, and the average prediction variance is 0.23, with a maximum around 6. This is a proprietary dataset, consequently it has not been published in the public domain.

3.7 Data Pre-Processing

It is taken for granted here that any rational approach to analysis will start by examining row and column minima and maxima, viewing column value distributions as histograms and scatter plots between columns to identify typing or transcription

errors in the data table. Beyond this, datasets for PLS analysis should clearly be centred prior to analysis so that cross products between **X** regressor and **Y** response represent covariance matrices. But should datasets be scaled? In principal, scaling can be applied to either the **X** matrix or **Y** matrix, or both. Scaling the **X** regressor matrix will change the relative covariance between each individual regressor and the response. For PLS1 univariate responses, scaling the response values will only change the overall magnitude of the covariance between **X** and **Y** , but will not change the relative covariance between individual regressors. Consequently, only scaling the regressor matrix will influence a PLS1 regression but not scaling the response vector. For PLS2, scaling either regressor or response matrix will change the pattern of covariance between these matrices, so will influence the PLS2 regression.

Wold, Sjostrom and Eriksson[119] recommend log transforms for variables with ranges that span more than one decade. If zero values are present then the fourth root is a good approximation for the log transform. They also take a pragmatic approach to using experience and knowledge to increase the scales for more informative **X** regressor variables, or reduce them if measurement variation issues are apparent. Their view is that it is better to make variables "passive" than to exclude them altogether.

Scaling does not always mean normalisation so that the column variances are unity. The data might be adjusted to account for specific calibration issues. For this aspect, Wehrens[108] gives a good description of how spectroscopic data can be improved by peak matching and baseline adjustment. Knowledge of the physical nature of the dataset variables can suggest if the variables should be scaled or not. Arneberg at al[3] gives a detailed description of how a square root scaling came to be used in a mass spectroscopy application. In this study heteroscedastic noise induced false negative correlations between major peaks and false positive correlation between minor peaks. Using a log transform destroyed the known linear correlation within the data, so a n^{th} root was preferred as it preserved perfect correlation while reducing partial correlation. The optimal transform of square root was identified by examining the replicates in the data. Without this specific scaling, false biomarkers were identified in the spectra.

The study datasets Gasoline and Biscuits are both from spectroscopic data, analysing these as centred not scaled is expected as simply normalising all columns of spectroscopic data tends to depress the data columns that contain the strongest signal and elevate those columns that are mainly background noise. Specific domain knowledge of these spectroscopic datasets is not available here, so more complex scaling has not been considered.

In the literature, scaling is something that is really only considered in detail for regressor matrices. For PLS2 multivariate response datasets there is no reason why both regressor and response matrices must be scaled in the same way. As each combination of regressor and response scaling leads to a different covariance structure and so PLS model, scaling regressors and responses should be considered independently to achieve the strongest model. Scaling permutations for the multivariate response example datasets are reported in the analysis reported later.

For ordinary least squares regression from scaled variates the residuals, coefficients and other statistics can be corrected by the scaling factors to recover the equivalent statistics from unscaled variates, within the limits of numerical rounding. PLS regression has a more nonlinear structure that cannot be represented by a linear function involving a hat matrix as OLS, so any form of scaling produces a distinctly different regression.

Transforming coefficients and residuals back to unscaled form is appropriate for interpreting the PLS model effects. PLS diagnostics and model structure are probably clearer in a scaled form. So the issue here is not about how the final model is interpreted, but is how to assess the effect of decisions on scaling may influence the PLS regression.

Chapter 4

Latent Variable Selection by Crossvalidation

When PLS is used in a calibration context, for example in the analysis of spectroscopy data, there is generally a large dataset that is used as "training data" prior to running any analysis of actual "test data". Augmenting the training dataset as required to stabilise the calibration leads to unambiguous estimates of the optimum number of latent variables. Where large datasets are being analysed, for example in observational data, the dataset can be split into typically 2/3 training and 1/3 test in order to identify the optimal number of latent variables, As selection of split can have a large influence on the structure of the model, specific methods for subset selection have been developed such as the DUPLEX method described by Snee[99]. Generally, insufficient data is available to support splitting the data into training and test sets so alternative methods derived from the whole dataset are required.

When datasets contain large number of replicated trials or observations, comparing the replicate error variation to the fit variation can be a useful check for over-fitting during latent variable selection. As reliable significance tests for variance require large number of samples to resolve small differences, using comparisons between fit and replicate errors for latent variable selection is not usually practical.

"Leave-one-out" crossvalidation is the latent variable method used in the early PLS literature. The Prediction Residual Error Sum of Squares ($PRESS$) statistic is

defined here for the general PLS2 case with m responses as the prediction error sum of squares

$$PRESS = \sum_{i=1}^{n}\sum_{j=1}^{m}(\mathbf{y}_{i,j} - \hat{\mathbf{y}}^{*}_{i,j,A})^2 \tag{4.1}$$

where $\hat{\mathbf{y}}^{*}_{i,j,A}$ is the predicted response for trial i response j from a model with A latent variables that excludes this trial i entirely from the model for all responses. Then the Root Mean Square Error of Cross Validation ($RMSECV$) is simply

$$RMSECV = \sqrt{PRESS/(n \times m)} \tag{4.2}$$

where n is the number of trials and m the number of responses. In the typical notation of PLS, the fit residual variation is usually given by RMSE

$$RMSE = \left(\sum_{i=1}^{n}\sum_{j=1}^{m}(\mathbf{y}_{i,j} - \hat{\mathbf{y}}_{i,,j,A})^2\right)/(n \times m) \tag{4.3}$$

where $\hat{\mathbf{y}}_{i,j,A}$ is the predicted response for trial i and response j from a model with A latent variables that includes all trials. The acronym MSEP for Mean Square Error of Prediction is often used in this context in the PLS literature. RMSECV is used here to show that this relates specifically to crossvalidation.

As excluding single or subsets of trials will change any centring or scaling, this is generally reset for each crossvalidation sample. A strategy of selecting the number of latent variables from the minimum values of these statistics is appealing, but awkward in practice as a clear minimum does not often appear. In these cases, softer criteria such as “The First Minimum” or “Start of the Plateau” have been proposed[112].

In the context of segmented crossvalidation, Martens et al[78] comment that there can be considerable uncertainty in the estimation of RMSEP. They show evidence of this from dissimilarity of RMSEP estimates across CV segments. This may be a particular problem with small datasets, as shown by Martens and Dardenne[77]. Hastie, Tibshirani and Freidman[46] recognised that ”Leave-One-Out” crossvalidation tends to select a number of latent variables that overfits the data, so propose ”K-fold” crossvalidation as an improvement. Xu and Liang[123] review the application of resampling for latent variable selection and propose Monte-Carlo Crossvalidation as a method between K-fold and full bootstrap resampling specifically for latent variable

selection in PLS. Any segmentation resampling method risks disrupting the collinearity structure or even reducing the rank of the resampled matrices compared to their original forms, with unknown consequences for the selection of the optimal number of latent variables. This issue has been referred to by Gouvénec at al[44] and by Wiklund et al[109], but remains an open question.

4.1 Crossvalidation against Fit Residuals Plots

The practical difficulty in choosing between the first or absolute minimum from either plots or tables of RMSECV against the number of latent variables was mentioned previously. Both the fit residual RMSE and the crossvalidated residual RMSECV should generally decrease as the number of latent variables increases up to the optimal selection. Beyond this point, RMSE will continue to decrease but RMSECV will diverge and increase due to overfitting. Consequently, the selection of the number of latent variables is generally much easier to identify visually from the "corner" in a plot of RMSE against RMSECV.

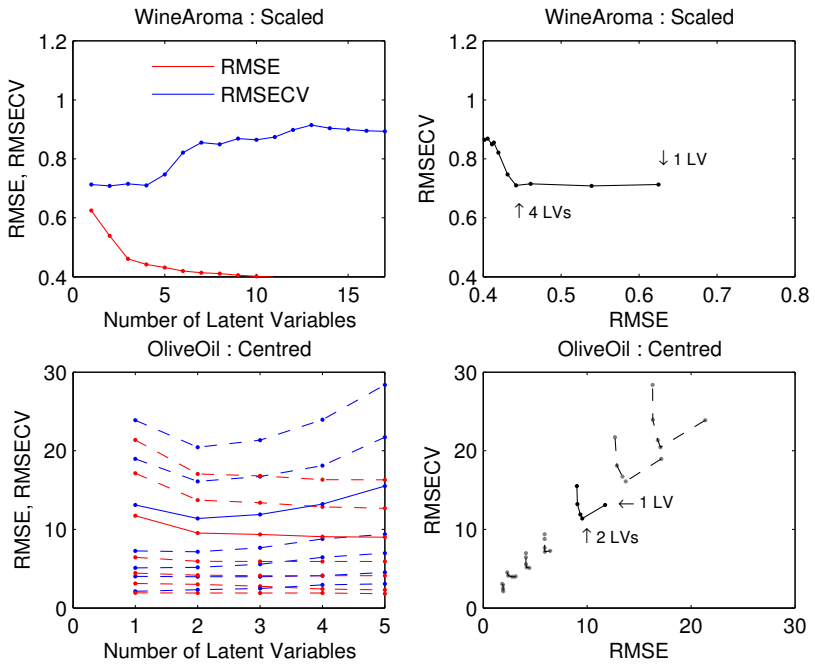

Figure 4.1: Plots for Latent Variable Selection from RMSE and RMSECV

Examples of this are shown in Figure 4.1. In the Wine Aroma RMSECV plot in the top left pane, the actual minimum RMSECV of 0.7731 at two latent variables cannot be distinguished from the RMSECV of 0.7757 at four latent variables. But the sharp corner at four latent variables is clear in the RMSE against RMSECV plot. In the bottom panes for the PLS2 Olive Oil dataset, the dotted lines are for the six individual responses and the solid line for the overall averages. Four of the responses have RMSECV minima at four latent variables, the other two minima are at one latent variable. The overall RMSECV minima at two latent variables is fairly clear in the bottom left pane. The corner at two latent variables is also clear in the RMSE against RMSECV plots for the overall mean and four from six of the individual responses. In this centred dataset, it is clear that the two responses with the largest variation are controlling the regression.

For WineAroma, Gasoline, Olive Oil and Biscuits example datasets, RMSE against RMSECV plots showed strong corners for latent variable location. The scaled Wine Aroma dataset used as an example plot does shows a clear corner at 4 latent variables, but the centred plot is very erratic which could indicate model instability. These plots are compared as Figure 4.2. Waste Glass and Abrasives, the two mixtures datasets did not show any obvious corners for latent variable location. These plots are shown as Figure 4.3. The centred line for the Waste Glass dataset in the top left pane shows a possible inflection at 3 latent variables, but both scaling options lines are monotonically decreasing. The structure of these datasets is described on pages 45 and 48. Both these mixtures datasets are based on experimental designs, so the only collinearity is that induced by the mixture component sum being constant. Consequently, selecting one less than the number of regressor variables for the number of latent variables might be reasonable. So in this case, the fact that neither plot selects a low number of latent variables is consistent with the structure of these datasets.

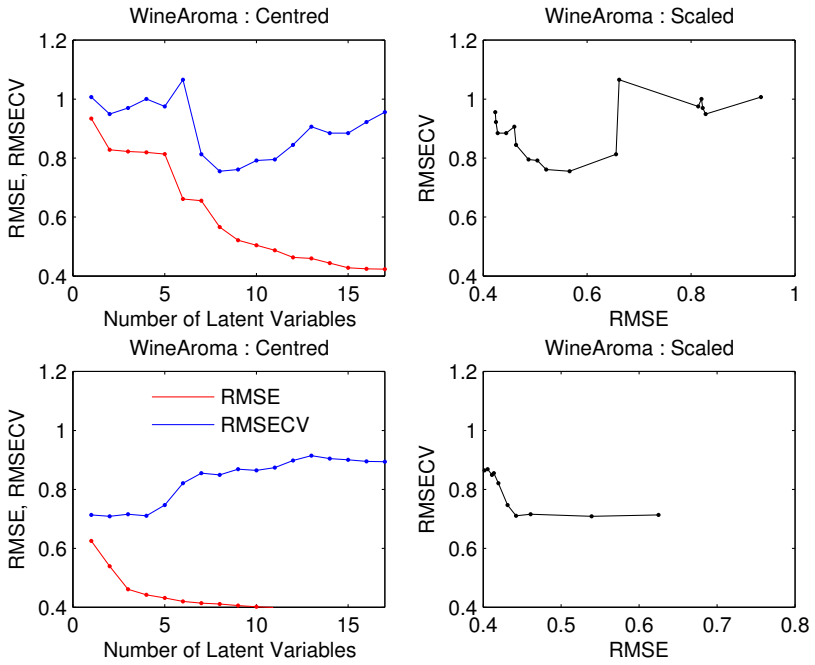

Figure 4.2: Wine Aroma Centred and Scaled RMSE and RMSECV Plots

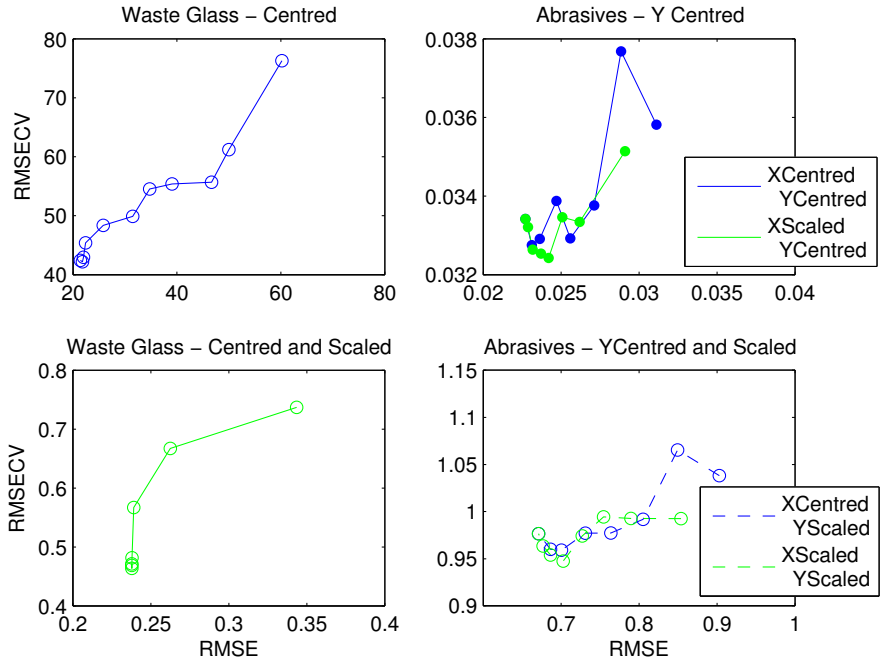

Figure 4.3: Mixtures Datasets RMSE and RMSECV Plots

4.2 Confidence Intervals for Crossvalidation

For PLS it is not clear how random variation in the regressors and responses is translated into uncertainty in the latent variable selection. As any RMSECV value is a statistical estimate and so a random variable, it follows that examination of the confidence intervals may lead to more reliable methods. Breiman et al[8] give an approximate confidence interval with a scaling factor of $1/\sqrt{2n}$ where n is the number of trials. A study of PLS RMSECV simulations by Faber[31] found that RMSECV is approximately χ^2 distributed, from which the same conclusion follows

$$\sigma_{RMSECV}/RMSECV \approx 1/\sqrt{2n} \tag{4.4}$$

Under the usual assumptions of independent and identically normally distributed residuals with zero mean, the confidence interval for the population standard deviation σ is from the sample standard deviation s is given by

$$(n-1)s^2/\chi^2(1-\alpha/2, n-1) \leq \sigma^2 \leq (n-1)s^2/\chi^2(\alpha/2, n-1) \tag{4.5}$$

Under the same assumptions, the confidence interval for an estimate of $RMSECV$ from a sample is

$$(n-1)RMSECV/\chi^2(1-\alpha/2, n-1) \leq RMSECV \leq (n-1)RMSECV/\chi^2(\alpha/2, n-1) \tag{4.6}$$

since

$$s = \sqrt{PRESS)/(n-1)} \tag{4.7}$$

$$= RMSECV \times \sqrt{n/(n-1)} \tag{4.8}$$

As a confidence interval calculation for RMSECV, all these assumptions are invalid. The crossvalidated residuals do not necessarily have a mean of zero. Further, the way that crossvalidation residuals are always derived from the same n samples also makes the "independent" assumption questionable. In particular, strong outliers can make the residuals measured by resampling non-random. For PLS, the sequence of residual matrices that are deflated after each latent variable are certainly not independent. Because of these issues, any probabilistic interpretation and significance levels of crossvalidated residuals for PLS latent variable selection is hard to justify.

This is well recognised in the PLS literature. For example, Martens et al[78] use the term "Reliability Range" rather than "Confidence Interval" for RMSEP statistics and propose this as a "rule of thumb". Even though the problems with crossvalidation are accepted, it remains the default method in most current PLS software.

4.3 Crossvalidation and the Example Datasets

The number of latent variables selected by the various RMSECV methods for the PLS1 univariate response datasets are shown in Table 4.1.First and absolute RMSECV minima location methods give fairly consistent selections. The RMSE vs. RMSECV plots generally showed strong corners to select a specific number of latent variables. This plot for the centred Wine Aroma dataset is very erratic, as is the RMSECV against latent variable plot for this dataset but 9 latent variables selection appears to be reasonable. The features of the Waste Glass mixtures dataset RMSE vs. RMSECV plot have been discussed previously in section 4.1 on page 53. In all cases, the K-fold crossvalidation prediction residuals were monotonically decreasing with increasing latent variables, so select the maximum number of latent variables by default.

	Wine Aroma PLS1 $n > k$		Gasoline PLS1 $n < k$		Waste Glass PLS1 mix'	
	Centred	Scaled	Centred	Scaled	Centred	Scaled
First Minimum	3	2	7	5	10	5
Abs. Minimum	8	2	7	5	10	5
RMSE vs. RMSECV	2 or 8?	4	5	4	11	11
5 Fold CV	17	17	30	30	11	11
10 Fold CV	17	17	30	30	11	11
First LV $< \chi^2$ CI	1	1	3	6	5	8
First LV < Breiman CI	7	1	3	4	3	4

Table 4.1: Numbers of Latent Variables Selected from RMSECV PLS1 Datasets.

In Table 4.1, the latent variables selected by confidence interval are the lowest latent variable whose interval contains the absolute minimum RMSECV value. The RMSECV plots with their confidence intervals are shown as Figure 4.4. In these plots the filled marker points indicate the first and absolute RMSECV minima. The confidence intervals from Breiman's[8] approximation are about half the range of the intervals calculated from the χ^2 distribution, but lead to similar conclusions for latent variable selection here. Each individual confidence interval for the Wine Aroma

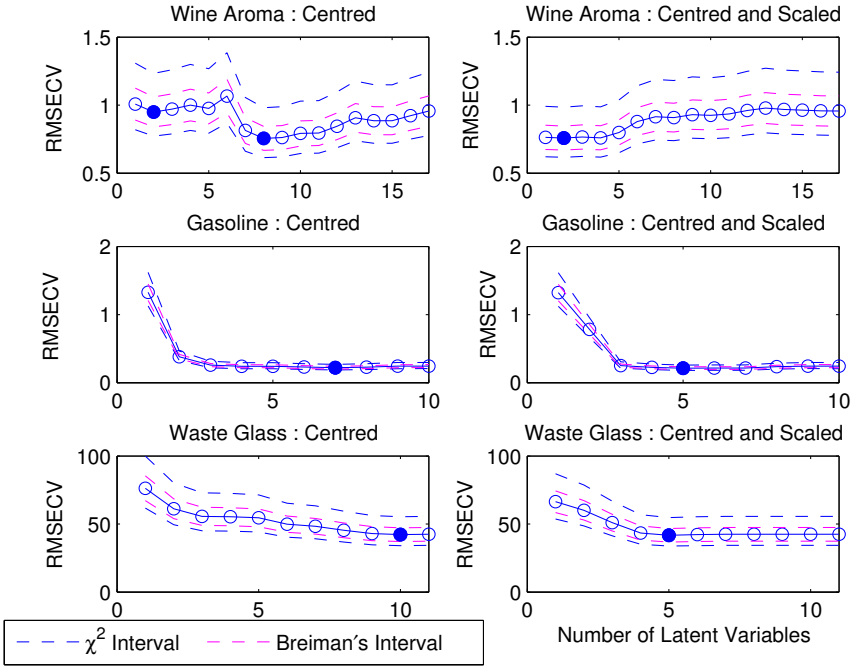

Figure 4.4: PLS1 Minimum Values in RMSECV

and Waste Glass datasets contains a good proportion of the other mean values, so the actual number of latent variables selected here is sensitive to way the interval is calculated. The confidence interval plots look more consistent for the Gasoline dataset which has the largest number of observations. For this dataset, only the confidence interval plots select 3 latent variables which is intuitively obvious from the RMSECV mean value plots.

RMSECV results for the PLS2 multivariate response example datasets overall are shown as Table 4.2. Apart from the K-Fold crossvalidation methods which again select the maximum number of latent variables, the selections are reasonably consistent within one latent variable. Generally, the number of latent variables selected by the most common occurrence in the individual responses is in agreement with the minimum in the combined RMSECV. The differences in the individual response RMSECV minima tabulated for the Biscuits dataset are caused by the first response RMSECV plot having a different shape than the other three responses. This is shown as Figure 4.5, where it is clear that the selection should be around 3 not 15 latent variables. The tabulated selected latent variables also look erratic for the WasteGlass dataset. The individual response plots are not shown, but the characteristic shape is a small rise in

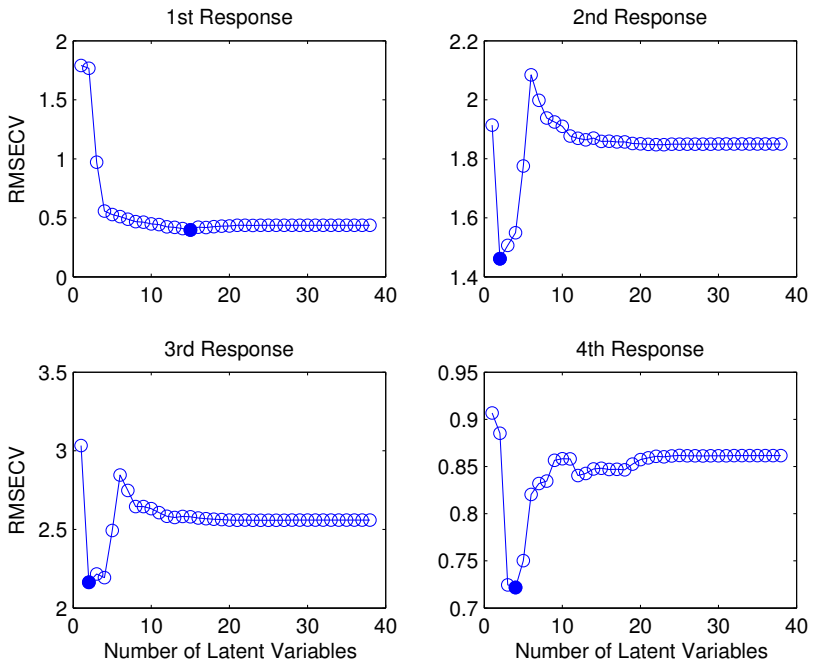

Figure 4.5: PLS2 Biscuits Individual Response RMSECV Plots

RMSECV for the first few latent variables followed by a sharp decrease with all higher latent variables having similar RMSECV values. This is not unexpected, considering that this dataset is from a designed mixtures experiment.

The multivariate PLS2 datasets appear to give clearer definition of the minimum RMSECV points than the univariate PLS1 datasets. This may be more due to the response averaging stabilising the latent variable selection for multivariate datasets than the increase in the overall number of individual observations. The variations is data scaling also appears to have less of an effect on latent variable selection for the PLS2 datasets than PLS1.

While the PLS literature does include apparently successful applications of K-fold and other variation on resampling crossvalidation, K-fold defaults to the maximum number of latent variables for all these datasets. Successful application may depend on larger datasets. A number of issues concerning this method were discussed in the introduction to this Chapter, overall it is concluded that this K-fold crossvalidation is not a viable method. There no strong evidence in favour of any of these crossvalidation

Olive Oil PLS2 $n > k$					
		XC YC	XC YS	XS YC	XS YS
Individual Responses	First Minimum Range	1-2	1-3	1-2	1-3
	First Minimum Median	2	2	1	2
	Abs. Minimum Range	1-2	1-3	1-3	1-3
	Abs. Minimum Median	2	2	1	2
Combined Responses	First Minimum	2	2	1	2
	Abs. Minimum	2	2	1	2
	RMSE vs. RMSECV	2	2	2	2
	5 Fold CV	5	5	5	5
	10 Fold CV	5	5	5	5
	First LV $< \chi^2$ CI	2	1	1	1
	First LV < Breiman CI	2	1	1	1
Biscuits PLS2 $n < k$					
		XC YC	XC YS	XS YC	XS YS
Individual Responses	First Minimum Range	2-15	3-9	2-13	2-6
	First Minimum Median	2	3	2	2
	Abs. Minimum Range	2-15	3-15	2-13	2-15
	Abs. Minimum Median	4	4	3	3
Combined Responses	First Minimum	4	4	3	3
	Abs. Minimum	4	4	3	3
	RMSE vs. RMSECV	?	4	4	4
	5 Fold CV	30	30	30	30
	10 Fold CV	30	30	30	30
	First LV $< \chi^2$ CI	3	4	2	3
	First LV < Breiman CI	3	4	3	3
Abrasive PLS2 mixture					
		XC YC	XC YS	XS YC	XS YS
Individual Responses	First Minimum Range	1-2	1-8	1-7	1-7
	First Minimum Median	1	4	3.5	1
	Abs. Minimum Range	1-7	1-8	1-8	1-7
	Abs. Minimum Median	7	4	4	5
Combined Responses	First Minimum	1	1	2	1
	Abs. Minimum	7	6	4	5
	RMSE vs. RMSECV	4	7	4	5
	5 Fold CV	8	8	8	8
	10 Fold CV	8	8	8	8
	First LV $< \chi^2$ CI	3	3	2	1
	First LV < Breiman CI	3	2	2	4

Table 4.2: Numbers of Latent Variables Selected from RMSECV PLS2 Datasets.

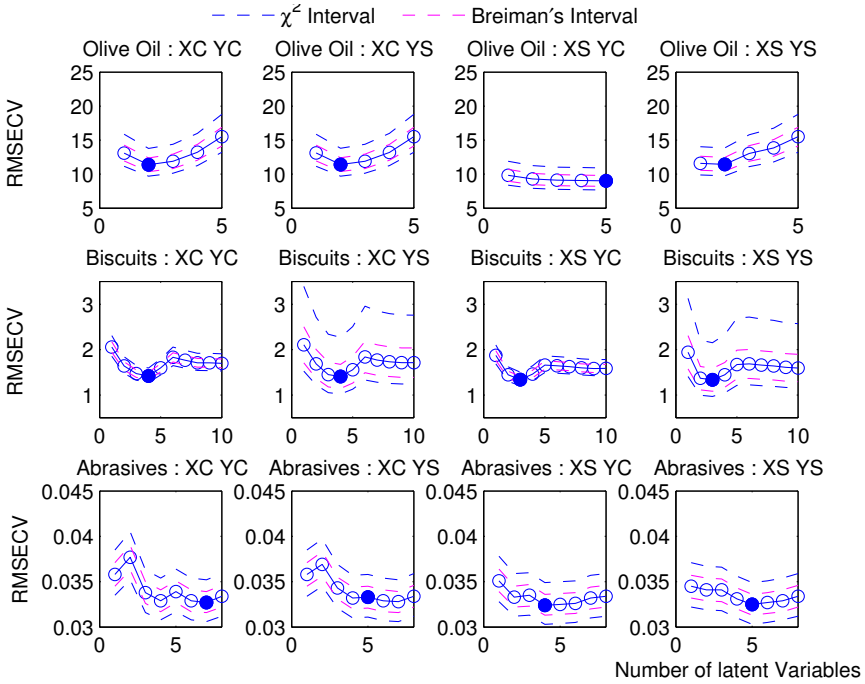

Figure 4.6: PLS2 Minimum Values in RMSECV - Combined Responses

method. The methods based on confidence intervals perform just as parsimonious versions of selection by absolute minimum RMSECV. So the Wold[112] rule of selecting the latent variable when the ratio of successive RMSECV values drops below say 95% is a viable alternative that avoids the dubious statistical assumptions.

The performance of the crossvalidation vs. fit residual plots proposed in section 4.1 is encouraging. For both mixtures datasets from designed experiments, all six plots showed monotonic changes towards the maximum number of latent variables as should be expected from the nature of these datasets. From the twelve plots for the other datasets, ten clearly identified a specific number of latent variables that is consistent with that of other methods. The other two plots were very erratic, a characteristic which was also apparent in the other crossvalidation methods.

Overall, it is fairly clear from the RMSECV tables how many latent variables should be selected for each dataset prior to further analysis. But the evidence against selecting one less or one more is weak.

Chapter 5

Latent Variable Selection by Permutations

A permutation or randomisation test is a hypothesis test where the reference distribution is obtained by the calculated test statistic under all possible permutations or as sample of all possible perms of the rows of the observations.

5.1 Randomised F-tests and t-Tests

Van der Voet[104] outlines a number of variations on t-test and F-tests by resampling data permutations. His proposed method for PLS is based on testing the differences between sample sums of squares and so is equivalent to a t-test.

Ordinary fit errors for two competing models A and B are given by

$$e_{i,A} = y_i - \hat{y}_{i,A} \tag{5.1}$$

$$e_{i,B} = y_i - \hat{y}_{i,B} \tag{5.2}$$

A test statistic $\overline{d}$ is then calculated as the mean difference between these errors.

$$d_i = e_{i,A}^2 - e_{i,B}^2 \tag{5.3}$$

$$\overline{d} = \sum d_i / n \tag{5.4}$$

As a randomisation test, the null hypothesis in this case is that the sum of the squares of the crossvalidation errors are equal, so that signs of the differences can be randomised

to generate the sampling distribution $\overline{d}$ under the null hypothesis. The pvalue for the test is obtained directly from proportion of calculated permutations that exceed the test value. As a method for determining the number of latent variables, crossvalidated residual variation at each latent variable increment are tested against the crossvalidated residual variation for the following increment. If the randomisation test shows no significant difference, then it is concluded that the any improvement in the residuals would be due to random chance, so no further latent variables are added.

The second example in van der Voet's paper concerns latent variable selection in PLS where the mean squared error in prediction crossvalidation in a reference set is compared to that of an evaluation set. Generating this sampling distribution under the null hypothesis by randomising signs assumes that the population means are equal. While the mean of the fit residuals for the reference set will be zero, it is unlikely that this will be so for the evaluation set. This difference in the means could have been corrected for, but the analysis as presented is not valid.

This issue is referred to "permutation unbiasedness" by Pesarin and Salmaso[89], p84. Testing crossvalidated residuals in this way is a particular problem, because the residual mean will be different for each permutation sample. Good[41] section 3.7.2 p58 presents a possible solution by comparing the variation about the sample medians based on a rank test by Aly[2].

5.2 Randomised Covariance Tests

Wiklund at al[109] gives an alternate form of the resampling test from van der Voet [104] that is more specific to PLS. As each iteration step, PLS minimizes the covariance between scores and responses. Consequently, they proposed a randomisation test based on this covariance. After A latent variables have been extracted, the test statistic is the covariance between the regressor $\mathbf{X}$ scores $\mathbf{t}_A$ and the response vector $\mathbf{y}$, $\mathbf{S}_0 =$

$Cov[\mathbf{t}_A, \mathbf{y}] = \mathbf{t}_A^T\mathbf{y}$. The null hypothesis being tested is

$$H_0 : \{\mathbf{y}_A \stackrel{d}{=} \mathbf{y}_{A+1}\} = \{\mathbf{S} = \mathbf{S}_0 = \mathbf{t}_A^T\mathbf{y}\} \tag{5.5}$$

$$H_1 : \{\mathbf{y}_A \stackrel{d}{>} \mathbf{y}_{A+1}\} = \{\mathbf{S} > \mathbf{S}_0 = \mathbf{t}_A^T\mathbf{y}\} \tag{5.6}$$

where $\stackrel{d}{=}$ means equal in distribution. This notation implies that the orders of the values in the $\mathbf{y}$ vector are exchangeable and that the reference distribution of $\mathbf{S}$ under the null hypothesis is generated by permuting the response rows $\mathbf{y}$ and recalculating the covariance. Wiklund et al state the importance of making the covariance permutation test on the deflated regressors and responses. While deflating either the regressors or responses is only necessary for calculating the PLS algorithm, deflating both is required for this permutation test as the noise values are inflated if they are not removed. So the specific method for testing a specific latent variable is to calculate the model with this number of latent variables, the compare the covariance between the deflated regressor and responses against their permutation distribution generated by permuting the deflated response rows. This covariance can either be calculated directly from the deflated regressors and responses or equivalently extracted from another PLS iteration with one latent variable.

Wiklund et al[109] only mention PLS1 with univariate responses. Using the equivalent $\mathbf{t}^T\mathbf{u}$ scores covariance in a permutation tests for PLS2 multivariate responses is an obvious extension as $\mathbf{t}^T\mathbf{u}$ remains a scalar for multivariate responses. For the permutations, it is important that it is only the row order of the responses that is permuted so that the response values for each row are kept together. This ensures that any correlation or collinearity structure within the response matrix is preserved. Wiklund et al go on to consider the form of the reference distribution to derive a probability estimate, but this is not necessary as an estimate of this can be obtained directly from the permutation test. With q random permutations, the number greater than or equal to some probability point p will have a binomial distribution $B(q, p)$ with mean qp and variance $qp(1-p)$. This gives an approximate 95% confidence interval of

$$p \pm 1.96[p(1-p)/q]^{1/2} \tag{5.7}$$

Around a critical pvalue of 0.05, 100,000 permutation samples are sufficient to determine the pvalue to around ± 0.001. For an exact permutation test for n observations,

100,000 permutation samples would be sufficient for 8 observations only, but not 9. For practical purposes 10,000 permutation samples gives a 95% confidence interval of ± 0.004 which is sufficient resolution around a critical pvalue of 0.05.

Other forms of null hypothesis are feasible here. The same paper comments that "in practice, any scrambling of Y data leaves some correlation between scrambled and un-scrambled data", so testing for zero covariance would not be effective. A more parsimonious test would be to compare the actual covariance for against the null distribution for the following latent variable - which is how van der Voet[104] structured his test.

$$H_o : \{y_1 \stackrel{d}{=} y_2\} = \{S = S_0 = t_{i-1}^t y\} \tag{5.8}$$

$$H_A : \{y_1 \stackrel{d}{>} y_2\} = \{S > S_0 = t_{i-1}^t y\} \tag{5.9}$$

This would test if the change in the covariance due to the additional latent variable exceeded that due to random noise in the residuals.

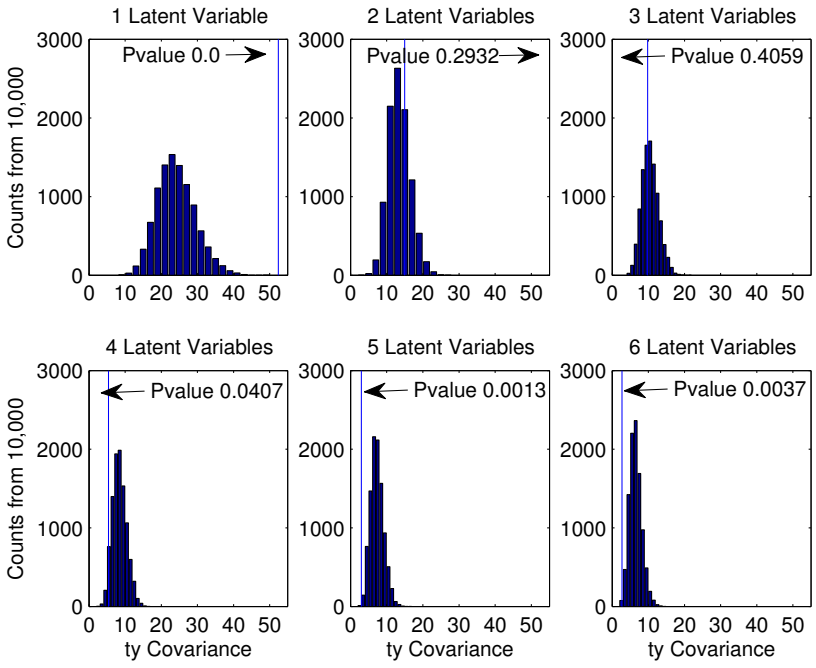

Figure 5.1: PLS1 WineAroma Covariance Tests

An illustration of covariance permutation tests for the PLS1 scaled WineAroma dataset is shown as Figure 5.1. The histograms are the null hypothesis distributions

generated by the permutation samples calculating the covariance for each permutation. The vertical blue lines show the $\mathbf{t}^T\mathbf{u}$ covariance test statistic for each latent variable. It might be anticipated that the covariance test statistic would start by exceeding the permutation distribution and gradually reduce with each additional latent variable until the test statistic moved into the permutation distribution. This covariance reduction into the permutation distribution is shown for the first three latent variables. For fourth, fifth and sixth latent variables, the covariance test statistic is less than the permutation distribution. This shows a particular "feature" of PLS in that it can detect some covariance even from random data. This is mentioned in Wiklund et al[109] and by Clark and Cramer[14] who found that this was a particular problem with small datasets in their simulation study. It is proposed here that the latent variable selection should either be the largest latent variable that exceeds the permutation distribution, or else the smallest latent variable thats is just less than the permutation distribution should there be none higher. Consequently, for practical purposes covariance permutations methods can be used in two ways. Either to identify specifically the smallest latent variable that is not representative of its permutation distribution, or as a more general method to identify ranges of plausible latent variables for comparison with other methods.

5.3 R^2 and Q^2 Permutation Plots

As an alternative to covariance permutations, Wiklund et al[109] and Eriksson, Trygg and Wold[29] extended earlier work by Lindgren at al[70] on R^2 permutation tests to include crossvalidated Q^2 permutation plots. The reasoning behind including crossvalidation into these plots is that as the Q^2 statistic calculated from crossvalidation is an estimate of the predictive power of a PLS model, then a permutation test of Q^2 would assess the statistical significant of this estimate. These plots were propsed as an improvement on covariance permutation histograms as they provide a warning of increasing chance correlations with increasing numbers of latent variables. An example of this plot for the centred Waste Glass example dataset is shown as Figure 5.2.

In this plot the ordinate is the correlation coefficient between the response vector

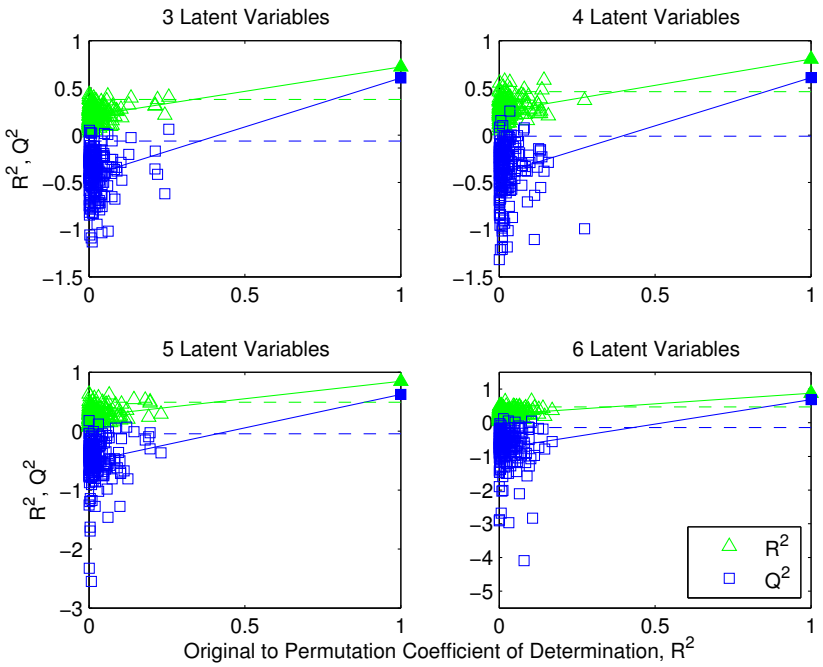

Figure 5.2: Waste Glass centred R^2 and Q^2 Permutation Plots

and it's permuted values and the abscissa is the reference and permuted R^2 and Q^2. Where the latent value is for a valid model, the R^2 and Q^2 are well outside of the range of their permuted values. In the plot shown here, the values on the right hand are the un-permuted values and the dotted lines are at the 95% limits of the permuted values for comparison. The comparison on the Y-abscissa is similar to the covariance permutation test. This plot adds further resolution along the X-ordinate where the permutation tests can generate spurious large correlations.

The R^2 and Q^2 permutation plots calculations, plots and examples given by Wiklund et al[109] and Eriksson,Trygg and Wold[29] are only for PLS1 univariate responses. For PLS2 multivariate responses, these plots can be applied to individual responses to identify which responses may be predictable from those which may not be. As R^2 and Q^2 are both based on ratios of sums of squares, the method can easily be extended to multivariate responses by simply accumulating the fit and total sums of squares across all responses so that their ratio is the overall response average. While permuting the row order, it is important that the values for each row are kept together as mentioned for the covariance permutations.

5.4 Permutation Tests and the Example Datasets

In the following discussion, a pvalue of 0.05 has been treated as a guide line rather than an absolute selection criterion. The pvalues for the permutation tests on the $\mathbf{t}^T\mathbf{y}$ covariance for the PLS1 datasets is shown in Table 5.1.

	Wine Aroma PLS1 $n > k$		Gasoline PLS1 $n < k$		Waste Glass PLS1 mix'	
	Centred	Scaled	Centred	Scaled	Centred	Scaled
1 Latent Variable	0.0051	0	0	0.0012	0.0277	0.0014
2 Latent Variables	0.0411	0.2916	0	0	0.0026	0.0120
3 Latent Variables	0.0651	0.4103	0	0	0.3782	0.0989
4 Latent Variables	0.0308	0.0439	0	0.0409	0.2118	0
5 Latent Variables	0.2164	0.0015	0.0107	0.0005	0.0574	0
6 Latent Variables	0.0089	0.0033	0.0635	0.1081	0.0952	0
7 Latent Variables	0.4744	0.0028	0.0310	0.4416	0.2471	0
8 Latent Variables	0.0184	0.0008	0.0732	0.1348	0.0333	0
9 Latent Variables	0.0579	0.0002	0.1673	0.3301	0.4541	0
10 Latent Variables	0.3916	0.0010	0.0386	0.4212	0.4758	0
Valid	1-4,6,8,9...	1,4-17	1-7, 9-10+	1-5	1,2,5,6,8	1,2,4-11
Select	2	1	6	5	2	2

Table 5.1: $\mathbf{t}^T\mathbf{y}$ Covariance Permutation Tests : PLS1 Datasets

For these univariate response PLS1 datasets the range of valid latent variables is generally split into two parts where the covariance is either above of below the permutation distribution. Even though these test statistics are quite erratic, selecting the largest latent variable that is significantly higher than its permutation distribution was not difficult. The results of the R^2,Q^2 have not been tabulated as all latent variables appear to be valid for these PLS1 datasets.

The covariance permutation pvalues for the PLS2 datasets shown in Table 5.2 show that the permutation tests based on $\mathbf{t}^T\mathbf{u}$ do not always identify a clear selection for the number of latent variables. Where a clear selection is possible, the permutation statistics for these PLS2 datasets appears to be less erratic than that of the PLS1 datasets. The R^2,Q^2 plots for the Biscuits dataset did not find any invalid latent variables. The OliveOil dataset found the valid latent variable range to be 2 to 3, or 2 to 4 depending on scaling. The plots for the Abrasives dataset found a valid latent variable range of 3 to 8. These ranges are consistent with the latent variables selected from the covariance permutation tests. During this analysis, it was sometimes difficult to decide on latent variable selection directly from the R^2,Q^2 plots, so the underlying

Olive Oil PLS2 $n > k$				
	XC YC	XC YS	XS YC	XS YS
1 Latent Variable	0.4625	0.1722	0.0562	0.0200
2 Latent Variables	0.2935	0.3472	0.3523	0.0501
3 Latent Variables	0.2408	0.1400	0.1198	0.0531
4 Latent Variables	0.3274	0.4217	0.2574	0.3568
5 Latent Variables	0.1090	0.2485	0.1263	0.3120
Valid	None	None	1	1,2,3
Select	?	?	1	3
Biscuits PLS2 $n < k$				
	XC YC	XC YS	XS YC	XS YS
1 Latent Variable	0.1482	0.1006	0.0170	0.0055
2 Latent Variables	0.0432	0.0803	0.0100	0.0618
3 Latent Variables	0.0565	0.0553	0	0
4 Latent Variables	0.0003	0.0010	0.1016	0.0688
5 Latent Variables	0.4300	0.2842	0.3476	0.2962
6 Latent Variables	0.4805	0.0682	0.0584	0.0672
7 Latent Variables	0.0280	0.2126	0.1055	0.2234
8 Latent Variables	0.2455	0.1486	0.3471	0.3692
9 Latent Variables	0.3836	0.4963	0.4130	0.3344
10 Latent Variables	0.4669	0.3869	0.0272	0.0545
Valid	2,3,4,7	3,4,6	1,2,3,6,10...	1,2,3,4,6,10...
Select	4	4	3	4
Abrasive PLS2 mixture				
	XC YC	XC YS	XS YC	XS YS
1 Latent Variable	0.4388	0.4604	0.3990	0.1300
2 Latent Variables	0.2198	0.4295	0.3580	0.2602
3 Latent Variables	0.2531	0.3738	0.4246	0.4696
4 Latent Variables	0.0889	0.0888	0.2678	0.1805
5 Latent Variables	0.0514	0.0040	0.1262	0.2190
6 Latent Variables	0.3745	0.3354	0.1710	0.2605
7 Latent Variables	0.0600	0.4774	0.4631	0.3587
8 Latent Variables	0.2955	0.4938	0.4819	0.1468
Valid	5,7	5	None	None
Select	5	5	?	?

Table 5.2: $\mathbf{t}^T\mathbf{u}$ Covariance Permutation Tests : PLS2 Datasets

data tables were also used.

For both PLS1 and PLS2 datasets, the difference between the number of latent variables selected for the centred and normalised datasets is also apparent here in the covariance permutation tests, which shows that these differences are associated with more than one method for selecting the number of latent variables.

As a test method, it is a concern that permutation methods could not locate a specific number of latent variables in all cases. This erratic behaviour was also noted by Wiklund et al[109] in their study of $\mathbf{t}^T\mathbf{y}$ permutations. Perhaps the most important statement concerning permutation tests in this paper is "... the components are assumed to enter the model in the natural order of decreasing relevance." The erratic behaviour of the permutation tests are explained here and in Faber and Rajko[30] by poor data pre-processing and subtle non-linearity in the spectra. The non-monotonic nature of the $\mathbf{t}^T\mathbf{y}$ or $\mathbf{t}^T\mathbf{u}$ covariance is not considered. This issue is examined in detail in Chapter 7.

Chapter 6

Latent Variable Selection by Information Criteria

Akaike's Information Criteria AIC[1] or the more conservative Schwartz[96] Bayesian Information Criteria BIC are generally used to select between multiple regression or time series univariate models containing different sets of terms. This is not a hypothesis test, but by selecting the model with the lowest AIC or BIC number the model that is the most likely fit to the data with the minimum number of parameters is selected.

For multiple regression, these information criteria are considered reliable and robust in that the minimum values are generally clearly defined - unlike $RMSECV$, so this approach is worth consideration for selecting the number of latent variables for PLS and other multivariate regressions. As model selection depends only on the difference in these values, information criteria like AIC and BIC are generally defined up to an additive or multiplicative constant.

The basic derivation of AIC from a linear least squares regression model's log-likelihood function with normally distributed errors is

$$AIC = n \log(RSS/n) + 2p \tag{6.1}$$

where p is the number of parameters in the model, n the number of observations and RSS is the residual sum of squares. From Venables and Ripley[106] for example,

expanding this about the residual sum of squares around an initial model and taking the first term as an approximation to the model specific part gives the form more generally used for calculation.

$$AIC = \frac{RSS}{\hat{\sigma}^2} + 2p \tag{6.2}$$

Where $\hat{\sigma}$ is the estimate of the error variance $RSS/(DoF-1)$. For ordinary regression, the number of degrees of freedom DoF is $n-p$. Estimation of the $\hat{\sigma}$ and RSS terms in the context of PLS latent variable selection is considered in the following section on degrees of freedom.

But this derivation is for linear regression. It could be applied to reduced rank regressions by interpreting the number of parameters p as the number of degrees of freedom. This generalisation is implied in the derivation of $AICc$ by Hurvich and Tsai[55] for univariate responses. For the following analysis, the uncorrected forms proposed by Krämer and Braun[64] specifically for PLS have been used.

$$AIC = \frac{RSS}{n} + \frac{2DoF\hat{\sigma}^2}{n} \tag{6.3}$$

$$BIC = \frac{RSS}{n} + \frac{log(n)DoF\hat{\sigma}^2}{n} \tag{6.4}$$

where DoF is the degrees of freedom for the PLS regression.

6.1 The Degrees of Freedom in PLS

Any calculation of information criteria depends critically upon the degrees of freedom, but the value of this quantity for PLS is far from clear. For ordinary regression, $n-p$ is used for the number of degrees of freedom. But Martens and Næs[80], Frank and Friedman1993[35] and others since argue that PLS uses more than this because it is a nonlinear function of the response. Further, this does not address the application of PLS to spectroscopic and other "landscape" datasets where $p > n$. As the error variance $\hat{\sigma}$ is estimated from $RSS/(DoF-1)$, it is clear that estimating the degrees of freedom has importance apart from latent variable selection. If no estimate of the degrees of freedom is available, then in practice the error variance must be estimated

by a resampling method.

Van der Voet[105] addresses this specific issue in the context of PLS and from an analysis of mean leverage in linear models proposed a pseudo-degrees of freedom calculated from ordinary and crossvalidated residuals.

$$PDoF = n(1 - RMSE/RMSECV) \tag{6.5}$$

where n is the number of observations and $RMSE$ and $RMSECV$ are from crossvalidated "Leave-One-Out" residuals as previously defined.

Both Phatak, Reilly and Penlidis[91] and Denham[17] propose using

$$DoF = trace(I_n - J^T\mathbf{X}^T)(I_n - \mathbf{X}J) \tag{6.6}$$

where I_n is the identity matrix of order n and J is the PLS coefficients Jacobian $\frac{\partial\hat{\beta}(y)}{\partial y}$. In a later paper Phatak, Reilly and Penlidis[92] point out that the term $(I_n - J^T\mathbf{X}^T)(I_n - \mathbf{X}J)$ is analogous to the hat matrix in ordinary regression.

A more fundamental definition of degrees of freedom comes from the structure of linear models

$$\hat{Y}_\lambda = H_\lambda Y \tag{6.7}$$

where $H_\lambda \in R^{p\times]}$ is the hat matrix independent of Y and λ is a fitting parameter.

So

$$H_\lambda = \frac{\partial\hat{Y}_\lambda}{\partial Y} \tag{6.8}$$

By definition

$$DoFs(\lambda) = trace(H_\lambda) \tag{6.9}$$

$$= E\left[trace(\frac{\partial\hat{Y}_\lambda}{\partial Y})\right] \tag{6.10}$$

Efron[23] gave this as the fundamental definition of degrees of freedom for generalised linear models. Based on this, Krämer and Sugiyama[65] used this specifically for PLS,

where the issue is that PLS is not a linear model that can be reduced to a hat matrix

PLS deflation can be written as a projection matrix $\mathcal{P}_{T,a}$ based on the score vectors $t_1...ta$, Höskuldsson[50] . This matrix $\mathcal{P}_{T,a}$ is considered as a "Euclidean orthogonal projection" by Boulet and Roger[7].

$$\hat{X}_a = X_0 - \mathcal{P}_{T,a}X_0 \tag{6.11}$$

$$\hat{Y}_a = \bar{Y}1_n + \mathcal{P}_{T,a}Y_0 \tag{6.12}$$

where $\mathcal{P}_{T,a} = T_{1..a}T_{1..a}^T$. So

$$\frac{\partial \hat{Y}_a}{\partial Y} = 1 + \frac{\partial \mathcal{P}_{T,a}Y_0}{\partial Y} \tag{6.13}$$

since $\mathcal{P}_{T,a}$ is a function of Y.

Hence, the number of degrees of freedom in a PLS regression with A latent variables is

$$\widehat{DoF}(A) = 1 + trace\left(\frac{\partial \mathcal{P}\mathbf{Ty}}{\partial \mathbf{y}}\right) \tag{6.14}$$

Since

$$\hat{Y}_a = X_0\hat{\beta}_a \tag{6.15}$$

$$= \bar{Y}1_n + \mathcal{P}_{T,a}Y_0 \tag{6.16}$$

Then

$$\frac{\partial \hat{Y}_a}{\partial Y} = X_0\frac{\partial \hat{\beta}_a}{\partial Y} \tag{6.17}$$

$$= \frac{\partial \mathcal{P}\mathbf{Ty}}{\partial \mathbf{y}} \tag{6.18}$$

The paper by Krämer and Sugiyama[65] on degrees of freedom in PLS also includes estimates of upper and lower bounds on the degrees of freedom. They prove that for PLS1, the lower bound on the degrees of freedom of the first latent variable is

$$DoF(A = 1) = 1 + trace(S)/\lambda_{max} \tag{6.19}$$

where S is the regressor correlation matrix and λ_{max} its largest eigenvalue. This is appropriate for centred and scaled regressors, for centred only regressors the equivalent form using the covariance matrix and its largest eigenvalue can be used. This covariance matrix is $\mathbf{X}^T\mathbf{X}/(n-1)$ for either "portrait" or "landscape" datasets. A naive estimate of the number of degrees of freedom is one for each latent variable plus one for the constant term, which gives a further lower bound. Krämer and Sugiyama give the upper bound on the number of degrees of freedom as the minimum of $[n-1, k+1]$ for PLS1. This is not quite correct as for $k > n$ "landscape" datasets there are sufficient degrees of freedom to account for the constant, in which case the upper bound is n not $n-1$.

For PLS2, where there are n trials as rows of observations for each of m responses. So the upper bound on the number of degrees of freedom is then $m(n-1)$ for $n >= k$ else mn. minimum of $[m(n-1)]$. Also for PLS2 there are m constants and as a naive estimate each latent variable would require m variables for each latent variable, that is $m(A+1)$ at A latent variables. The derivation of equation (6.19) is specifically for PLS1, but by taking the response vector as the first response scores vector $\mathbf{u}_1$ it is apparent that it is also applicable to PLS2.

For mixtures datasets the regressor variables have constant sum, which in effect reduces the number of independent variables by one. For ordinary least squares analysis of mixtures models without a constant term are the norm. Consequently, the mixtures constraint reduces the maximum number of degrees of freedom by one of PLS1 datasets or by m for PLS2 datasets where m is the number of responses.

So overall the rule for the upper bound on the number of degrees of freedom for should be

$$If\ (n >= k) \qquad DoFs = min[m*(n-1), m*(k+1)] \tag{6.20}$$

$$If\ (n < k) \qquad DoFs = min[m*n, m*(k+1)] \tag{6.21}$$

$$If\ mixtures \qquad DoFs = DoFs - m \tag{6.22}$$

6.2 Extending the PLS Degrees of Freedom Calculation to PLS2

For PLS1, the projection matrix $\mathcal{P}_{T,A}$ defined in equation 6.12 comes from the definition of PLS1 NIPALS

$$\mathbf{q}_1 = \mathbf{t}_1^T\mathbf{y}/\mathbf{t}_1^T\mathbf{t}_1 \tag{6.23}$$

$$\hat{\mathbf{y}}_A = \mathbf{T}_{1..A}\mathbf{Q}_{1..A} = \sum_{i=1}^{A}\mathbf{t}_i\mathbf{q}_i \tag{6.24}$$

$$= \sum_{i=1}^{A}\mathbf{t}_i\mathbf{t}_i^T/(\mathbf{t}_i^T\mathbf{t}_i)\mathbf{y} \tag{6.25}$$

$$= \mathcal{P}_{T,A}\mathbf{y} \tag{6.26}$$

This method for calculating degrees of freedom in PLS1 is shown as Figure (6.1)

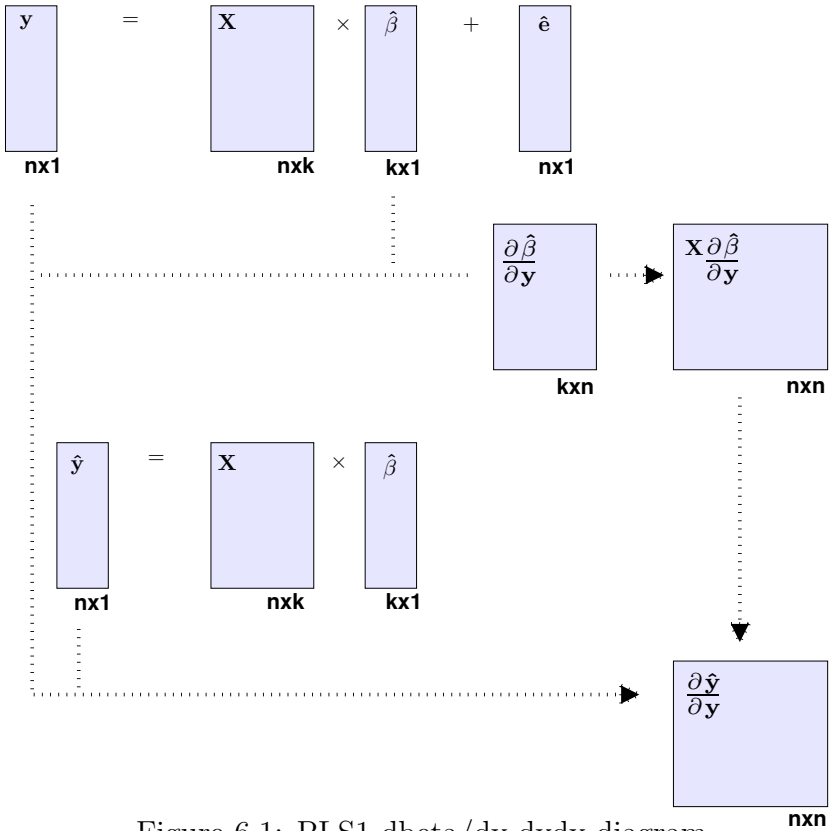

Figure 6.1: PLS1 dbeta/dy dydy diagram

So for PLS1, $\mathbf{y}$ and $\hat{\mathbf{y}}_A$ are $\in \mathbb{R}^{n\times 1}$, so $\mathcal{P}_{T,A}$ is $\in \mathbb{R}^{n\times n}$.

The equivalent form for the Y response matrix deflation for PLS2 is

$$\mathbf{q}_1 = \mathbf{t}_1^T\mathbf{y}/\mathbf{t}_1^T\mathbf{t}_1 \tag{6.27}$$

$$\hat{\mathbf{Y}}_A = \sum_{i=1}^{A} \mathbf{c}_i\mathbf{t}_i\mathbf{q}_i \tag{6.28}$$

$$= \sum_{i=1}^{A} \mathbf{c}_i\mathbf{t}_i\mathbf{t}_i^T/(\mathbf{t}_i^T\mathbf{t}_i)\mathbf{Y} \tag{6.29}$$

$$= \mathcal{P}_{T,A}\mathbf{Y} \tag{6.30}$$

So for PLS2, $\mathbf{Y}$ and $\hat{\mathbf{Y}}_A$ are $\in \mathbb{R}^{n\times m}$, so $\mathcal{P}_{T,A}$ apparently remains $\in \mathbb{R}^{n\times n}$. But as part of the derivation of the PLS degrees of freedom, equation (6.18) uses the partial derivative of the coefficients β with respect to the original y matrix. But this derivative is only defined if β and y are vectors, not matrices as is the case for PLS2. See for example Magnus and Neudecker[73], p82. For PLS2, equation (6.18) can be written in a vectorised form by sequentially stacking the columns.

For PLS1 $\mathbf{y}$ and $\mathbf{y}_A$ is $\in \mathbb{R}^{n\times 1}$, so in equation (6.18)

$$\frac{\partial\hat{\mathbf{Y}}_\mathbf{A}}{\partial\mathbf{Y}} = X_0\frac{\partial\hat{\beta}_A}{\partial\mathbf{Y}} \tag{6.31}$$

$\frac{\partial\hat{Y}_A}{\partial Y} \in \mathbb{R}^{n\times n}$, $X_0 \in \mathbb{R}^{n\times k}$. $\hat{\beta}_A \in \mathbb{R}^{k\times 1}$, so $\frac{\partial\hat{\beta}_A}{\partial Y} \in \mathbb{R}^{k\times n}$.

For PLS2, both β and $\mathbf{Y}$ are matrices, so their partial derivatives are not so straightforward. For A latent variables, the basic PLS2 fit equation is

$$\hat{\mathbf{Y}}_A(Y) = \mathbf{X}\hat{\beta}_A(Y) \tag{6.32}$$

The Jacobian between two matrices is given in Magnus and Neudecker[73] p173, from which

$$\frac{\partial\hat{\mathbf{Y}}_A(Y)}{\partial Y} = \frac{\partial vec(\mathbf{X}\hat{\beta}_A)}{\partial vec(\mathbf{Y}_A)^T} \tag{6.33}$$

$$= \mathbf{I}_m \otimes \mathbf{X}\frac{\partial vec(\hat{\beta}_A)}{\partial vec(\mathbf{Y}_A)^T} \tag{6.34}$$

The expansion of a product within the vec operator is also from Magnus and Neudecker[73] p31.

Here $vec(\mathbf{Y})$ and $vec(\mathbf{Y}_A)$ are $\in \mathbb{R}^{nm\times 1}$, so $\frac{\partial vec(\hat{Y}_a)}{\partial vec(Y)} \in \mathbb{R}^{nm\times nm}$. The Kronecker product $\mathbf{I}_m \otimes \mathbf{X}$ is the "outer" product so $\in \mathbb{R}^{nm\times km}$. $vec(\hat{\beta}_A) \in \mathbb{R}^{km\times 1}$ and $vec(\mathbf{Y}) \in \mathbb{R}^{nm\times 1}$, so $\frac{\partial vec(\hat{\beta})}{\partial vec(Y)} \in \mathbb{R}^{km\times nm}$, so the dimensions are consistent.

This method for calculating degrees of freedom in PLS2 is shown as Figure (6.2)

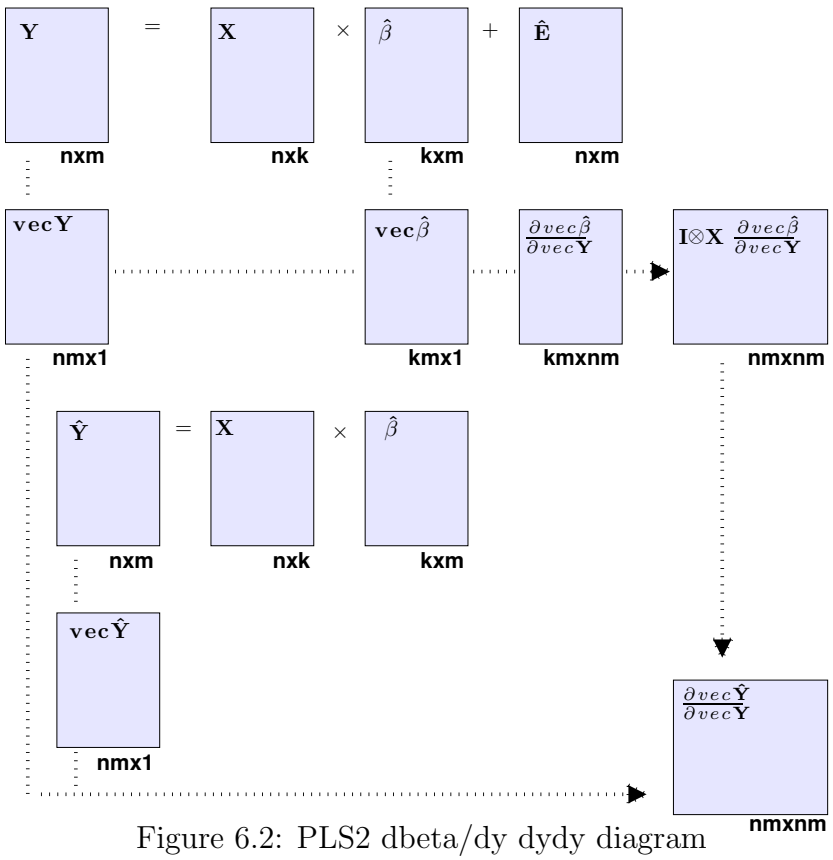

Figure 6.2: PLS2 dbeta/dy dydy diagram

Vectorising the matrices and their derivatives in this way is consistent with Efron [23] original definition of a generalised degrees of freedom, where the degrees of freedom are defined in terms of the number of observations. For PLS2 there are n trials but $n \times m$ observations. Just to be clear, vectorising the PLS2 calculation in this way is intended only for calculating the degrees of freedom after the the PLS deflation has been done. Vectorising $\mathbf{X}$ and $\mathbf{Y}$ prior to calculating PLS2 would destroy the collinear structure between the responses and so turn the calculation into PLS1.

Evaluating Efron's definition of generalised degrees of freedom shown as equation (6.10) would require deriving this as a function of the weights, loadings and scores.

This has been derived for NIPALS or Lanczos decompositions versions of PLS in Krämer and Sugiyama[65]. This derivation is valid for both $n < k$ narrow "portrait" datasets and $n > k$ wide "landscape" datasets as they use use examples of both. But as stated in the methodological background their derivation is limited to regressor matrices scaled to unit variance. The plsdof R library also automatically centres and scales all regressor matrices to unit variance. This derivation is also limited to PLS1 univariate responses as the partial derivative in equation 6.14 which is their equation 11 and only defined for vector responses. A further issue is that if the PLS regression coefficients are used in any derivation then the derivation is not general but specific to the specific to the coefficient definition used. So it is clear that extending their derivations to more general cases is not straightforward.

A practical solution here is to estimating these partial derivative elements in equation equation (6.10) directly using numerical methods. This has been found quite feasible and removes the constraints that are in the way of an algebraic solution. Numerical derivatives are directly applicable to "portrait" or "landscape" datasets. As this partial derivative is of the fitted response values against the observed values, any scaling factors cancel out, leaving the derivative to only detect the effect scaling on the way the changes in covariance influence the structure of the PLS model. Further, by stacking matrix responses into vectors as described above, numerical derivatives are also applicable to PLS2 multivariate response datasets. In short, they offer a complete solution. Details on these calculations are given later in the discussion of the example datasets.

6.3 Centring, Scaling and the Degrees of Freedom Calculation

Consider the centred and the scaled versions of OLS regression. Let X_c and X_z be the column centred and column centred then scaled forms of the regressors. Then

$$\mathbf{X}_{c,ij} = \mathbf{X}_{z,ij}\sigma_{X,j} \tag{6.35}$$

The centred and scaled versions of the regression coefficients can be given by the

generalised inverse

$$\beta_{cc} = X_c / Y_c \tag{6.36}$$

$$\beta_{zc} = X_z / Y_c \tag{6.37}$$

So

$$\beta_{cc,j} = \beta_{zc,j} / \sigma_{X,j} X_{c,j} \tag{6.38}$$

$$\frac{\partial \beta_{cc,j}}{\partial Y_{ci}} = \frac{\partial \beta_{zc,j}}{\partial Y_{ci}} / \sigma_{X,j} X_{c,j} \tag{6.39}$$

And

$$\mathbf{X}_{c,ij} \frac{\partial \beta_{cc,j}}{\partial Y_i} = \mathbf{X}_{z,ij} \sigma_{X,j} \tag{6.40}$$

$$= \frac{\partial \beta_{zc,j}}{\partial Y_{ci}} / \sigma_{X,j} X_{c,j} \tag{6.41}$$

$$= \mathbf{X}_{z,ij} \frac{\partial \beta_{zc,j}}{\partial Y_{ci}} \tag{6.42}$$

So the trace function in the degrees of freedom calculation is invariant to the X scaling, and it is clear how to rescale scaled coefficients and their derivatives back into the original dimensions and units. For OLS regression these relations are exact within calculation error. For PLS regression the coefficients, derivatives and degrees of freedom will be different between scaling and centring because scaling changes the collinear structures between the variates. By rescaling back to the original dimensions and units gives a basis for comparing the effects of scaling on the collinear structures.

6.4 Numerical Derivatives and Degrees of Freedom Calculation

The numerical derivatives for the degrees of freedom results presented here have been calculated by a high accuracy numerical method from the numDeriv R library by Gilbert[39]. The high accuracy "Richardson" default option was used in all cases. This was combined with PLS calculations from the pls R library by Mevik. Wehrens and Liland[82]. Degrees of freedom for PLS were also calculated using the plsdof R library by Krämer and Braun[64]. During the development of this work numerical derivatives were also calculated in MATLAB using generalised Romberg extrapolation

by D'Errico2006[18]. The version of PLS used was NIPALS written directly from the original source publications. No differences beyond minor rounding error effects where found.

6.5 Information Criteria and the Example Datasets

Comparing the analytical solution for the degrees of freedom from the plsdof library to that of the numerical derivatives is only possible for the scaled datasets. Further, it was found that these library routines failed to find a solution for the Gasoline dataset. Where comparisons were possible for the scaled Wine Aroma and WasteGlass datasets, the values from the library agreed with the numerical derivatives to at least 4 decimal places over most of the data range. Near the theoretical maximum number of degrees of freedom both methods showed minor instability and the close numerical match between the methods was lost.

The theoretical lower and upper bounds on the numbers of degrees of freedom shown in section 6.1 on page 72 are tabulated with actual minimum and maximum degrees of freedom as Table 6.1 for the PLS1 example datasets and as Table 6.2 for PLS2.

	Wine Aroma PLS1 $n > k$		Gasoline PLS1 $n < k$		Waste Glass PLS1 mix'	
n Trials	37		60		35	
k Regressors	17		401		12	
m Responses	1		1		1	
	Centred	Scaled	Centred	Scaled	Centred	Scaled
Min DoFs 1st LV	2.06	5.10	2.38	2.39	2.90	6.79
Max DoFs any LV	18	18	60	60	12	12
Deriv' DoFs 1st LV	2.05	5.64	2.26	2.27	5.71	12.31
Deriv' DoFs < naive	2	0	0	0	0	0
Deriv' DoFs max any LV	21.22	10.29	60.06	60.01	12.02	12.31

Table 6.1: PLS1 Limits on Degrees Of Freedom

The numbers of degrees of freedom from numerical derivatives for the Wine Aroma and Gasoline PLS1 are close to or beyond the theoretical limits. The maximum number of degrees of freedom for the Gasoline dataset given by Krämer and Sugiyama for PLS1 as a minimum of $[n-1, k+1]$ would give a value of 59. But the degrees of freedom from derivatives shows that the maximum is 60, which is why the correction was shown on 74. The values for the first latent variable are very close to the theoretical

lower bound, while the maximum degrees of freedom are well beyond the theoretical maximum for the centred Wine Aroma dataset. This dataset also showed two points where the number of degrees of freedom was less than the naive limit of one plus the number of latent variables.

	Olive Oil PLS2 $n > k$			
	XC YC	XC YS	XS YC	XS YS
Min DoFs 1st LV	2.00	2.00	2.71	2.71
Max DoFs any LV	36	36	36	36
Deriv' DoFs 1st LV	12.00	12.77	12.00	12.54
Deriv' DoFs < naive	0	0	0	0
Deriv' DoFs max any LV	36.00	36.00	36.00	36.00
	Biscuits PLS2 $n < k$			
	XC YC	XC YS	XS YC	XS YS
Min DoFs 1st LV	2.61	2.61	3.22	3.22
Max DoFs any LV	160	160	160	160
Deriv' DoFs 1st LV	8.69	8.70	9.45	9.46
Deriv' DoFs < naive	0	0	0	0
Deriv' DoFs max any LV	161.08	161.41	159.96	160.09
	Abrasive PLS2 mix			
	XC YC	XC YS	XS YC	XS YS
Min DoFs 1st LV	3.43	3.43	5.21	5.21
Max DoFs any LV	126	126	126	126
Deriv' DoFs 1st LV	31.41	31.12	34.74	33.33
Deriv' DoFs < naive	0	0	0	0
Deriv' DoFs max any LV	126.00	126.00	126.00	126.00

Table 6.2: PLS2 Limits on Degrees Of Freedom

The comparison between the degrees of freedom from numerical derivatives and the theoretical lower and upper bounds is much more consistent for the PLS2 datasets, showing only slight deviations above the theoretical maximum.

The degrees of freedom for the PLS1 example datasets are plotted as Figure 6.3. In these plots the green lines represent the minimum and maximum degrees of freedom limits and the naive estimate as the diagonal as mentioned previously at the end section 6.1 on page 72. For the PLS1 datasets, it is clear that the number of degrees of freedom for the centred datasets does not increase monotonically, while these scaled datasets are very nearly monotonic. The spike in the WineAroma centred plot at 6 latent variables is very curious, particularly as the RMSE against RMSECV plot shown as Figure 4.2 on page 55 also shows strong erratic behaviour. Apart from this single spike, the calculated degrees of freedom do not go outside of the theoretical limits. The Gasoline dataset is spectroscopic data where the maximum number of

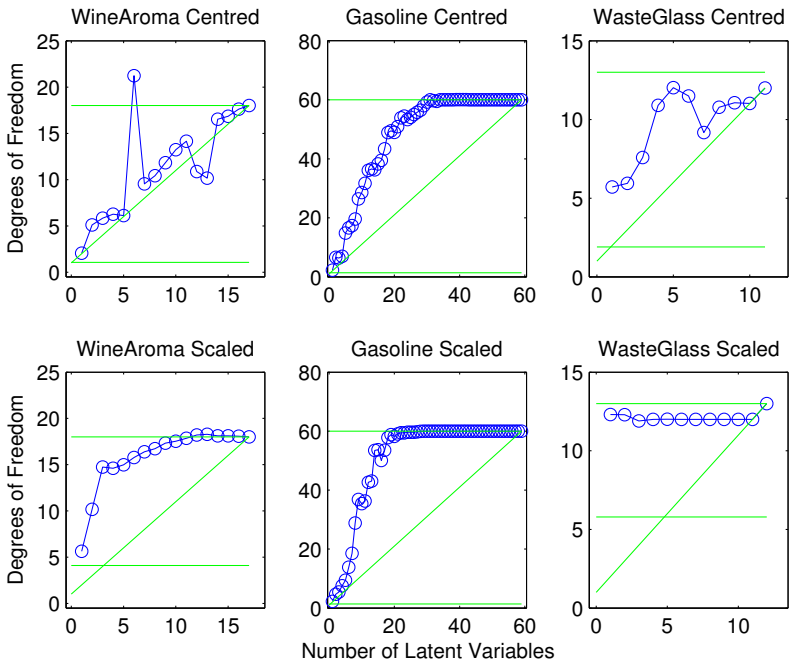

Figure 6.3: PLS1 Degrees of Freedom Plots, from Numerical Derivatives

degrees of freedom is 59. So from the asymptote in is plot it is apparent that the informative part of the response must be contained in about 20 to 30 wavelengths. As the Waste Glass dataset is from a statistically balanced experimental design for a mixtures dataset with 12 independent component variables and one overall component sum constraint, the 11 degrees of freedom or clear from the scaled dataset.

By comparison, in the degrees of freedom plots for the PLS2 multivariate response datasets for centred and scaled datasets are very similar, Figure 6.4. This apparent stability for PLS2 methods was also seen in the other latent variable selection methods. The asymptotic behaviour degrees of freedom for the $n < k$ "landscape" is also seen in the plot for the Biscuits dataset.

6.5.1 Van der Voet's Pseudo-Degrees of Freedom

In section 6.1 on page 72, a method for estimating the degrees of freedom from comparing fitted and crossvalidated residuals from van der Voet[105] was described. These pseudo degrees of freedom have been plotted against those from the numerical

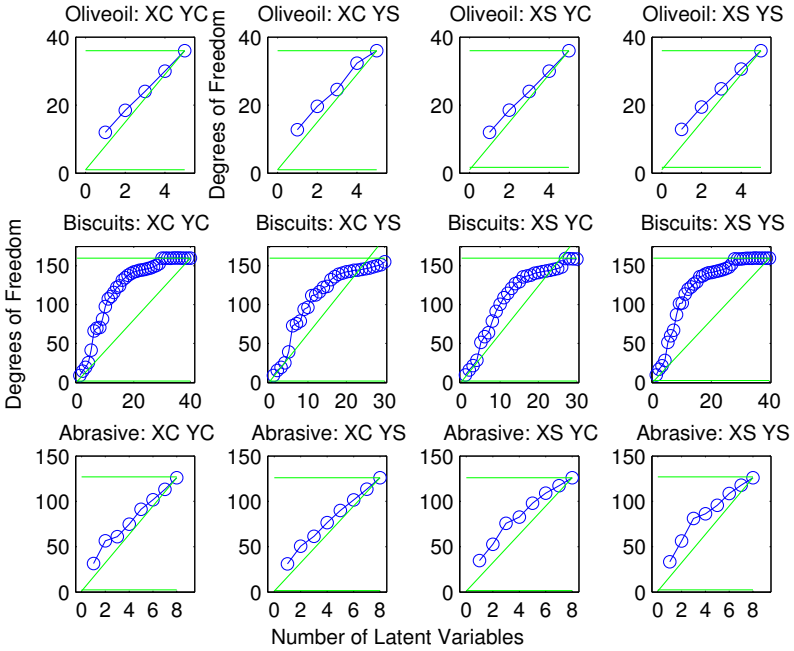

Figure 6.4: PLS2 Degrees of Freedom Plots, from Numerical Derivatives

partial derivatives in Figure 6.5 for the PLS1 datasets.

For the Wine Aroma and Gasoline datasets, this plot shows that the pseudo-degrees of freedom estimate is a good approximation to the true degrees of freedom. The erratic behaviour of the degrees of freedom for the centred WineAroma dataset is present in both pseudo and numerical derivative degrees of freedom, as indicated by the spike at 6 latent variables. This correspondence between pseudo and numerical derivative degrees of freedom breaks down for the Waste Glass mixtures datasets. The points where there is the largest difference are for the scaled Wine Aroma and both Centred and scaled Waste Glass. In all these cases the pseudo degrees of freedom exceed the upper bound. This only reason for this can be that the crossvalidated residuals RMSECV are high compared to the RMSE fit residuals. The cause could be that the leave-one-out resampling is making significant changes to the collinearity or predictive power of the models from these small datasets.

This comparison plot for the PLS2 datasets is shown as Figure 6.6, where the correspondence is again clear. Comparing the plots across both PLS1 and PLS2 datasets, it is apparent that there is a tendency for the pseudo-degrees of freedom to be slightly

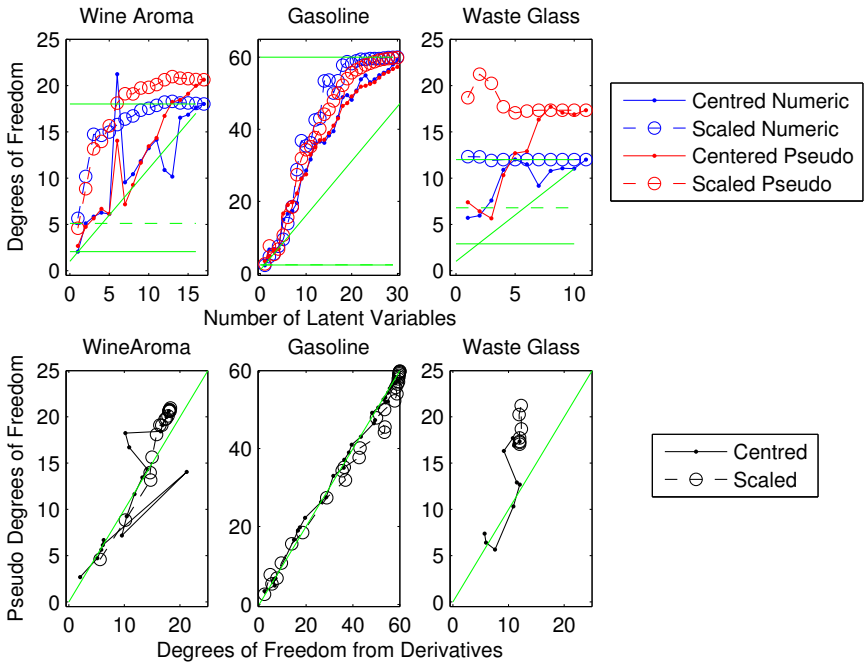

Figure 6.5: PLS1 Datasets Comparing Pseudo- to Derivative Degrees of Freedom

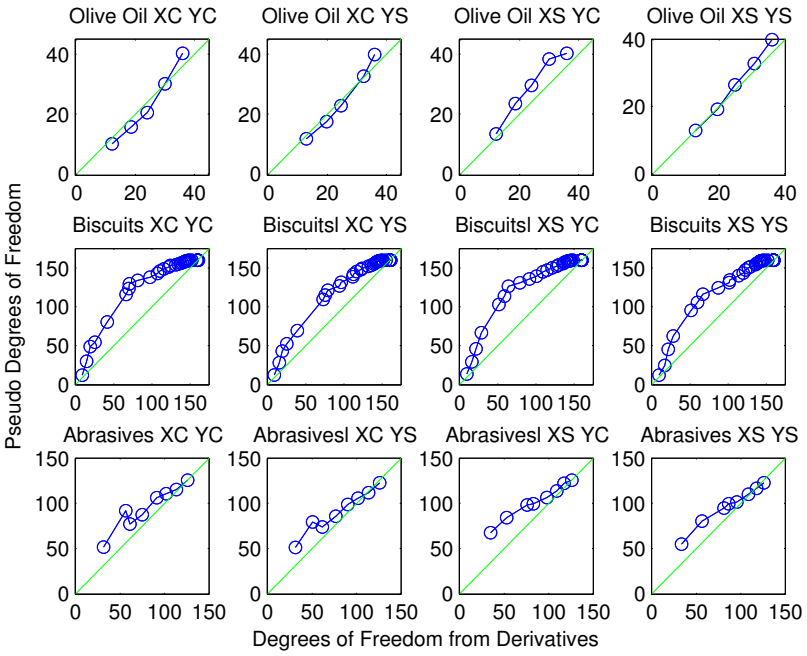

Figure 6.6: PLS2 Datasets Comparing Pseudo- to Derivative Degrees of Freedom

higher than that from the numerical derivatives. The plots for the Abrasive dataset show the largest difference at low latent variable numbers. Again, this is a small dataset that has one strong constraint as the physical mixture component sum, so resampling here may also be having an adverse effect of the model structures.

At this point it is clear that the PLS2 regressions appear to be much more stable than the PLS1 regressions. The PLS models for the centred Wine Aroma seem to be particularly unstable. So the comparison between the degrees of freedom and the theoretical lower and upper bounds shown in Tables 6.1 and 6.2 now appears to be more reasonable. Without the centred Wine Aroma dataset, the discrepancies between the degrees of freedom calculated from numerical derivatives and the theoretical bounds are small and might be explained as calculation error. The PLS1 examples shown by Krämer and Sugiyama[65] are entirely consistent with the calculated bounds. Consequently it is concluded that the bounds calculations shown in section 6.1 on page 72 are consistent with the example datasets.

6.5.2 Overall Conclusions about Information Criteria

The AIC and BIC information criteria have been calculated from equation (6.4) on page 72 using degrees of freedom derived from the numerical partial derivatives and from pseudo degrees of freedom. While there is no difference between the degrees of freedom estimates from the numerical derivatives and those calculated from the algebraic solution in the in plsdof library, the value of the residual variance $\hat{\sigma}^2$ in the plsdof library can calculated within a kernel algorithm which results in different values to those calculated simply from $RSS/(DoF - 1)$. The differences found for these example datasets were small and have no impact on latent variable selection.

The AIC and BIC information criteria based on the degrees of freedom derived from the numerical partial derivatives are shown as Figure 6.7 and Table 6.3 for the PLS1 datasets. In these plots the dashed vertical lines are at the minima, blue for AIC and red for BIC. For WasteGlass, there is little difference between the minimum AIC and BIC points from pseudo degrees of freedom, so the latent variable selection is not well resolved which is why the number of latent values selected by AIC is higher

than that by BIC. It is also interesting that the spike at 6 latent variables is not seen in the AIC or BIC plot from pseudo degrees of freedom, even though it is there in the degrees of freedom against latent variables plot shown as figure 6.3. This is a good indication to the cause. Both centred and scaled plots from the Gasoline dataset show an initial sharp decrease in AIC and BIC from the first to second latent variable and a smaller decrease for the third. After this the incremental change is very small up to the absolute minimum value at or near the maximum number of latent variables. This feature was also reported by Krämer and Sugiyama[65] in a simulation study. The Waste Glass mixtures dataset also shows similar asymptotic behaviour.

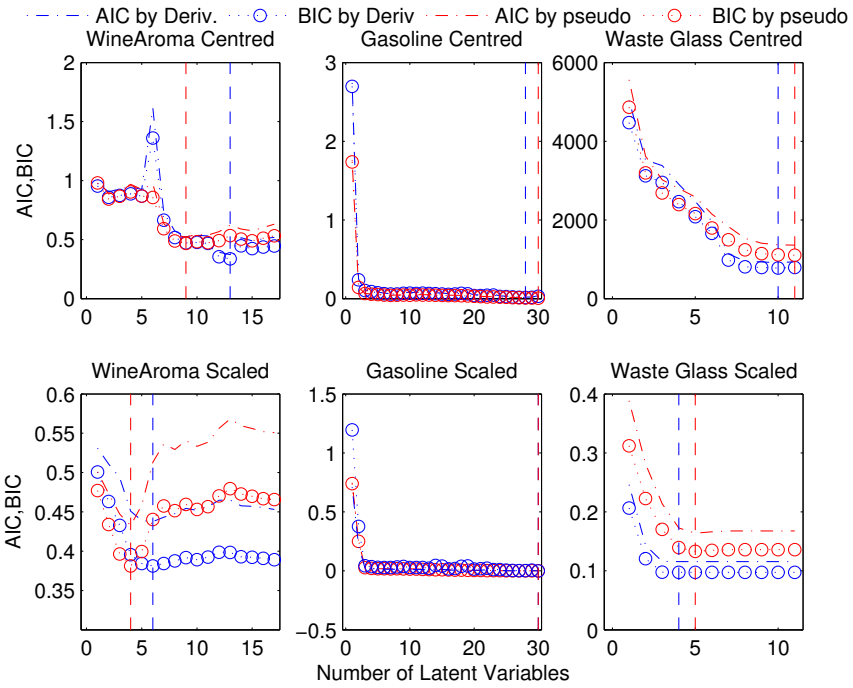

Figure 6.7: PLS1 Information CriteriaPlot

		Wine Aroma PLS1 $n > k$		Gasoline PLS1 $n < k$		Waste Glass PLS1 mix'	
		Centred	Scaled	Centred	Scaled	Centred	Scaled
min AIC	Deriv' DoFs	13	6	32	29	10	4?
min AIC	Pseudo' DoFs	9	9	58	59	11	5
min BIC	Deriv' DoFs	13	6	32	29	10	4?
min BIC	Pseudo' DoFs	4	4	58	59	11	5
Select		13	6	3	3	10	4

Table 6.3: PLS1 Latent Variable Selection By Information Criteria

Figure 6.8 and Table 6.4 show the corresponding AIC and BIC plots for the PLS2 datasets. The Olive Oil dataset shows clear minima in the plots, and selection by

AIC or BIC and numerical derivative or pseudo degrees of freedom all coincide at 2 latent variables. The $n < k$ "landscape" Biscuits dataset shows the same asymptotic behaviour as the PLS1 "landscape" dataset Gasoline, without showing such a clear location to select the number of latent variables. From inspection of the AIC and BIC data, there are no indications of stability or corners to indicate any particular latent variable selection. Stability and numerical minima in the information criteria occur around 30 to 25 latent variables, which is clearly overfitted as 100% fit to the response is achieved around 20 latent variables. From their simulation study, Krämer and Sugiyama[65] concluded that in those cases of PLS where information criteria plateau, the more complex models can have comparable predictive performance, which suggests that this failure to identify a specific number of latent variables may not be important in practice. For this Biscuits dataset and depending on the scaling combination, the response variance explained increases from around 50% to 99% across the first 10 latent variables. So this dataset does not have any information criteria plateau. The Abrasive mixtures dataset does shows clear information criteria minima, but with some difference between the various scaling and information criteria for number of latent variable selected.

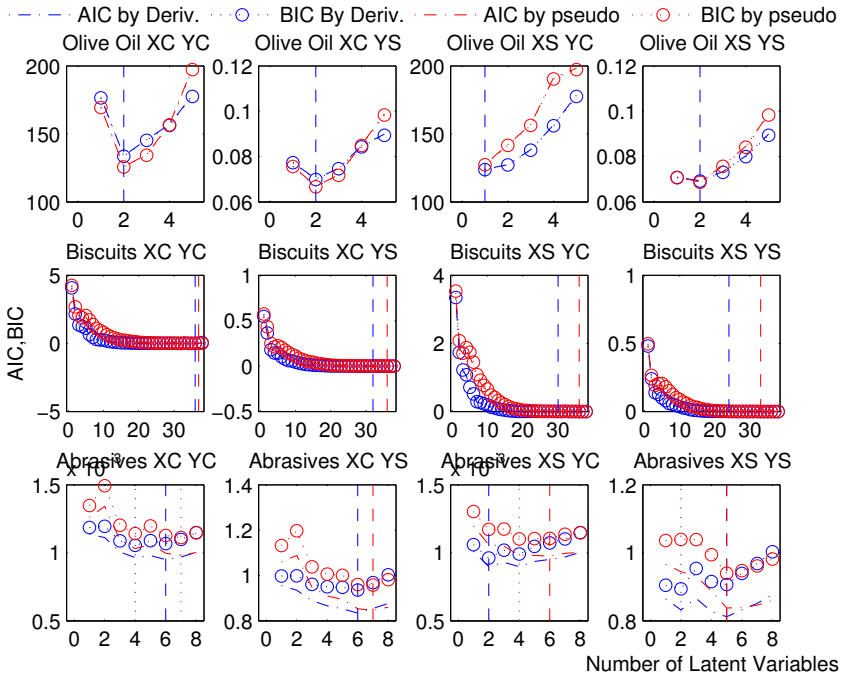

Figure 6.8: PLS2 Information CriteriaPlot

Form the tabulated values of latent variables selected, it is apparent that there

		Olive Oil PLS2 $n > k$			
		XC YC	XC YS	XS YC	XS YS
min AIC	Deriv' DoFs	2	2	1	2
min AIC	Pseudo' DoFs	2	2	1	2
minBIC	Deriv' DoFs	2	2	1	2
min BIC	Pseudo' DoFs	2	2	1	2
Select		2	2	1	2
		Biscuits PLS2 $n < k$			
		XC YC	XC YS	XS YC	XS YS
min AIC	Deriv' DoFs	31	32	35	35
min AIC	Pseudo' DoFs	38	38	38	38
min BIC	Deriv' DoFs	31	32	35	35
min BIC	Pseudo' DoFs	38	38	38	38
Select		?	?	?	?
		Abrasive PLS2 mix			
		XC YC	XC YS	XS YC	XS YS
min AIC	Deriv' DoFs	4	6	2	5
min AIC	Pseudo' DoFs	7	7	6	5
min BIC	Deriv' DoFs	4	6	2	5
min BIC	Pseudo' DoFs	7	7	4	5
Select		4	6	2	5

Table 6.4: PLS2 Latent Variable Selection By Information Criteria

is some variation is in those derived from pseudo degrees of freedom for the Wine Aroma dataset. The pseudo degrees of freedom method also failed to identify any rational values for the Biscuits dataset. Even though there is a close correspondence between pseudo and derivative degrees of freedom and their functions, there is a clear tendency for pseudo degrees of freedom to be higher than those from derivatives to the point where they can exceed the theoretical maximum. This difference does not lead to conservative estimates as the increase in degrees of freedom increases the number of latent variables selected. This may be only by one or two increments for most of these these dataset examples but as this this could lead to overfitting the conclusion here is that pseudo degrees of freedom are not a viable alternative to those based on numerical derivatives.

For the information criteria based on numerical derivatives, there are differences in latent variable selection between the various scalings as each scaling represents a different model structure. But in all these examples, the number of latent variables selected by AIC and BIC criteria are an exact match. This equivalence between AIC and BIC was also reported by Krämer and Sugiyama[65]. In OLS stepwise regression, the fine detail of the optimum model curvature is generally decided by the choice of AIC, AICc or BIC for term selection, which is one of the strongest criticisms of stepwise

regression. For these PLS examples, AIC and BIC based on numerical derivatives are selecting the same number of latent variables, suggesting that this application is more robust. The difference between AIC and BIC is in the terms accounting for the numbers of observations. But some of these example datasets are not large compared to those used for stepwise regression where selecting a particular information criteria is critical. So it is concluded that the robust aspects of applying information criteria to PLS latent variable selection must be something more fundamentally related to the increments in fit that are causing AIC and BIC to coincide. From equation (6.4) on page 72.

$$AIC = \frac{RSS}{n} + \frac{2DoF\hat{\sigma}^2}{n} \tag{6.43}$$

and $\hat{\sigma}^2 \approx RSS/(n - DoF)$, then

$$AIC \approx RSS\left(\frac{n + DoF}{n - DoF}\right) \tag{6.44}$$

The $(n + DoF)/(n - DoF)$ factor will increase rapidly once the degrees of freedom become more than a small proportion of the number of observations, so this apparent robustness must be due to the way PLS has a strong effect on the incremental change if fit residuals with each additional latent variable. It also suggests that the degrees of freedom may have diagnostic applications for PLS.

Chapter 7

Covariance Explained Plots

The objective function of PLS is to minimize the covariance between regressors and responses, or more specifically between the $\mathbf{X}$ and $\mathbf{Y}$ scores $t^T u$. So it was anticipated that this covariance would be a strongly decreasing monotonic function with the number of latent variables. This covariance for the example datasets is show as Figure 7.1 where a logarithmic scale is used to show the detail of the covariance changes over the entire range. In this plot, the Y centred and Y scaled covariances for the Olive Oil and Abrasive are nearly coincident. In all these plots the filled data points are those where the covariance has increased from the previous latent variable.

In Figure 7.1 there are instances where the incremental covariance between latent variables increases in 5 out of 6 of the example datasets. These increases in covariance occur in both centred and scaled datasets. The strongest apparent increase in covariance is in the analysis of centred WineAroma variates where the strongest deviations are at 4, 5 and 13 latent variables. These points correspond to the anomalies in the covariance permutation tests and degrees of freedom plots.

The issue here is not that PLS is failing to extract the maximum covariance between $\mathbf{t}^T\mathbf{u}$ score vectors. This covariance maximization is apparent numerically and has been proved algebraically either from the SVD decomposition of $\mathbf{X}^T\mathbf{Y}$ or by it's Lagrangian multipliers. But this covariance maximization is only "within" each latent variable and shows nothing about previous or subsequent latent variables. This makes PLS different to PCR regression where the variance explained is monotonic due to

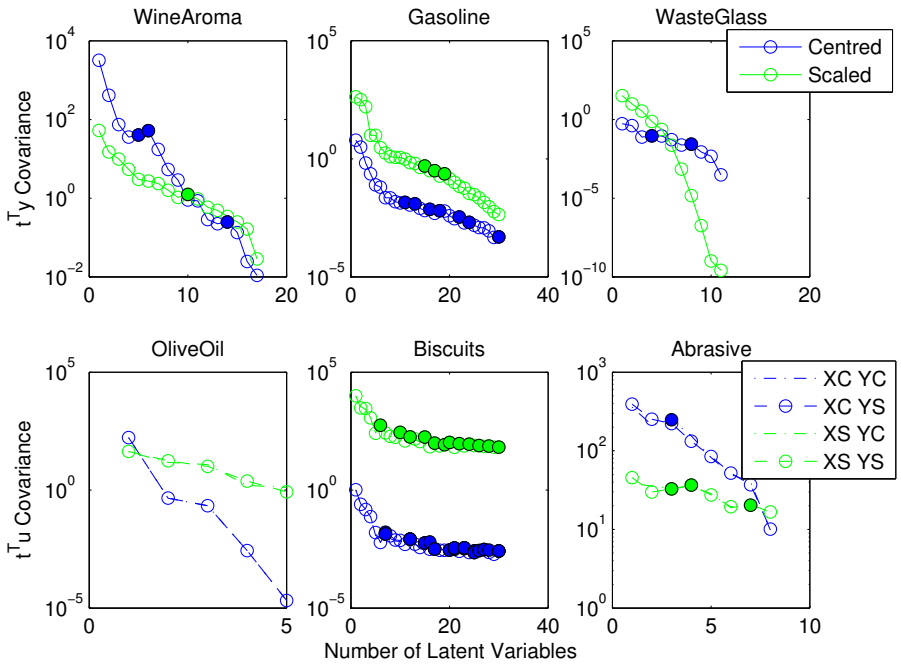

Figure 7.1: PLS Covariance Explained

the ordering of the eigenvalues. This non-monotonic behaviour for PLS covariance explained only relates to the step changes for individual latent variables. The cumulative variance explained will always be monotonic, as will be the (cumulative) regressor and response residuals.

For PLS, the X and Y variance explained generally both decrease with increasing latent variables. Figure 7.2 shows the X and Y covariance explained for the centred WineAroma dataset where the variance generally decreases, but occasionally increases are apparent. It is clear that for 4,5 and 6 latent variables weighted covariance maximization is reducing the X variance but increasing the Y variance. Patterns like this in the variance explained coincide with the incremental increase in scores covariance.

The reason for this is clearer in the linear variance explained plot Figure 7.3. After about 3,4 or 5 latent variables there is very little covariance left. This causes the weights vector calculation to become noise-sensitive and numerically unstable. This can result in the apparent increase in the residual covariance or covariance explained and as such is a clear indication of over-fitting. Martens and Martens[79] infer that this numerical instability is related to the weights $\mathbf{w}_{n_{DoFs}}$ being the first eigenvector

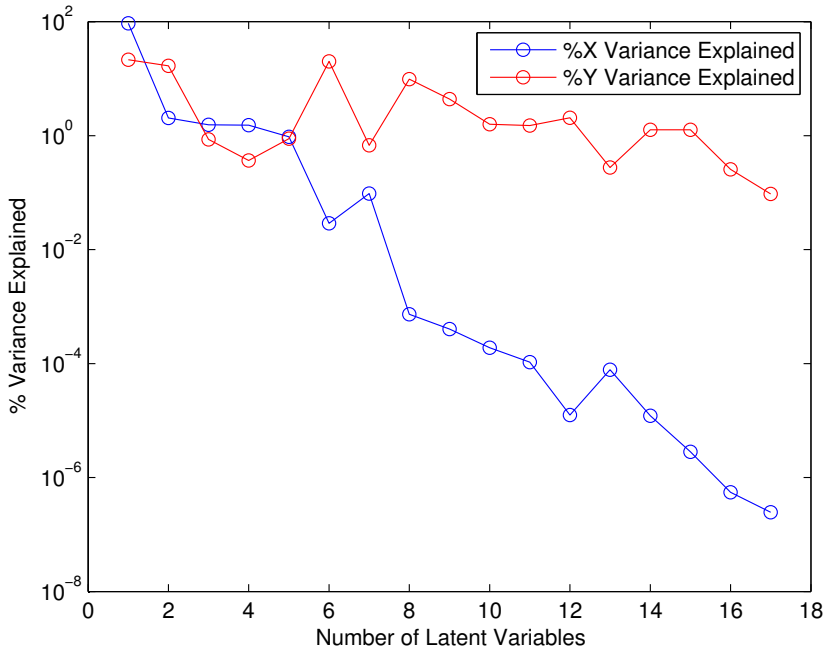

Figure 7.2: WineAroma Centred X and Y Incremental log Variance Explained

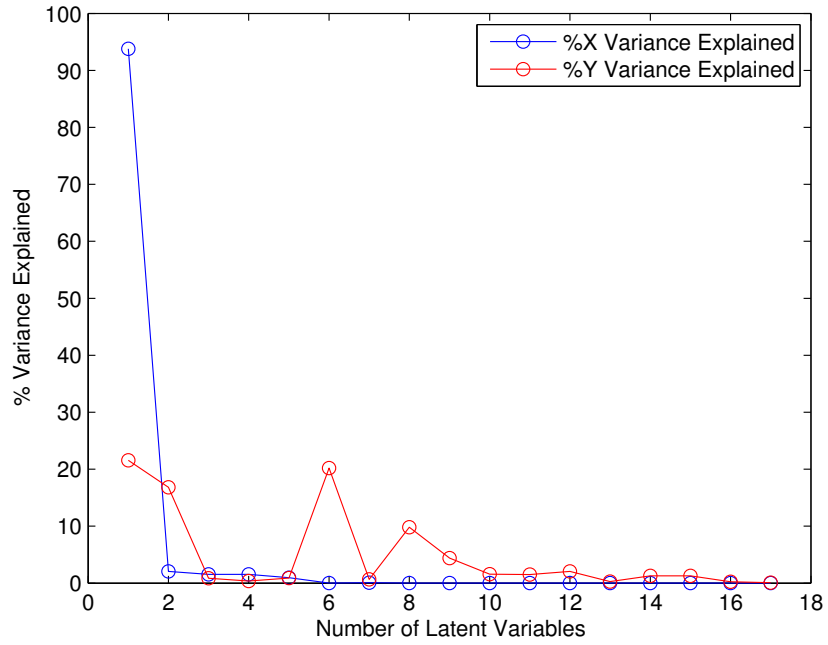

Figure 7.3: WineAroma Centred X and Y Incremental Variance Explained

of $(\mathbf{E}\mathbf{F}^T\mathbf{F}\mathbf{E}^T)_{n_{DoFs}-1}$. They suggest using the first eigenvalue of $(\mathbf{E}^T\mathbf{E})_{A-1}$ as the weights in these circumstances, but the practical conclusion is that this indication of over fitting should set the limit for the number of latent variables examined. So from tabulated values of the variances explained, the maximum number of latent variables is taken as the point at which either the $\mathbf{X}$ or $\mathbf{Y}$ cumulative variance explained first exceed 99% to prevent over-fitting.

	Wine Aroma PLS1 $n > k$		Gasoline PLS1 $n < k$		Waste Glass PLS1 mix'	
	Centred	Scaled	Centred	Scaled	Centred	Scaled
X Variance Explained	99.87%	99.78%	97.32%	98.81%	99.29%	47.59%
Y Variance Explained	40.52%	83.91%	99.06%	99.05%	93.85%	94.18%
Covariance Explained	97.91%	99.97%	98.58%	99.11%	99.63%	99.95%
Max Latent Variables	5	16	7	7	9	5

Table 7.1: PLS1 Maximum Numbers of Latent Variables from Over-fitting Criteria

	Olive Oil PLS2 $n > k$			
	XC YC	XC YS	XS YC	XS YS
X Variance Explained	99.59%	99.59%	100.00%	100.00%
Y Variance Explained	17.99%	37.92%	51.64%	57.12%
Covariance Explained	99.60%	99.60%	100.00%	100.00%
Max Latent Variables	1	1	5	5
	Biscuits PLS2 $n < k$			
	XC YC	XC YS	XS YC	XS YS
X Variance Explained	94.24%	94.61%	86.68%	86.86%
Y Variance Explained	99.25%	99.44%	99.21%	99.07%
Covariance Explained	94.50%	94.84%	86.94%	86.99%
Max Latent Variables	10	11	9	9
	Abrasive PLS2 mix			
	XC YC	XC YS	XS YC	XS YS
X Variance Explained	99.29%	99.26%	100.00%	100.00%
Y Variance Explained	57.74%	51.14%	55.39%	53.31%
Covariance Explained	99.16.%	99.14%	100.00%	100.00%
Max Latent Variables	7	7	8	8

Table 7.2: PLS2 Maximum Numbers of Latent Variables from Over-fitting Criteria

The maximum of 5 latent variables before overfitting for centred WineAroma is now reasonable as this accounts for nearly all the X variation. This dataset is dominated by the scale of a single variable as was mentioned in Section 3.1, so this limit of high X fit and low X fit is explained. Fitting the scaled dataset is an improvement as the response variance explained reaches 83.9% for 99.78% X compared to 40.52% Y for 99.87%X for the centred dataset. The maximum of 5 latent variables for other "portrait" landscape OliveOil is also reasonable as this accounts for all of the X variation for both centred and scaled datasets. This overfitting criteria does not appear to have

much effect on latent variable selection for the "landscape" datasets, Gasoline and Biscuits. This is also reasonable as these datasets have far more degrees of freedom available as is apparent from Figures 6.3 and 6.4. The centred WasteGlass dataset has a peak in the number of degrees of freedom plot, shown as Figure 6.3 on page 83. This peak is much weaker than the peak in the centred Wine Aroma plots, so not restricting the number of latent variables before overfitting to 3 or 4 latent variables is also reasonable here. This overfitting criteria does not make any strong restrictions on the mixtures datasets. The Abrasives PLS2 mixtures dataset has maxima at 7 or 8, where the maximum possible number of latent variables is 8. The maximum number of latent variables for Waste Glass is 11, so restricting the number of latent variables to 9 for the centred dataset is also reasonable. The scaled Waste Glass dataset is the exception here as 5 latent variables account for 99.95% of the X variation.

So it is now clear that applying this simple overfitting criterion has explained many of the anomalies that have appeared during the the latent variable selection for the example datasets. The use of a 99% limit was a quite arbitrary choice, but the actual value is not that important as any limit between 95% and 99.5% would give very similar results for these datasets. From these examples, it is clearly important to consider overfitting for both X and Y when selecting the number of latent variables for PLS.

Chapter 8

A PLS Simulation Study

8.1 Introduction and Background to PLS Simulation

The primary objective of this PLS simulation is to compare the performance of the various latent variable selection methods. But the question of identifying a best method leads to deeper questions about latent variable selection. How sensitive is the performance of a PLS model to selecting the optimal number of latent variables? Is it even sensible to look for an optimal number of latent variables, or are the requirements of model effects and model prediction so different that different PLS models with different numbers of latent variables are required?

Specific open questions concerning overall performance are ...

- How does PLS model fit performance in terms of residual RMSE, variance and bias relate to the structure of the dataset. Or equivalently, are there are combinations of dataset structure that are particularly good or bad for PLS performance?
- How do PLS factor effect measures like coefficients or VIPs relate directly to the responses or to the correlation structures? Or equivalently, how reliable is PLS for identifying the controlling regressors?
- How sensitive is PLS model prediction (with confidence intervals) to the characteristics of the dataset?

Open questions concerning LV selection are ...

- In practical terms, is there really an optimal number of latent variables where maximum model predictive performance and interpretive power coincide? Or should different PLS models be fitted from the same dataset depending on their purpose?
- Or equivalently, how sensitive is the performance of PLS models to mis-specification of the number of latent variables?
- How sensitive are the LV selection methods to the characteristics of the dataset?
- What is the strongest method for latent variable selection?
- Are there any specific circumstances where this strongest method is unreliable ?

In essence, all these questions concern sensitivity. The existence of an optimal number of latent variables is really about how sensitive is prediction and effect interpretation from PLS models to selecting a specific number of latent variables. Clearly, this sensitivity will be dependent on the characteristics of the data being modelled. Comparing alternative latent variable selection methods becomes much easier if the "'true"' number of latent variables in the dataset is known. But it is quite possible that selecting the best method may also depend on the characteristics of the data.

These questions are investigated here by a simulation study. It follows that the simulation needs to generate a set of PLS models with controlled structures and known response expectations and coefficients, to provide a basis for assessing the accuracy and reliability of the fitted PLS models. With many possible options for generating the sample datasets with different characteristics for a simulation, the settings for each simulation trial have been arranged as a statistically balanced experimental design, so that regression methods can be used to resolve how sensitive latent variable selection and PLS model performance is to specific aspects of the simulated datasets.

Many PLS publications use simulation to investigate some specific aspect of PLS behaviour, but investigations into the overall behaviour of PLS using simulation are

much less common. Næs and Martens[84] is a comparison of PLS1 and PCR by simulation where the data dimensions and true number of latent variables was fixed while the regressor and response error levels were varied. Their conclusions only concern prediction and were that PLS performs better than PCR with models up to the fixed number of latent variables with no difference after this.

Li, Morris and Martin[68] extended this simulation method to PLS2 datasets for a comparison of latent variable selection by crossvalidation and information criteria. The difference between the number of observations and regressors was taken as the degrees of freedom for the information criteria calculation, which is now know to be an overestimate. In this simulation the regressor dimensions was variable while the response structure, random error levels and collinearity structure was fixed. Conclusions are restricted to comments on variations on crossvalidation.

Kiers and Smilde[61] compare MLR, PCR, PLS, RR and other more specialised methods in a set of 5 simulation studies of multivariate reduced rank regression. The regressor collinearity structure, coefficient structure and response error levels were all systematically varied in the simulations. The true number of latent variables were either 1 or 2. No variation to the regressor matrix was introduced. Concludes that PLS and PCR typically recover the coefficients better than other methods when there is high collinearity, typically worse than other methods with low collinearity. Recovering the coefficients is often poor by all methods unless the coefficients lie in the subspace of the first few PCs of the predictor variables. Prediction less effected by collinearity than the coefficients. Latent variable selection methods were not considered.

Hulland, Ryan and Rayner[54] report a study of PLS1 path modelling by simulation. The number of regressors was fixed while the number of latent variables and correlation structure were variables. Coefficient identification was the only aspect considered. This study is explicit in that it structures the simulation as an experimental design. It is also unusual in that it includes response distribution as a factor, which was found to have no strong effect on coefficient identification.

This short review of PLS simulation does not claim to be comprehensive, but does illustrate how all the factors that may be relevant for PLS performance have been studied, but not all in the same simulation. So potentially strong synergistic or antagonistic interaction between factors cannot be identified.

8.2 PLS Simulation Methods

The published PLS simulations are studies of specific aspects of PLS and so cover a restricted number of of the possible factors that could influence a PLS regression. The intention here is to include a comprehensive range of PLS factors, so that particular characteristics of datasets that lead to specific sensitivities of a LV fitting method or overall PLS model performance may be identified.

As PLS1 univariate response datasets require different algorithms to PLS2 multivariate responses, these have also been investigated using separate simulations.

The simulation method presented here is based on the reduced rank multivariate regression methods from Naes and Martens [84] and from Li, Morris and Martin [68]. These methods have been extended to give more control over the regressor correlation and collinearity structure and in particular the relation between regressor and response collinearity structures. Further extensions were required to include response distributions as a simulation factor. The simulation strategy used here is to use a common method to first fix the regressor matrix structure, then to fix the coefficient structure and use this to calculate the responses.

8.2.1 Regressor Matrix Simulation

Let v be a vector of length A^*, then $\sigma = diag(v)$.

Let $\mathbf{R}$ be a matrix of size $n \times A^*$ with independent columns of a specific distribution and variances equal to the elements of v, that is covariance σ.

Then let ξ be a random orthonormal matrix of size $k \times A^*$.

Then the regressor matrix is given by $\mathbf{X} = \mathbf{R}\xi^T$. Consequently the properties of $\mathbf{X}$ are

- Dimensions $n \times k$.
- Reduced rank, A^*. This fixes the degree of collinearity between the regressors.
- Is a linear function of $\mathbf{R}$ so inherits the distribution from $\mathbf{R}$.
- Has covariance σ irrespective of the inherited distribution. This is dependant on the way $\mathbf{R}$ is generated.
- The eigenvalues of the covariance of $\mathbf{X}$ is v.
- The sequence of eigenvalues of the covariance of $\mathbf{X}$ is VDR_X - the Regressor Variance Decay Rate. This is explained in detail later in section 8.2.3.
- This sequence of eigenvalues determines the level of correlation and collinearity between the columns in $\mathbf{X}$. This fixes the degree of correlation between the regressors, independently from the degree of collinearity.
- The eigenvectors of the covariance of $\mathbf{X}$ is ξ.

8.2.2 The Regressor and Response Distributions

There is nothing in the PLS method that uses any assumptions on the data distribution.

But a number of references state that for practical purposes, PLS "works better" if the data is normal or else transformed into something like normal. Evidence in support of this is anecdotal. Probably the strongest reference is "Megavariate and Multivariate Data Analysis" [27], [28] which is a 2 volume guide to the practical aspects of PLS by Eriksson, Johansson, Kettaneh-Wold, Trygg, Wikstrom and Svante Wold which does go into some detail on data transforms.

PLS simulation with normal variates is possible because linear transforms matrices with normally distributed columns produce other matrices that are also normally distributed. For multivariate data, these linear transforms do not change the correlation between the variates. But distributions other than normal are usually changed by linear transforms. Also multivariate normal distributions are changed by nonlinear transforms which tend to reduce the (absolute) correlation between the variates [4].

While nonlinear transforms of multivariate normal data may change the covariance between variates, they do this is a determinate way. Johnson, Ramberg and Wang [57] show how the multivariate means, variances and correlation change on lognormal and hyperbolic sine transforms. They also give the following inverse formulae, which give then multivariate normal means, variances and correlations required to produce specific means, variances and correlations on lognormal or hyperbolic sine transforms.

If $\mathbf{X}$ has a multivariate lognormal distribution with k variates, means $\mu^{'}$, variances $\sigma^{'2}$ and correlations $\rho^{'}$ and $\mathbf{X} = exp(\mathbf{Y})$. Then $\mathbf{Y}$ is multivariate normal with means μ, variances σ^2 and correlations ρ given by

$$\mu_i = log(\mu_i^{'2})/\sqrt{\sigma_i^{'2} + \mu_i^{'2}} \qquad i = 1, \ldots, k \qquad (8.1)$$

$$\sigma_i^2 = log(1 + \sigma^{'2}/\mu^{'2}) \qquad i = 1, \ldots, k \qquad (8.2)$$

$$\rho_{i,j} = (1/(\sigma_i \sigma_j)) log\left[1 + (\rho_{i,j}^{'} \sigma_i^{'} \sigma_j^{'})/(\mu_i^{'} \mu_j^{'})\right] \qquad i, j = 1, \ldots, k \qquad (8.3)$$

Similarly, if $\mathbf{X}$ has a multivariate hyperbolic sine distribution with k variates, means $\mu^{'}$, variances $\sigma^{'2}$ and correlations $\rho^{'}$ and $\mathbf{X} = sinh(\mathbf{Y})$. Then $\mathbf{Y}$ is multivariate normal with means μ, variances σ^2 and correlations ρ given by

$$\mu_i = (1/2)cosh^{-1}\left[1 + 2\mu_i^{'2}/\left(1 - \mu_i^{'2} + \sqrt{\mu_i^{'4} + 2\mu_i^{'2} + 2\sigma_i^{'2}}\right)\right] \quad i = 1, \ldots, k \quad (8.4)$$

$$\sigma_i^2 = \begin{cases} 2log(\mu_i^{'}/sinh(\mu_i)) & \mu_i^{'} \neq 0 \\ (1/2)log(2\sigma_i^{'2} + 1) & \mu_i^{'} = 0 \end{cases} \quad i = 1, \ldots, k \quad (8.5)$$

$$\rho_{i,j} = (1/(\sigma_i\sigma_j))\, log\left((B + \sqrt{B^2 + 4AC})/2A\right) \quad i, j = 1, \ldots, k \quad (8.6)$$

where

$$A = cosh(\mu_i + \mu_j) \quad (8.7)$$

$$B = 2\rho_{i,j}^{'}\sigma_i^{'}\sigma_j^{'}exp\left[-(\sigma_i^{'2} + \sigma_j^{'2})/2\right] + 2sinh(\mu_i)sinh(\mu_i) \quad (8.8)$$

$$C = cosh(\mu_i - \mu_j) \quad (8.9)$$

As logarithmic transforms of normally distributed data produce negative skewed and hyperbolic sine transforms produce positive skewed and kurtotic distributions, these equations are a practical solution to the problem of simulating data with very different distributions and the same covariance structure. The simulation reported by Hulland, Ryan and Rayner [54] for structural equation modelling by PLS does include distribution as a factor by transforming the data. The transformation used was a univariate power method from Fleishmann [34] which does not account for the change in the multivariate covariance structure from the transform. This simulation concluded that the form of the data distribution had no effect on the PSL model, but it is not clear how valid this conclusion might be due to confounding of distribution and covariance factors. Further, it is also not clear if the anecdotal evidence for normalising PLS data is really due to the change in the distribution or the change in the covariance. Distribution and covariance are arranged as independent factors within this simulation, so that these issues may be resolved.

8.2.3 The Regressor Factors in the Simulation

n - Number of observations, k - Number of regressors, m - Number of responses

Published PLS studies cover a wide range of numbers of variables and observations. But these ranges are not really continuous in that the datasets tend to be either "'portrait"' or "'landscape"'. The ranges used are n observations25 to 100, with k regressors 6 to 20 for the portrait/mixtures simulation and k regressors 200 to 500 for the landscape simulations. For PLS2, m responses 5 to 20.

A^* - Number of Latent Variables for perfect fit

So $(k - A^*)$ is a measure of the collinearity. The number of latent variables used in the simulation 2 to 8 for portrait and landscape, 2 to k-1 for mixtures.

VDR_X - The Regressor Variance Decay Rate

This is the decay rate in the eigenvalues of the regressor matrix $\mathbf{X}$, to fix the degree of correlation between the factors. If all k regressors were in a full factorial plan then they would have equal variances of 1. So the variances are scaled so that their sum is always equal to k. It was decided to arrange the simulation plan with three levels of decay rate, fast, medium and slow. Due to the random nature of the simulation, this categorical factor does not create three specific and unique levels of collinearity as measured by the regressor condition number, so condition number or correlation coefficients could be used as an alternative variable in the subsequent analysis.

Medium decay is linear over the first A^* terms with sum k. The scaling factor for the remaining $k - A^*$ terms is set to zero.

$$vscale = k/(A^* \times (A^* + 1)) \tag{8.10}$$

$$MediumScaleVDR_X = vscale \times (A^* : 1) \text{ otherwise } 0 \tag{8.11}$$

So with $k = 6$ regressor variables and $A^* = 4$ latent variables, $vscale = 0.3$ and the medium decay rate scaling factors are $[2.4, 1.8, 1.2, 0.6, 0.0, 0.0]$.

Fast and slow decay rate mean an increasing or decreasing change in variances between latent variables. Having a fixed ratio in the incremental variance implies that the variances are a geometric series. Fast decay is a geometric series where the variance reduces by a third with each LV. The sum of the series 1,3,9,27,... for A^* terms is $(3^{A^*}-1)/2$. So reversing the series order and multiplying by a scale factor of $2k/(3^{A^*}-1)$ gives a rapidly decreasing sequence of A^* terms with sum k.

$$vscale = 2k/(3^{A^*}-1) \tag{8.12}$$

$$FastScaleVDR_X = vscale \times 3^{(A^*-1:0)} \text{ otherwise } 0 \tag{8.13}$$

So with $k = 6$ regressor variables and $A^* = 4$ latent variables again, $vscale = 0.15$ and the fast decay rate scaling factors are $[4.05, 1.35, 0.45, 0.15, 0.0, 0.0]$ which sum to 6.

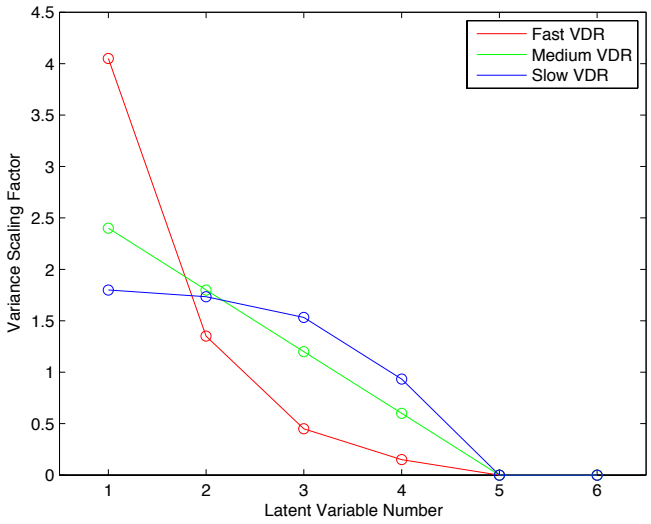

Figure 8.1: Regressor Variance Decays Rates for 6 Regressors and 4 Latent Variables

Slow decay is a geometric series where the difference in the variances reduces by a third between each latent variable, again with sum k.

$$vseq_i = 3^{(A^*-1)} - \Sigma_{i=0}^{i=A^*-2} 3^i \tag{8.14}$$

$$vscale_j = k/\Sigma_{i=1}^{i=j} vseq_i \tag{8.15}$$

$$SlowScaleVDR_X = vscale \times 3^{(A^*-1:0)} \text{ otherwise } 0 \tag{8.16}$$

So with $k = 6$ regressor variables and $A^* = 4$ latent variables again, $vseq = [27, 26, 23, 14]$ and $vscale = 6/90$ and the fast decay rate scaling factors are $[1.8, 1.7333, 1.5333, 0.$

which sum to 6. These Regressor Variance Decay Rates are compared as Figure 8.2.3.

SNR_X - The Regressor Error Levels

SNR_X and SNR_Y as degree of fit errors in regressors and responses. Due to the random basis of the simulation samples, the degree of regressor and response variation between samples is not constant. So to maintain the independence of the regressor and response error levels from the other factors the error levels have been defined in terms of signal to noise levels.

For SNR_X let σ_X be the overall standard deviation of the regressor matrix $\mathbf{X}$ expressed as a vector. Then

$$\sigma_{E_X} = \sigma_X / SNR_X \tag{8.17}$$

$$E_X = \mathcal{N}_k(0, diag(\sigma_{E_X})) \tag{8.18}$$

Where $\mathcal{N}_k$ refers a multivariate normal distribution of k dimensions. So the columns of E_X are all independent and identically distributed. The range of both SNR_X and SNR_Y in the simulation 10 to 100.

One way of looking at this (Kiers and Smilde's [61]) is that $\mathbf{X}$ and $\mathbf{Y}$ represent the population values, $X + E_X$ and $Y + E_Y$ represent the sample.

8.2.4 Internal Regression Coefficients in the Simulation

PLS1 Regression Coefficients

For PLS1 univariate responses and with the response vector as any linear function of R, such as

$$\mathbf{Y} = \mathbf{R}b \tag{8.19}$$

where $b \in \mathbb{R}^{A^* \times 1}$ then $\mathbf{Y}$ will be in the row space of $\mathbf{R}$ and so $\mathbf{X}$. So $\mathbf{Y}$ must be an exact solution for PLS at A^* latent variables.

Since

$$\beta = \mathbf{X}^{+}\mathbf{Y} \tag{8.20}$$

$$= (\mathbf{R}\xi^{T})^{+}\mathbf{R}b \tag{8.21}$$

then

$$\mathbf{R}\xi^{T}\beta = \mathbf{R}b \tag{8.22}$$

so

$$b = \xi^{T}\beta \tag{8.23}$$

The coefficients in the PLS regression β are related to the "internal coefficients" b by $b = \xi^{T}\beta$. As b is of length A^{*} and β of length k and $k > A^{*}$ then b can be calculated directly from β but in generally not β from b.

These internal regression coefficients in PLS1 are generated sequentially so are uniformly distributed. This means that the Y response samples are essentially linear transforms of the regressors X so inherit their distribution directly without influence from the internal coefficient distribution. For PLS1 the internal regression samples are characterised by a single factor - PIR, the Pattern in the Internal Regression defined later in section 8.2.5.

PLS2 Regression Coefficients

For PLS2 there are 3 sets of correlations/covariances - within-X, within-Y and between X and Y. But these cannot be fixed independently for simulation as fixing within-X and between X and Y fixes within-Y. The simulation method used here for PLS2 follows the PLS1 method of fixing X and the internal regression coefficients with some the following additional factors.

The regressor simulation samples for PLS2 are generated exactly as those for PLS1. For PLS2, the multivariate nature of the response introduces additional factors for collinearity and correlation within the responses.

The requirement for uniformly distributed columns presents a particular difficulty for the PLS2 simulation. While normally distributed columns maintain their distribution under the linear transforms required to set specific correlations and collinearity, in general uniform distributed columns do not maintain their distribution under linear transforms. Unlike the lognormal and sinh transforms used for the regressor simulation samples, no closed form for a uniform transform is known that also preserves the covariance. Hotelling and Pabst [52] give a transform that preserves the Spearman correlation. The covariance matrix and (Pearson) correlation matrix are the same apart from the scaling of the columns. In practice, the Pearson and Spearman correlation are generally quite close, but there is no direct relation between the two as Spearman correlation is a nonparametric method based on rank. More complex methods that preserve the Pearson correlation or covariance are known such as the iterative method of Li and Hammond[67]. While the set covariance generates the samples, the actual sample covariance for each sample is used in the analysis. So small differences due to from Hotelling Pabst transformation method make no practical difference for the simulation.

Transformations for Multivariate Uniform Samples

Let $\mathbf{\Sigma}$ be a covariance matrix from standardised random variables with normal distributions. As these variables are standardised, the elements of $\mathbf{\Sigma}$ are also the correlation coefficients.

If $\mathbf{\Sigma}$ are the Spearman correlation coefficients between the standardised random variables, then from Hotelling and Pabst [52].

$$\mathbf{\Sigma}^{adj} = 2sin(\frac{\pi}{6}\mathbf{\Sigma}) \tag{8.24}$$

$$\mathbf{Z} = \mathcal{N}(0, \mathbf{\Sigma}^{adj}_{i,j}) \tag{8.25}$$

$$\mathbf{U} = \Phi(\mathbf{Z}) \tag{8.26}$$

where the columns of $\mathbf{U}$ are uniformly distributed with Pearson correlation coefficients $\mathbf{\Sigma}$.

The only problem with this transformation is that it starts from the Spearman not

Pearson correlation coefficients that could then be directly related to the covariance. But it can be shown[19] that the maximum absolute difference between Pearson and Spearman correlation coefficients after the transformation is less than 0.02.

8.2.5 The Response Factors in the Simulation

PIR - The Pattern in the Internal Regression

The PIR factor in the simulation determines the "Pattern in the Internal Regression", which is essentially a weighting factor to determine how the response is related to the high variance and low variance regressors.

The vector b can be thought of as a weighting for the columns of $\mathbf{R}$ that fix the response $\mathbf{Y}$. In the simulation, the columns of R are ordered from high variance to low variance. So high values at the start of the vector b are related to these high variance columns. In effect, this is changing the relation of the response from highly correlated regressors towards less highly correlated regressors. A 3 level weighting scheme has been used, based on a linear weighting with sum(abs(b))=A^* and sum(b)=0.

For High level of PIR, the internal coefficients b are weighted towards the high variance regressor columns.

$$bseq_i = (A^* + 1 - i)(-1)^{(i-1)} \tag{8.27}$$

$$bseq = bseq - mean(bseq) \tag{8.28}$$

$$bscale = A^*/sum(abs(bseq)) \tag{8.29}$$

$$HighScalePIR = bscale * bseq \tag{8.30}$$

So for $A^* = 4$ then $PIR = [1.4, -1.4, 0.6, -0.6]$ and sum is zero.

For Medium level of PIR, equal weights given to all regressor columns.

$$bseq_i = (-1)^i \tag{8.31}$$

$$MediumScalePIR = bseq - mean(bseq) \tag{8.32}$$

$$\tag{8.33}$$

So for $A^* = 4$ then $PIR = [1, -1, 1, -1]$ and sum is zero.

For Low level of PIR, the internal coefficients b are weighted against the high variance regressor columns.

$$bseq_i = (i)(-1)^{(i-1)} \tag{8.34}$$

$$bseq = bseq - mean(bseq) \tag{8.35}$$

$$bscale = A^*/sum(abs(bseq)) \tag{8.36}$$

$$LowScalePIR = bscale \times bseq \tag{8.37}$$

So for $A^* = 4$ then $PIR = [-0.6, 0.6, -1.4, 1.4]$ and sum is zero.

For the PLS1 simulation, the elements of the internal regression coefficient vector b are ordered by scale so that the PIR (Pattern in Internal Regression) controls how the response is related to the highly correlated or weakly correlated regressors. The same basic method has been used for PLS2 where the rows of the internal regression coefficient matrix B are ordered by absolute size with or against the highly correlated regressors.

VDR_Y - The Response Variance Decay Rate

This is only applicable for the PLS2 simulation and is analogous to VDR_X, the Regressor Variance Decay Rate. VDR_Y determines the decay rate in the eigenvalues of the response matrix $\mathbf{Y}$, to fix the degree of correlation between the factors. If all m responses were in a full factorial plan then they would have equal variances of 1. So the variances are scaled so that their sum is always equal to m. Just as VDR_X, VDR_Y is arranged in the simulation plan with three levels of decay rate fast, medium and slow.

Medium decay is linear decay over m terms with sum m.

$$vscale = 2m/(m(m+1))) \tag{8.38}$$

$$MediumScaleVDR_Y = vscale \times (m : 1) \tag{8.39}$$

So with $m = 4$ response variables, $vscale = 0.4$ and the medium decay rate scaling factors are $[1.6, 1.2, 0.8, 0.4]$.

Fast and slow decay rate mean an increasing or decreasing change in variances between latent variables, so are generated from geometric series that sum to m. For the Fast decay rate

$$vscale = 2m/(3^m - 1) \tag{8.40}$$

$$FastScaleVDR_Y = vscale \times 3^{(m-1:0)} \tag{8.41}$$

So with $m = 4$ response variables again, $vscale = 0.1$ and the fast decay rate scaling factors are $[2.7, 0.9, 0.3, 0.1]$ which sum to 4.

For the Slow decay rate

$$vseq_i = 3^{(m-1)} - \Sigma_{i=0}^{i=m-2} 3^i \tag{8.42}$$

$$vscale_j = k/\Sigma_{i=1}^{i=j} vseq_i \tag{8.43}$$

$$SlowScaleVDR_Y = vscale \times 3^{(m-1:0)} \tag{8.44}$$

So with $m = 4$ response variables again, $vseq = [27, 26, 23, 14]$ and $vscale = 0.0444$ and the fast decay rate scaling factors are $[1.2000, 1.1556, 1.0222, 0.6222]$ which sum to 4. These Response Variance Decay Rates are compared as Figure 8.2.5.

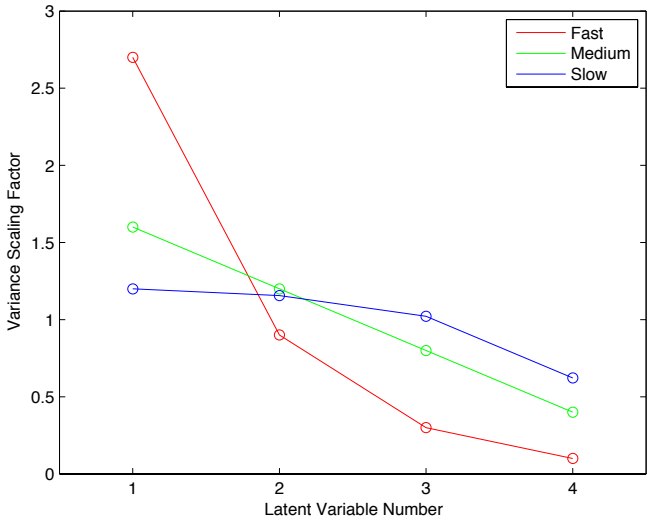

Figure 8.2: Response Variance Decays Rates for 4 Responses

RCF **- The Response Correlation Factor**

This is only applicable for the PLS2 simulation. The *RCF* is a three level categorical factor in the simulation that determines the degree of the correlation within the internal coefficient covariance matrix. For High levels of *RCF*, the off diagonal elements of the internal coefficient covariance matrix are uniformly distributed between -0.75 and +0.75. For Medium levels of *RCF* the elements are distributed between -0.50 and +0.50 and between -0.25 and +0.25 for Low levels of *RCF*.

The diagonal elements of the internal coefficient matrix are the scaling vector determined by the VDR_Y factor. While this is sufficient to form a symmetric matrix it is not necessarily positive definite, so the nearest positive definite matrix used for the simulation sample calculation.

s **- The Response matrix rank**

This is only applicable for the PLS2 simulation. The positive definite matrix from the *RCF* calculation is of full rank. This is reduced to rank s by SVD transforms. Let

$$\mathcal{U}diag(\mathcal{S})\mathcal{V}^T = SVD(\mathbf{B}_{cov}) \tag{8.45}$$

where $\mathbf{B}_{cov}$ is the full rank covariance matrix of the internal correlation coefficients of dimension $m \times m$. To reduce the rank from m to s, set the elements of the vector $\mathcal{S}$ from $s+1$ to m to zero. Then

$$\mathbf{B}'_{cov} = \mathcal{U}diag(\mathcal{S})\mathcal{V}^T \tag{8.46}$$

and $\mathbf{B}'_{cov}$ is an approximation to the original covariance matrix with rank s.

SNR_Y **- The Response Error Levels**

For PLS1 simply added to the response vector. For PLS2 structured identically to the Regressor Error Levels from equations 8.17 and 8.18 so that the response errors are also independent and identically distributed.

8.2.6 Summary of the Simulation Calculation

Starting from the regressor factors n,k,A^*,VDR_X,SNR_X and *Distribution*.

- Calculate the regressor variance decay scaling vector v from k, A^* and the categorical factor VDR_X, using equations 8.10 to 8.16.

- Generate a sample matrix $\mathbf{R}$ with n rows and A^* independent columns of the specified distribution. The variance of the columns of $\mathbf{R}$ is given by v.

  For a sample matrix $\mathbf{R}$ with columns with normal distributions, let

  $$\mathbf{R} = (N)_k(0, \Sigma = diag(VDR)) \tag{8.47}$$

  where $(N)_k$ is the multivariate normal distribution.

  For a sample matrix $\mathbf{R}$ with columns with lognormal distributions, let

  $$\mathbf{R} = (LN)_k(0, \Sigma = diag(VDR)) \tag{8.48}$$

  where $(LN)_k$ is defined by equations 8.1 to 8.3.

  For a sample matrix $\mathbf{R}$ with columns with sinh distributions, let

  $$\mathbf{R} = (Sinh)_k(0, \Sigma = diag(VDR)) \tag{8.49}$$

  where $(Sinh)_k$ is defined by equations 8.4 to 8.9.

- Generate a random orthonormal matrix η with k rows and A^* columns. There are various ways to do this. The method used in the simulation is to calculate the covariance af a matrix of k uniformly distributed columns. The first A^* eigenvectors of this covariance matrix are a random orthonormal matrix of k rows by k columns.

- The base regressor matrix $\mathbf{X}_{base}$ is then $\mathbf{R}\eta^T$

- Calculate the regressor random error level σ_X from the overall variance the elements of $\mathbf{X}_{base}$ and the specified signal to noise ratio SNR_X, using equation 8.17.

- Calculate the regressor random error values E_X with n columns and k columns as independent multivariate normal samples with standard deviation σ_X, using equation 8.18.

- Calculate the sample regressor matrix $\mathbf{X}_{sample}$ from $\mathbf{X}_{base} + \mathbf{E}_X$.

For PLS1 univariate responses

- Using the Pattern in the Internal Regression factor PIR, calculate the scaling factor vector using equations 8.27 to 8.37.

- From A^* and PIR calculate the internal regression vector b. For PLS1 the internal regression vector is the PIR scaling factor from equations 8.27 to 8.37.

- Rescale b so that the responses are in the range -1 to +1.

- Calculate the base response vector y_{base} from $\mathbf{X}_{base}b$.

For PLS2 multivariate responses

- Calculate the response variance decay scaling vector v from m and the categorical factor VDR_Y using equations 8.38 to 8.44.

- Calculate the range of the elements of the coefficient covariance matrix from RCF as described in section 8.2.5.

- Generate a sample coefficient covariance matrix $\mathbf{B}_{cov}$ of dimension $m \times m$ with the off-diagonal elements uniformly distributed within the specified range and diagonal elements v from VDR_Y.

- Find the nearest positive definitive matrix to $\mathbf{B}_{cov}$. The simulation uses the nearPD function in the R Matrix package.

- Calculate the corresponding correlation matrix $\mathbf{B}_{corr}$. This is a straightforward scaling.

- Transform B_{corr} for generating uniform variates using equations 8.24 to 8.26.

- Generate a sample matrix $\mathbf{B}$ of size k rows by m columns with uniformly distributed elements and correlation matrix $\mathbf{B}_{corr}$.

- Scale the range of the uniformly distributed columns of B to match the standard deviations associated with $\mathbf{B}_{cov}$.
- Reduce rank of $\mathbf{B}$ from m to s by singular value decomposition, equations 8.45 to 8.46.
- Sort the rows of $\mathbf{B}$ according to the PIR factor. The actual scaling vector used for the row sort is calculated from equations 8.27 to 8.37.
- Calculate the base response matrix $\mathbf{Y}_{base}$ from $\mathbf{X}_{base}\mathbf{B}$.
- Calculate the response random error level σ_Y from the overall variance the elements of $\mathbf{Y}_{base}$ and the specified signal to noise ratio SNR_Y.
- Calculate the regressor random error values $\mathbf{E}_Y$ with n columns and m columns as independent multivariate normal samples with standard deviation σ_Y
- Calculate the sample regressor matrix $\mathbf{Y}_{sample}$ from $\mathbf{Y}_{base} + \mathbf{E}_Y$.

8.2.7 Summary of the Properties of the Internal Regression Coefficients and Responses

The properties of the internal regression coefficients $\mathbf{B}$ are then

- Dimensions $A^* \times m$
- Column means are zero.
- The columns are uniformly distributed.
- Rank s in the range 1 to $min(A^*, m)$
- The covariance is approximately $\mathbf{B}_{cov}$.

The properties of the response matrix $\mathbf{Y}$ are then

- Dimensions $n \times m$
- Column means are zero.
- The distribution of the columns of $\mathbf{Y}$ are that of the columns of $\mathbf{X}$.

- Rank s in the range 1 to $min(A^*, m)$
- The columns of $\mathbf{Y}$ are all in the columns space of $\mathbf{X}$.

8.3 Experimental Designs for the Simulation

A thorough assessment of the performance of latent variable selection methods or more generally the overall performance of PLS as an algorithm implies that any particular sensitivity to any design factor can be identified. The literature on PLS does not generally identify any specific sensitivity. But it is quite possible that some combinations of factors in PLS could have particularly advantageous or adverse effects that are not generally recognised. If so, these should be apparent as interactions between design factors. Consequently, the details of the experimental design are important here as the experimental design should be particularly powerful for identifying interactions. If this was a physical experiment response surface designs would be appropriate as they can be in some way optimal for both main linear effects and quadratic curvature. Computer experiment have no issues with trial blocking or replication that are so important for physical experiments, so space filling designs are often used as they are particularly efficient at identifying the nature of response curvature. Consequently, an combination of response surface and space filling designs called a hybrid design by Johnson, Montgomery and Kennedy[56] has been used for this simulation.

8.3.1 The Simulation Factors and Ranges

From the previous section on the simulation methods, it follows that there are 8 design factors in the PLS1 simulation and an additional 4 to characterise the response the response for PLS2. These factors and the ranges selected for this simulation are shown as Table 8.1.

8.3.2 Design Generation

As PLS1 and PLS2 involve different sets of factors and generally different algorithms, these different cases have been studied in two separate but comparable simulations. In both PLS1 and PLS2 simulations, portrait and mixture datasets have the same factors

Factor	Simulation	Minimum	Maximum
1 Number of Observations, n	PLS1+PLS2	25	100
2 Number of Regressors, k	PLS1+PLS2	6	512
Portrait and Mixture	PLS1+PLS2	6	25
Landscape	PLS1+PLS2	25	512
3 Number of Responses, m	PLS2	4	15
4 "True" number of Latent Variables, A^*	PLS1+PLS2	2	9
5 Regressor Signal-to-Noise ratio, SNR_X	PLS1+PLS2	10	100
6 Response Signal-to-Noise ratio, SNR_Y	PLS2	10	100
7 Regressor Variance Decay Rate, VDR_X	PLS1+PLS2	Slow Medium Fast	
8 Response Variance Decay Rate, VDR_Y	PLS2	Slow Medium Fast	
9 Pattern in Internal Regression	PLS1+PLS2	Low Medium High	
10 Data Distribution	PLS1+PLS2	Normal LogNormal Sinh	
11 Response Matrix Rank, s	PLS2	2	9
12 Response Correlation Factor, RCF	PLS2	Slow Medium Fast	

Table 8.1: PLS Simulation Design Factors and Ranges

but different ranges to Landscape datasets so these variations have been combined into the same simulation.

The simulation designs were both developed by combining and augmenting a series of designs for specific characteristics of datasets used for PLS models. Apart from the number of regressors, the space for the the portrait design is enclosed within that of the landscape dataset. Consequently, it was arranged that the portrait design be developed first then augmented to a landscape design so that the simulation datasets could be analysed separately or in combination.

The design started with an algorithmic D-optimal (quadratic) response surface design over the factors and ranges shown in Table 8.1. This base design was intended to cover the portrait design space and not landscape so was constrained to $n > k + 1$. One other constraint was applied to the PLS1 design and two further for PLS2 to ensure that all trials were logically consistent for multivariate datasets.
For PLS1

$$n > k + 1 \tag{8.50}$$

$$A^* \leq k \tag{8.51}$$

$$\tag{8.52}$$

Additional constraints for PLS2

$$s \leq m \tag{8.53}$$

$$s \leq A^* \tag{8.54}$$

For the PLS1 designs with 8 factors, there are 69 terms in the full model and it was found that a 100 trial design could give a design with high G-efficiency but was not that strong for prediction variance or coefficient power. For the PLS2 designs with 12 factors, there are 156 terms in the full model and it was found that 200 trials were required to give a comparably efficient design.

The second stage was to add an independent space filling design with the same number of trials over the same portrait/mixture design space. These space filling designs were also subject to the relevant constraints in equations 8.50 to 8.54. These space filling designs were generated as latin hypercube designs across the entire design space by exclusion sampling, with the total number of trials adjusted by trial and error until the required number of trials was achieved within the constrained design region. The categorical factors in the design were set by mapping the levels to equal regions in the design space. As the response surface designs were generated to a D-optimal criterion, all the trial points were located at corners or on edges of the design space. Consequently, adding the space filling trials that are all inside the design space boundaries makes no change to the design space determinant. So it makes no practical difference if the D-optimal response surface design is generated first or augmented from the space filling design.

The third stage was to extend the design space to higher numbers of regressors to simulate landscape datasets. It was found that an additional 50 trials was sufficient for PLS1 and 100 additional trails were required for PLS2. Finally an additional number of trials equal to the D-optimal augmentation were included as a space filling design over the landscape extension region. This strategy produced a 300 trial design for PLS1 and a 600 trial design for PLS2 that were strong in both landscape and portrait spaces. Table 8.2 shows the final characteristics of the designs have good prediction variance. and coefficient resolution power. During the sequential development of these

designs, it is apparent that the numerical G-efficiency and D-efficiency of the designs decreased. As prediction variance and coefficient power was increasing, this reduction in efficiency must be due to more trials being included in the design than are strictly necessary.

Design	n	Prediction Var		G-Efficiency	D-efficiency
		Max	Avg		
PLS1	300	0.65	0.19	69.1%	54.2%
PLS2	600	0.67	0.24	72.0%	49.6%

Table 8.2: PLS1 and PLS2 Design Efficiency Summary

As each design trial setting is used to generate a randomised PLS model, every design trial was replicated so that an assessment of within-trial variation was available for the analysis. Thirty replicates were used for every trial PLS1 trial, giving 9000 samples in the PLS1 simulation. Twenty replicates were used for every PLS2 trial, giving 12000 samples in the PLS2 simulation.

The simulation methods described previously in section 8.2.6 were coded in R with the pls model fitting from the R pls package[82]. This uses the kernel PLS method which is numerically equivalent to NIPALS. Multiple computers with concurrent R sessions were used for the simulation calculations. The total computing time was equivalent to around 8,500 hours for a single R session.

8.3.3 Validating the Simulation Factor Settings to the Example Datasets

The simulation designs have been set up with some factor levels apparently fixed arbitarily or simply what might appear to be reasonable values. Tables 8.3 and 8.4 compare the parameters of the example datasets discussed in previous chapters to the ranges of those parameters that occur in the simulation samples. These statistics for the example datasets are from centred and scaled versions. Statistics for the simulation samples are for fit datasets including added random errors as these are equivalent scaling to the simulation datasets.. Clearly it is not possible to determine

absolutely parameters such as variance decay rate or signal to noise ratio for the example datasets. But the set of parameters chosen for these comparisons are closely related to and characterise the multivariate structure of both example and simulation datasets.

	WineAroma	Gasoline	WasteGlass	Simulation Samples		
				5% Quartile	Median	95% Quartile
X Median Correlation Coefficient	0.19	0.77	0.05	0.17	0.40	0.76
X ADTest Fail Rate	47%	14%	100%	0%	30%	100%
PC1 X Variation	24%	35%	17%	8%	19%	38%
nPCs for 95% X Variation	11	29	10	6	18	87
Median Skewness X Columns	0.82	0.19	1.95	-2.29	-0.07	0.67
Median Kurtosis X Columns	3.40	2.99	14.30	2.55	3.89	18.26
Y ADTest P-value	0.45 pass	3.86e-6 fail	0.048 (just) fail	0.00	0.06	0.88
Response Skewness	0.68	-0.61	0.97	-2.85	0.00	2.69
Response Kurtosis	3.05	2.13	4.06	2.29	4.23	27.96

Table 8.3: PLS1 Simulation and Example Datasets Matrix Comparison

	OliveOil	Biscuits	Abrasives	Simulation Samples		
				5% Quartile	Median	95% Quartile
X Median Correlation Coefficient	0.54	0.41	0.18	0.14	0.33	0.72
X ADTest Fail Rate	80%	22%	0%	0%	17.6%	100.0%
PC1 X Variation	59%	16%	24%	7%	17%	35%
nPCs for 95% X Variation	3	34	6	4	18	90
Median Skewness X Columns	0.62	0.16	0.39	-0.99	-0.04	0.26
Median Kurtosis X Columns	2.48	2.71	1.77	2.65	3.44	10.24
Y Median Correlation Coefficient	0.60	0.80	0.54	0.33	0.59	0.90
Y ADTest Fail Rate	50%	0%	29%	0%	4%	66%
PC1 Y Variation	64%	71%	56%	25%	42%	65%
nPCs for 95% Y Variation	3	2	8	3	8	13
Median Skewness Y Columns	-0.64	-0.08	-0.75	-1.02	0.01	1.93
Median Kurtosis Y Columns	2.05	1.95	4.56	2.48	3.74	22.01

Table 8.4: PLS2 Simulation and Example Datasets Matrix Comparison

From Tables 8.3 and 8.4 it is apparent that all the characteristic parameters of the example datasets are either within or not far outside of the 5%-95% interval of the simulation samples. Most of the exceptions concern the skewness and kurtosis. There is a suggestion from these tables that response skewness and kurtosis is more closely related to regressor matrix values in the PLS1 than PLS2 datasets.

In the simulation, the distribution factor for the samples has been set through a selection of normal, lognormal or sinh distribution for the regressor elements rather than specifically for skewness and kurtosis. Figures 8.3 and 8.4 shows how these categorical distribution factors in the simulation map to specific regions in skewness-kurtosis space.

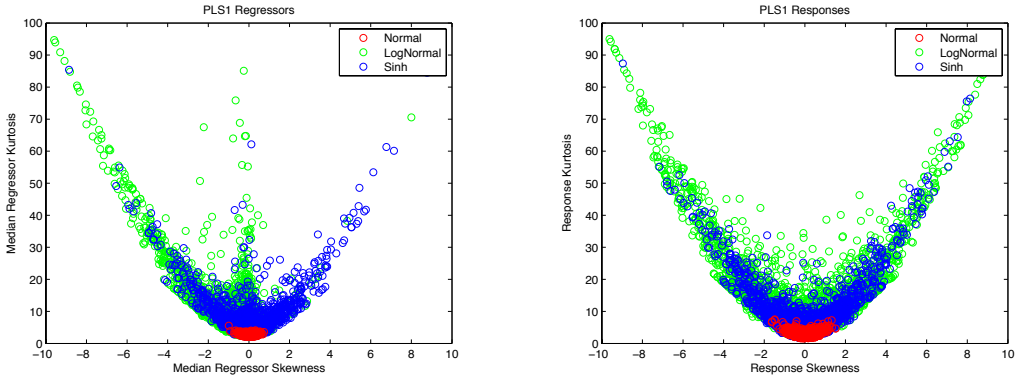

Figure 8.3: PLS1 Skewness vs. Kurtosis Plots

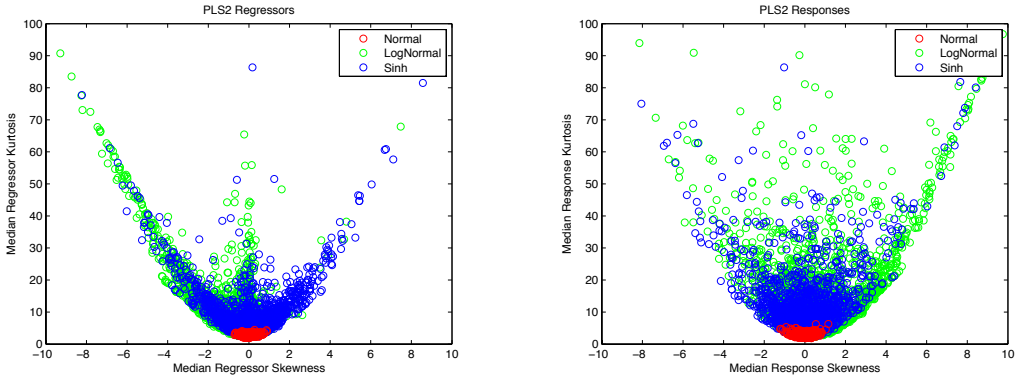

Figure 8.4: PLS2 Skewness vs. Kurtosis Plots

Consequently, it is reasonable to conclude that the distribution factor in the simulation does fulfil its objective of introducing systematic variation into the simulation distributions. If the categorical distribution factor is significant in the subsequent analysis then this is (only) evidence that PLS is sensitive to some aspect of regressor or response distribution. Identifying which aspects of the distribution that cause this sensitivity in PLS is a secondary issue and may prove problematic from from this simulation data set. From this, it is concluded that the way a small part of the example dataset skewness and kurtosis characteristics are outside of the simulation ranges is not important. Overall, it is concluded that the two experimental designs cover design spaces that are typical of PLS datasets such as the example datasets.

8.4 Simulation Analysis Methods

In order to assess the structure of the simulation data, the analysis starts with a series of plots to show the structure is related to the number of latent variables in the simulation samples. In the following sections on latent variable selection method an analysis is made to determine if the performance of any of these methods is particularly sensitive to some aspect of the simulation sample structure. The regression analysis is more concerned with comparing the relative size of the effects of the design parameters, not so much whether the probability of any small effect may be zero or not. With a strong design and many observations, even small changes could have significant p-values.

8.5 Simulation Data Model Fit Analysis

As described previously, each simulation sample has $\mathbf{X}_{base}$ and $\mathbf{Y}_{base}$ matrices that are a solution to the PLS model for the specific characteristics of the sample such as number of observations, number of regressor variables and so on. This is a solution to the PLS model in the sense that both X and Y residuals are numerically zero at the number of latent variables specified for the sample, A^*. Clearly, these residuals will not be zero for PLS models with these matrices and other numbers of latent variables.

In these base PLS models, the $\mathbf{X}_{base}$ and $\mathbf{Y}_{base}$ matrices have not been scaled. Scaling does not stop these matrices being a PLS solution for A^* latent variables, but would change the correlation structures that have been set up in the simulation samples.

As an overall summary for the simulation, box plots of the residuals RMSE and the variance and bias components for all simulation trials are shown in Figures 8.5 to 8.12 on pages 124 to 131. In this analysis each simulation trial has a specific number of latent variables fixed in the base model. To generate these plots, every sample dataset from each of the simulation trials has been fitted with a sequence of numbers of latent variables to determine how latent variable selection may influence model fit. The pivotal nature of the number of latent variables in the base model is clearly evident in most of these plots. This difference between the number of latent variables in the base model and the number of latent variables in the fitted model is shown to be an important factor in the analysis reported later in this chapter.

For the PLS1 simulation, box plots of the residuals RMSE and the variance and bias components for all simulation trials are shown in Figures 8.5 and 8.6. Residuals plots from these base matrices for individual simulation trials all show similar patterns before and after the A^* number of latent variables. So the residuals in these figures are plotted against the difference in LVs from the sample A^*. With 3000 data items, the plots in Figure 8.5 are visually dominated by the outliers so Figure 8.6 is the same data plotted on a log scale.

The observations from these plots are that the RMSE error levels are low for base models with numbers of latent variables less than the solution value A^*. Over fitted models with numbers of latent variables greater than A^* show increasing median of $\mathbf{X}_{base}$ residuals and approximately constant median of $\mathbf{Y}_{base}$ residuals with increasing latent variables. The linear plots in Figure 8.5 show that over fitting these base models can lead to very high levels of residuals. Residual standard deviation and bias components of RMSE both show similar patterns to RMSE. The last two plots in Figure 8.6 show the response residual bias on a log scale, it is clear here that bias prior to Astar is minimal and jumps after A^*.

For each simulation sample, a normally distributed random error is added to each of the base matrices $\mathbf{X}_{base}$ and $\mathbf{Y}_{base}$. Analogous linear and log plots for model fits are shown as Figures 8.7 and 8.8. It is immediately apparent that the pivotal nature of A^* in the base model plots has changed. Comparable plots for the PLS2 simulation are shown as Figures 8.10 to 8.12.

Many of the individual plots in Figures 8.5 to 8.12 show strong patterns in the outliers. For the PLS1 simulation, these patterns are most obvious in the lines of outliers in the Base RMSE X and Base Resid SD X plots in the left hand column of Figure 8.5. In the Base RMSE X plot, the three highest points at each latent variable number greater than A^* refer to the same three trials in the simulation. The same trials are outliers in the Base Resid SD X plot but are not outliers in the Base Resid Bias X plot. These three trials all have 512 regressors, 2 latent variables in the base models and distributions based on the sinh transform. Two further trials have the same combination of levels for these three factors but appear to have more typical levels of residuals.

These three outlier trials apparent in the regressor base residual have more typical residuals in the PLS1 regressor fitted residuals plots shown in the left hand column of Figure 8.7. The two lines of outliers apparent for latent variables greater than A^* in the Fit RMSE X plot and the Fit Resid X plot refer to two trials that both have low PIR and lognormal distributions. These two trials are not outliers in the Fit Resid Bias X plot. Thirty five other trials in the simulation have the same combination of levels for these two factors but have more typical levels of residuals.

Any patterns in the outliers are not so apparent in the PLS1 response residuals plots in the right hand columns of Figures 8.5 to 8.8. Only the base response residual plots in Figure 8.5 shows any strong outliers where all the maximum points in three Base RMSE Y, Base Resid SD Y and Base Resid Bias Y plots correspond to the same trial. This trial is otherwise typical and does not appear as an outlier in any of the regressor plots.

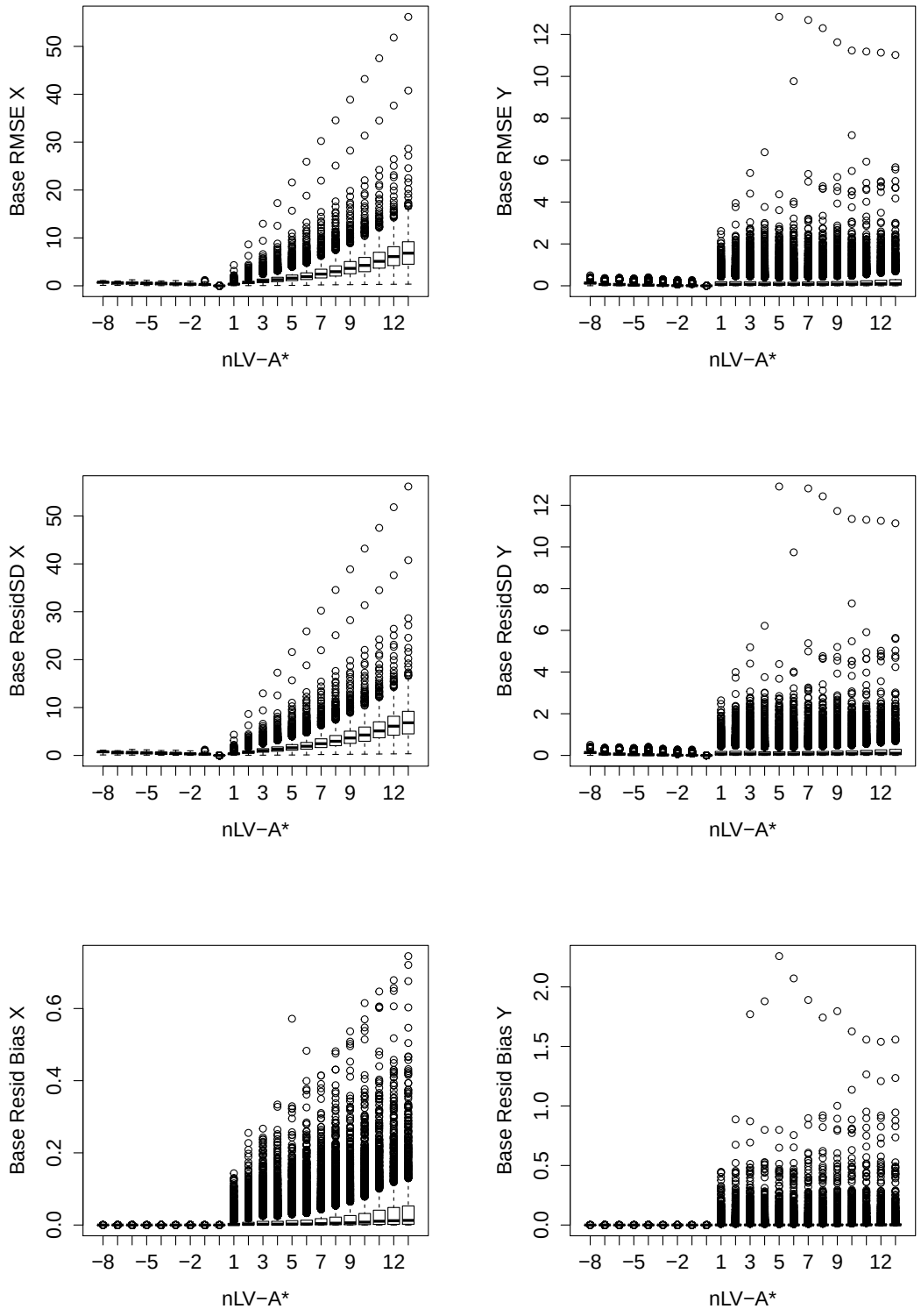

Figure 8.5: PLS1 Base Model Residuals against LV Discrepancy

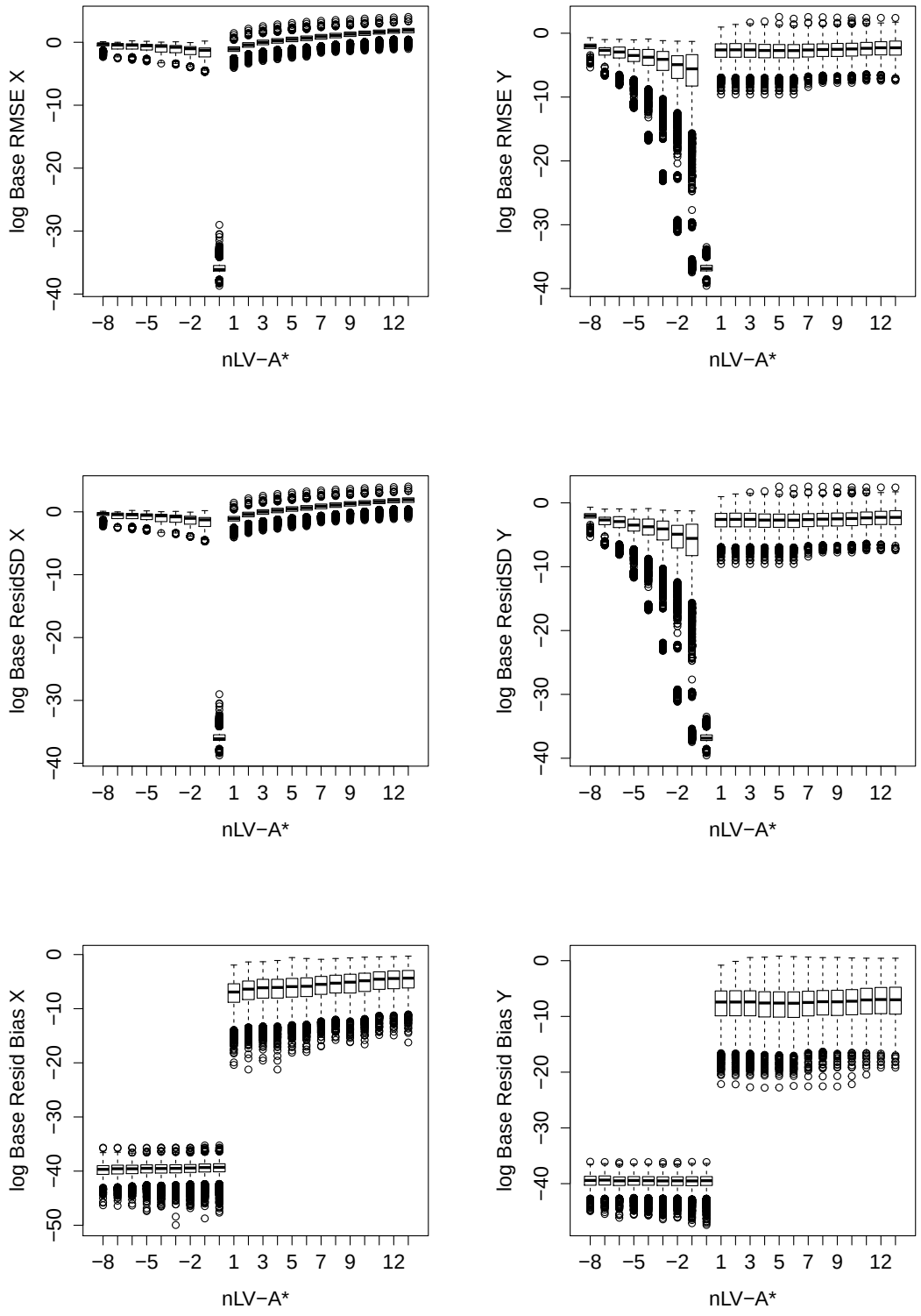

Figure 8.6: PLS1 Base Model log10 Residuals against LV Discrepancy

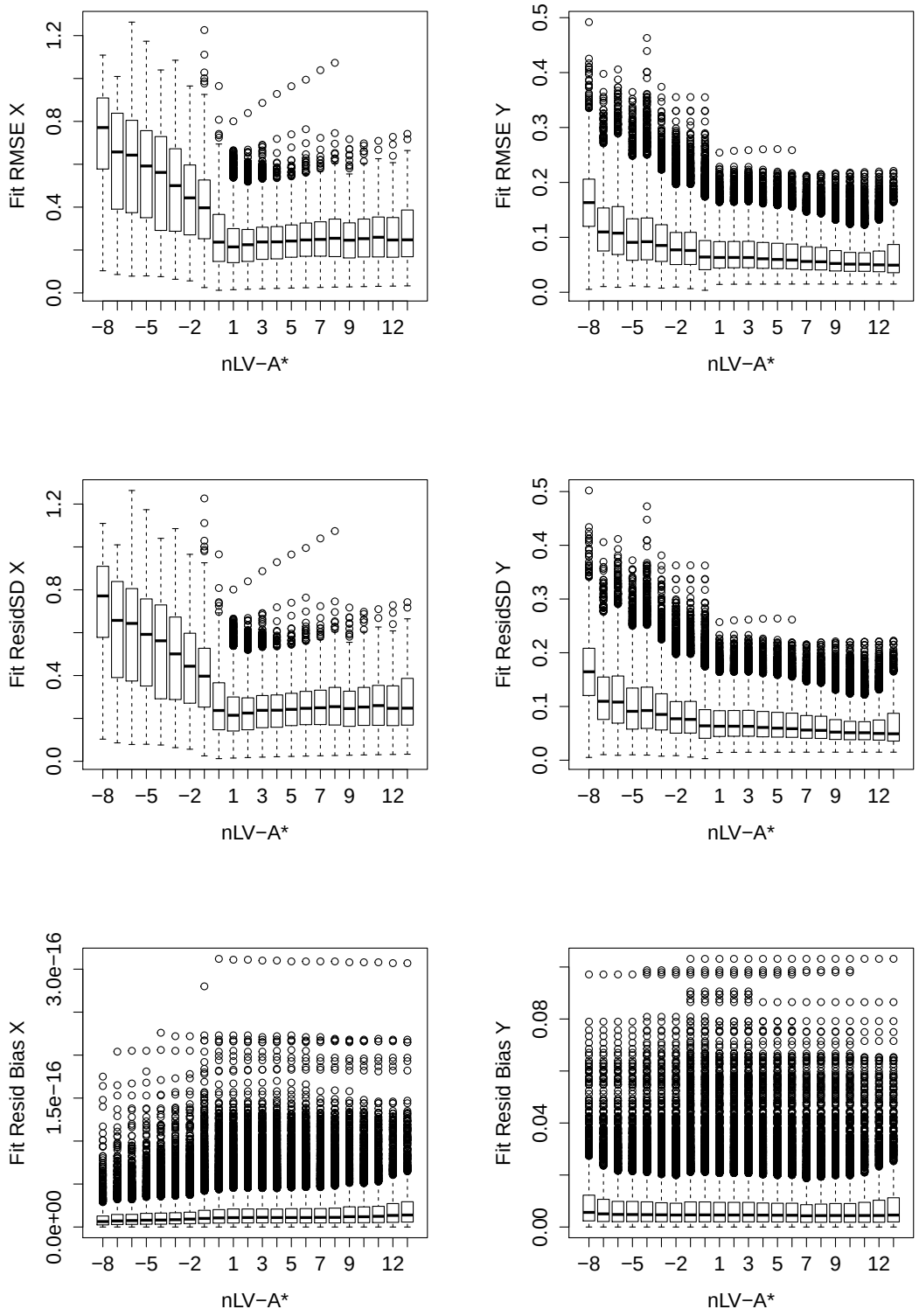

Figure 8.7: PLS1 Fitted Model Residuals against LV Discrepancy

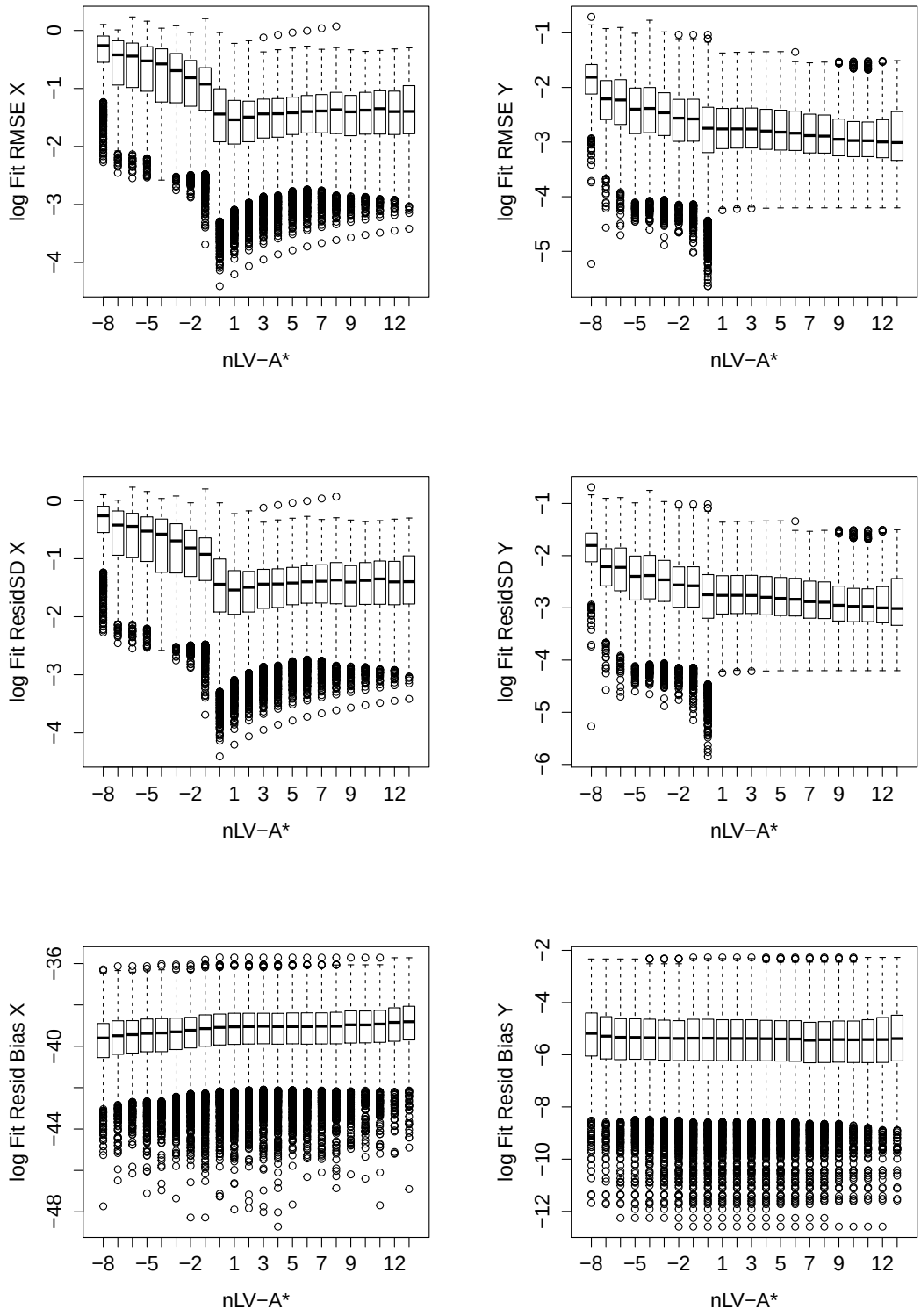

Figure 8.8: PLS1Fitted Model log Residuals against LV Discrepancy

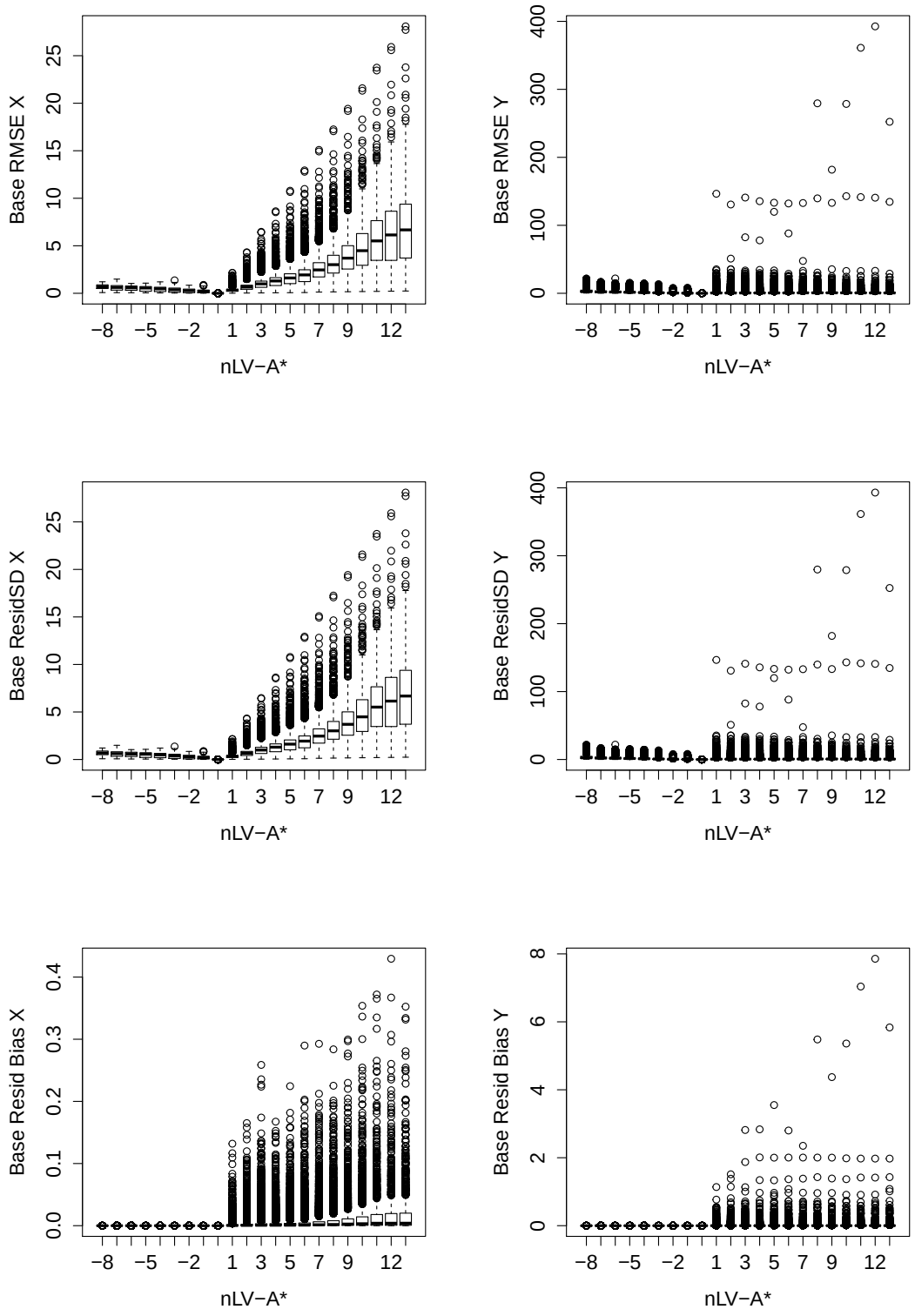

Figure 8.9: PLS2 Base Model Residuals against LV Discrepancy

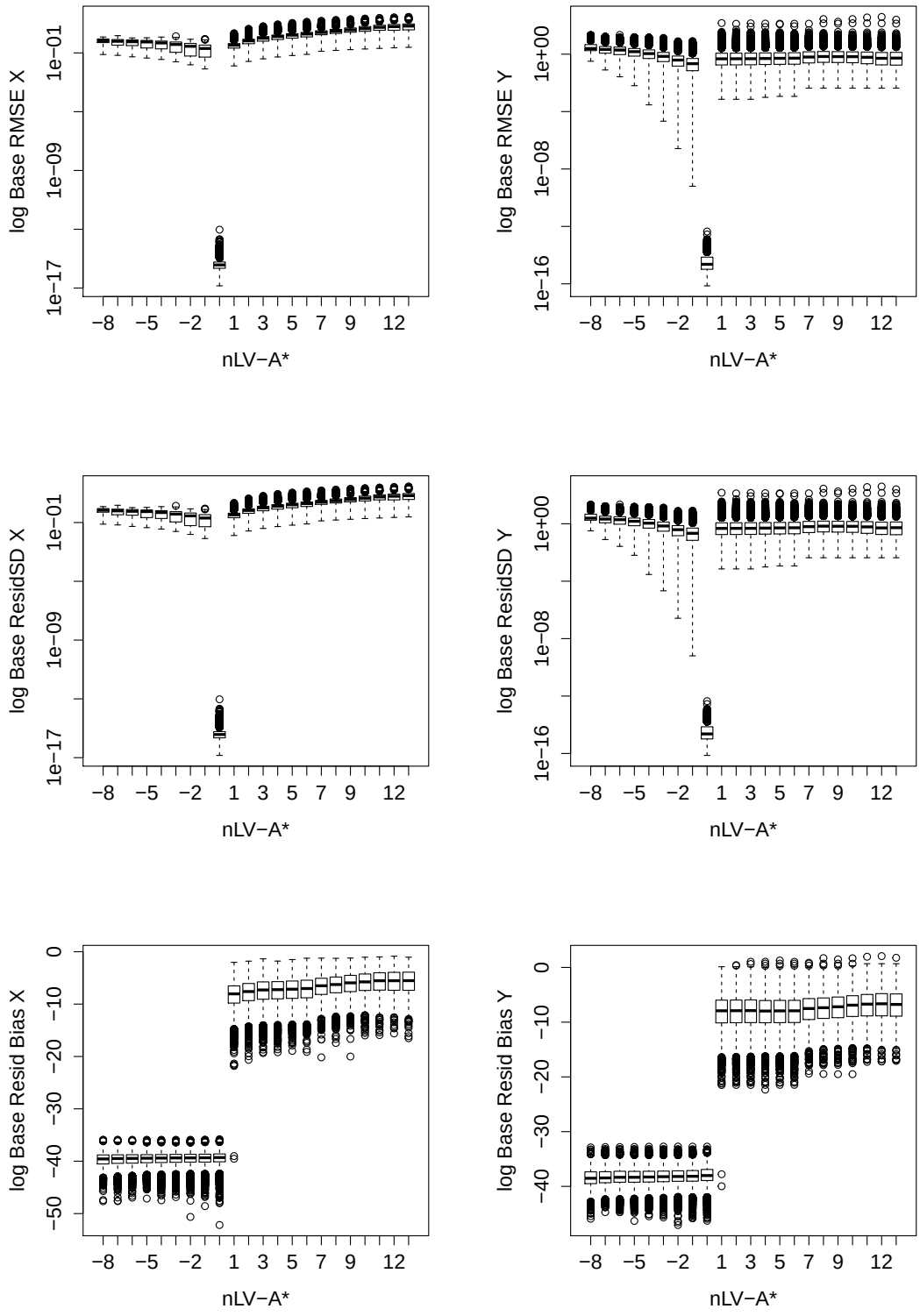

Figure 8.10: PLS2 Base Model log Residuals against LV Discrepancy

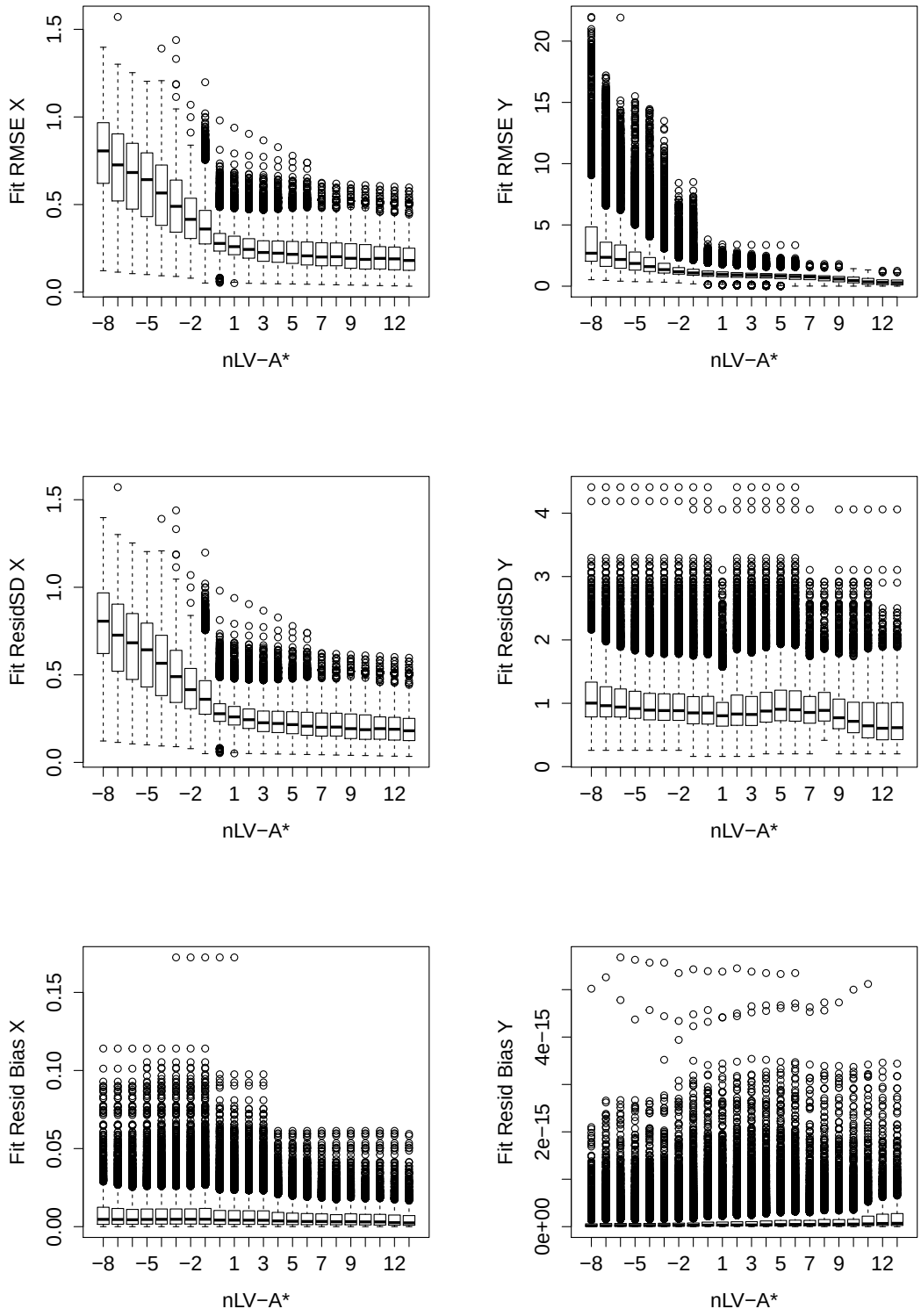

Figure 8.11: PLS2 Fitted Model Residuals against LV Discrepancy

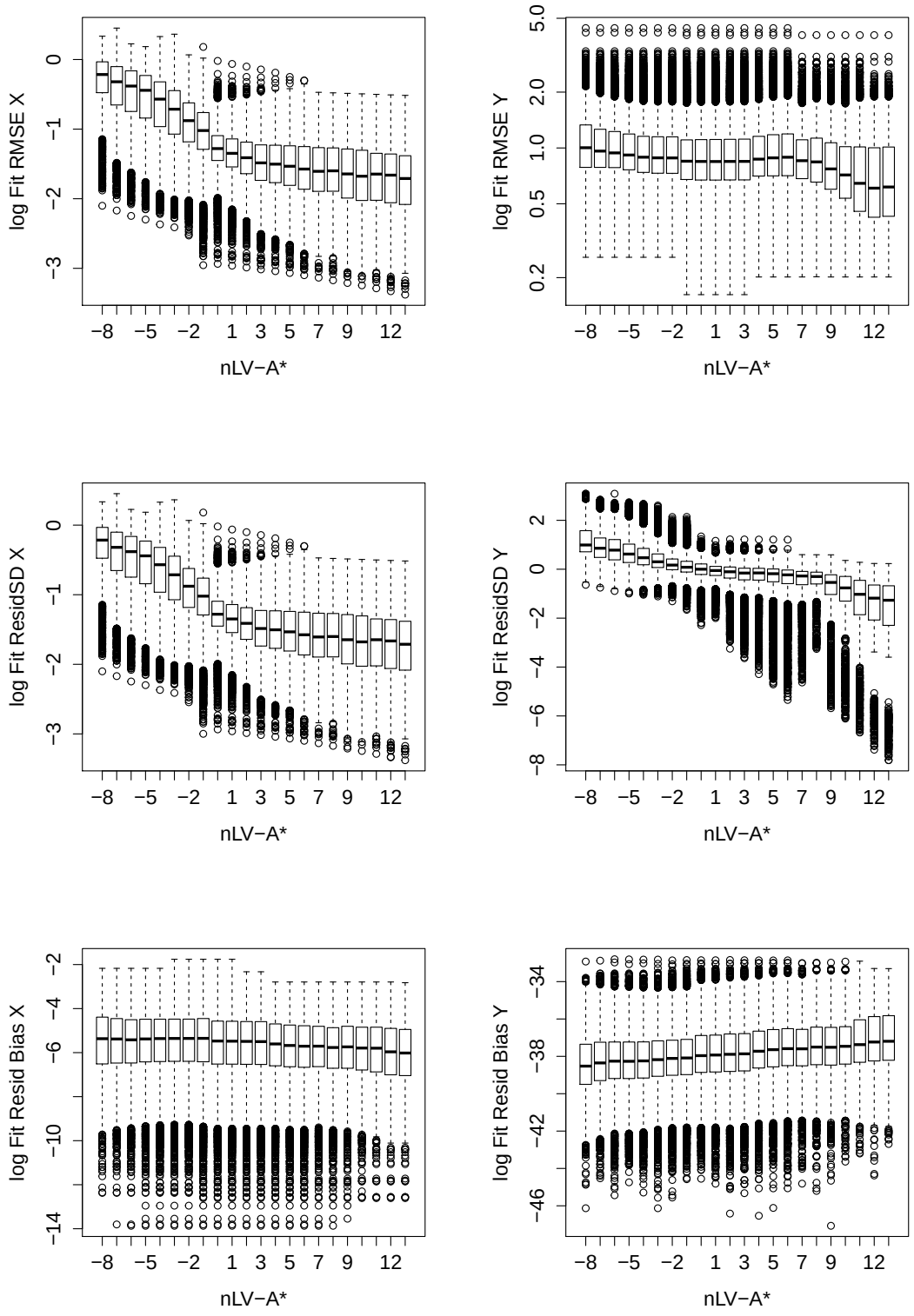

Figure 8.12: PLS2 Fitted Model log Residuals against LV Discrepancy

Lines of residual outliers are also apparent in the PLS2 regressor base residuals plots in the left hand column of Figure 8.9. The top three lines of outliers in the Base RMSE X and Base Resid SD X plots correspond to the same three simulation trials, but these three trials do not appear as outliers in the Base Resid Bias X plot. The top two outliers trials here have 512 regressor variables and 2 latent variables in the base models just as the PLS1 regressor base outliers, but the third highest outlier in this PLS2 plot has only 25 regressor variables. All three outliers in this PLS2 plot have 2 latent variables in the base models and response matrices with 4 variables of rank 2. Out of all 600 simulation trails, 41 share the same levels for these three factors but do not appear to be atypical in their regressor residuals.

The PLS2 regressor residuals fit plots on the left hand column of Figure 8.11 and all three sets of PLS2 response residuals plots in the right hand columns of Figures 8.9 to 8.12 do not show any systematic patterns involving more than one trial. None of the trials that are are maximum value outliers in these plots are also outliers in any of the PLS2 regressor base residuals plots.

From this inspection of the patterns in the residuals it is noted that while the outliers may have some factor settings in common, other simulation trials with the same factor settings do not appear as outliers. Consequently, these patterns in the outliers do not show any diagnostic potential for subsequent analysis. This may be a consequence of the statistically balanced design used for the simulation plan.

		PLS1		PLS2	
	Fit Statistic	Base	Fit	Base	Fit
$nLV < A^*$	$RMSE_X$, $ResidSD_X$	≈ 0	Decreasing	≈ 0	Decreasing
	$RMSE_Y$, $ResidSD_Y$	Decreasing	Decreasing	Decreasing	Decreasing
	$ResidualBias_X$	≈ 0	Very Low	≈ 0	Very Low
	$ResidualBias_Y$	≈ 0	Constant	≈ 0	Constant
$nLV = A^*$	$RMSE_X$, $ResidSD_X$	Clear minimum	Minimum at $A^* + 1$?	No clear Minimum	No clear minimum
	$RMSE_Y$, $ResidSD_Y$	Clear minimum	No clear minimum	No clear Minimum	No clear minimum
	$ResidualBias_X$	≈ 0	Very Low	≈ 0	Very Low
	$ResidualBias_y$	≈ 0	Constant	≈ 0	Constant
$nLV > A^*$	$RMSE_X$, $ResidSD_X$	Increasing	Decreasing	Increasing	Decreasing
	$RMSE_Y$, $ResidSD_Y$	$\approx$ Constant	Decreasing	$\approx$ Constant	Decreasing
	$ResidualBias_X$	Increasing	Very Low	Increasing	Very Low
	$ResidualBias_Y$	$\approx$ Constant	Very Low	$\approx$ Constant	Very Low

Table 8.5: Data Inspection Summary Table

The data inspection summary Table 8.5 shows that adding random errors to the regressors and responses makes selecting a specific number of latent variables for the best fit much less clear. While a minimum in the fit errors is more or less apparent in the PLS1 simulation, no clear fit error minimum appears in the PLS2 simulation. In Figure 8.12 there is a suggestion that the variation in the fit errors may change around A^* latent variables, but this is the only indication of an optimal latent variable model in the PLS2 simulation data inspection of these residuals plots.

8.6 Latent Variable Selection Methods Analysis

From Tables 8.6 and 8.7 for PLS1 and PLS2 simulations, it is apparent that all the latent variable selection methods have their selection maximum at nLV=A^* for both PLS1 and PLS2. It also appears that all these methods have a tendency towards selecting numbers of latent variables higher than A^*. This tendency towards overfitting is stronger in the PLS1 datasets than PLS2 and for the permutation and information criteria than the crossvalidation selection methods.

nLV-A^*	1st Min RMSECV	abs Min RMSECV	Permulations	Info' Criteria BIC	LV Occurence
-10	0	6	0	84	55.3%
-9	1	20	0	116	62.0%
-8	5	14	0	152	75.0%
-7	4	29	0	171	79.0%
-6	8	62	1	393	100.0%
-5	6	34	1	362	100.0%
-4	17	45	3	402	100.0%
-3	41	152	7	501	100.0%
-2	133	202	27	438	100.0%
-1	419	529	421	373	100.0%
LV=A^*	2426	2455	2074	1521	100.0%
1	1341	1287	1110	1040	100.0%
2	1007	947	1033	777	76.7%
3	961	904	923	652	70.0%
4	948	897	851	715	58.0%
5	714	649	731	453	44.7%
6	608	523	608	399	38.0%
7	240	205	671	221	25.0%
8	25	15	479	20	21.0%
9	0	0	35	0	0.0%
10	0	0	0	0	0.0%
nLV < A^*	7.1%	12.2%	5.1%	34.0%	
nLV = A^*	27.2%	27.4%	23.1%	17.3%	
nLV > A^*	65.6%	60.5%	71.8%	48.7%	

Table 8.6: PLS1 Simulation Latent Variable Selection Summary Table

nLV-A^*	1st Min RMSECV	abs Min RMSECV	Permulations	Info' Criteria BIC	LV Occurence
-10	1	16	8	20	29.7%
-9	1	18	21	11	43.0%
-8	3	34	57	17	58.7%
-7	5	39	44	10	70.5%
-6	12	137	83	51	100.0%
-5	17	91	104	39	100.0%
-4	31	115	100	51	100.0%
-3	84	194	104	238	100.0%
-2	242	362	133	554	100.0%
-1	977	1012	240	1357	100.0%
LV=A^*	4614	4347	2543	955	100.0%
1	1112	1364	709	1377	100.0%
2	820	806	844	983	84.3%
3	788	734	1014	1017	84.2%
4	671	627	1107	1276	80.5%
5	591	500	978	986	70.3%
6	309	242	880	451	57.0%
7	146	97	745	453	41.3%
8	67	39	666	926	29.5%
9	0	0	56	0	0.0%
10	0	0	0	0	0.0%
nLV $< A^*$	13.1%	18.7%	8.6%	21.8%	
nLV $= A^*$	44.0%	40.3%	24.4%	8.9%	
nLV $> A^*$	42.9%	40.9%	67.1%	69.3%	

Table 8.7: PLS2 Simulation Latent Variable Selection Summary Table

In the following analysis, the tendency for overfitting has been analysed by using the difference between the sample base "exact solution" number of latent variables A^* and the best number of latent variables selected in any other way. This gives a comparable basis for comparing all the simulation samples for overfitting. The overfitting tendency as this difference in the numbers of latent variables is then treated as an ordinal response in a logistic regression analysis that includes all the factors in the simulation including $A*$. So (functions of) A^* appear are in both the model regressors and response and A^* generally reported as a strong factor in the regression. An alternative approach would be to use A^* only as a regressor and use the best number of latent variables directly as the response. In this approach A^* appears as a very strong factor that tends to reduce the resolution of the other factors. Consequently, the difference between the latent variable numbers has been used as the response. To be clear, the reason why A^* is often reported as a strong factor in these regressions is because it effects overfitting and is not an artefact of the way it has been used to bring the response to a comparable basis.

This difference in the latent variable numbers is not a continuous response, it is at integer levels and is ordered. Consequently, the regression analysis was ordinal logistic

using the "'ordinal"' R package [12]. The purpose of this analysis is to compare the relative strengths of all the simulation factors on the tendency to overfitting. Consequently prior to the analysis, all the continuous simulation factors were normalised to bring their effects reported by the regression onto a more comparable basis with the categorical factors. So the regression coefficients represent the relative strength of each factor. Rescaling the continuous factors was not done prior to the PLS model building in the simulation, because this would change the covariance and cross-covariance within and between regressor and response matrices. The normalisation here has been done after the PLS model building so can have no influence on the simulation models. The ordinal logistic regressions on each of the latent variable selection criteria were run in two stages. First a factor screening model with all factors as linear terms, followed by a best fit quadratic model generated by forward stepwise selection from all possible factor combinations up to second order. BIC was used as the stopping rule for the stepwise regression. Details of all these ordinal logistic regression models and their confidence interval calculations for the latent variable methods overfitting are shown in Appendix B.

There is no direct analog of the R^2 coefficient of determination from multiple regression in logistic regression.A simple alternative that is applicable to ordinal data is McFadden's pseudo R^2[83] which is defined from a comparison of the log likelihood of the full model against a models with the intercept term only. The levels of overall model fit for these ordinal logistic model fits are compared in Table 8.8.

$$R^2_{McFadden} = 1 - \frac{log(L_{full})}{log(L_{intercept})} \tag{8.55}$$

	Model	RMSECV 1st Min.	RMSECV Abs Min	Permutations	Info Criteria BIC
PLS1	Linear	0.4067	0.2723	0.4273	0.1244
PLS1	Quadratic	0.4334	0.2901	0.4792	0.1618
PLS2	Linear	0.1916	0.1417	0.0925	0.2745
PLS2	Quadratic	0.2678	0.2068	0.1100	0.3729

Table 8.8: Latent Variable Selection Methods Logistic Model Fit Summary

Low R^2 values are a characteristic of logistic regressions, so the values shown in this table do not indicate particularly poor models. The observation here concerns the small differences between the fits for linear screening models and stepwise quadratic models, which indicates that the added higher order terms do not have strong effects. The adverse behaviour of stepwise regression, in particular the tendency to overfit are well known. See Harrell[45] for example. Even though stepwise methods tend to overfit, no strong second order terms appear in the models. Consequently, the use of stepwise regression here appears to be reasonable.

Due to the large number of trials in the simulation, every term with even a small effect in the regressions is likely to have a very low p-value and so appear as highly significant in the regression tables. In this analysis, the strength of factor effects in logistic models have been assessed using their odd's ratio or directly from the coefficients which are the log of the odd's ratios. The effect sizes of the simulation factors for the overfitting tendency for the main four latent variable selection methods from the linear screening models are ranked in Table 8.9. The ranking here is from the absolute value of the regression coefficient. Comparison plots for these coefficients are shown as Figure 8.13. In these plots the coefficients from the linear screening models are plotted directly. The effects for the quadratic stepwise models are represented by the values of the linear factors only. As the continuous factors in these regressions have zero means, any interaction terms will have no contribution to the response at the mean point of the dataset. So the subset of linear terms from the quadratic model approximates the gradient of the response surface at the mean point of the dataset. Points in these plots represent the coefficient estimates with error bars for their 95% confidence intervals. The close match between linear screening and quadratic stepwise points in these plots is further evidence that the effects of any second order effects must be quite small.

The conclusion from the overfitting effects Table 8.9 is that the crossvalidation, permutation and information criteria latent variable selection methods have different sensitivities to the structure of the simulation samples. The appearance of High $A*$ and Slow or Medium VDR_X is interpreted as measures of complexity of the correlations and collinearity within the simulation samples. So finding an overfitting tendency to

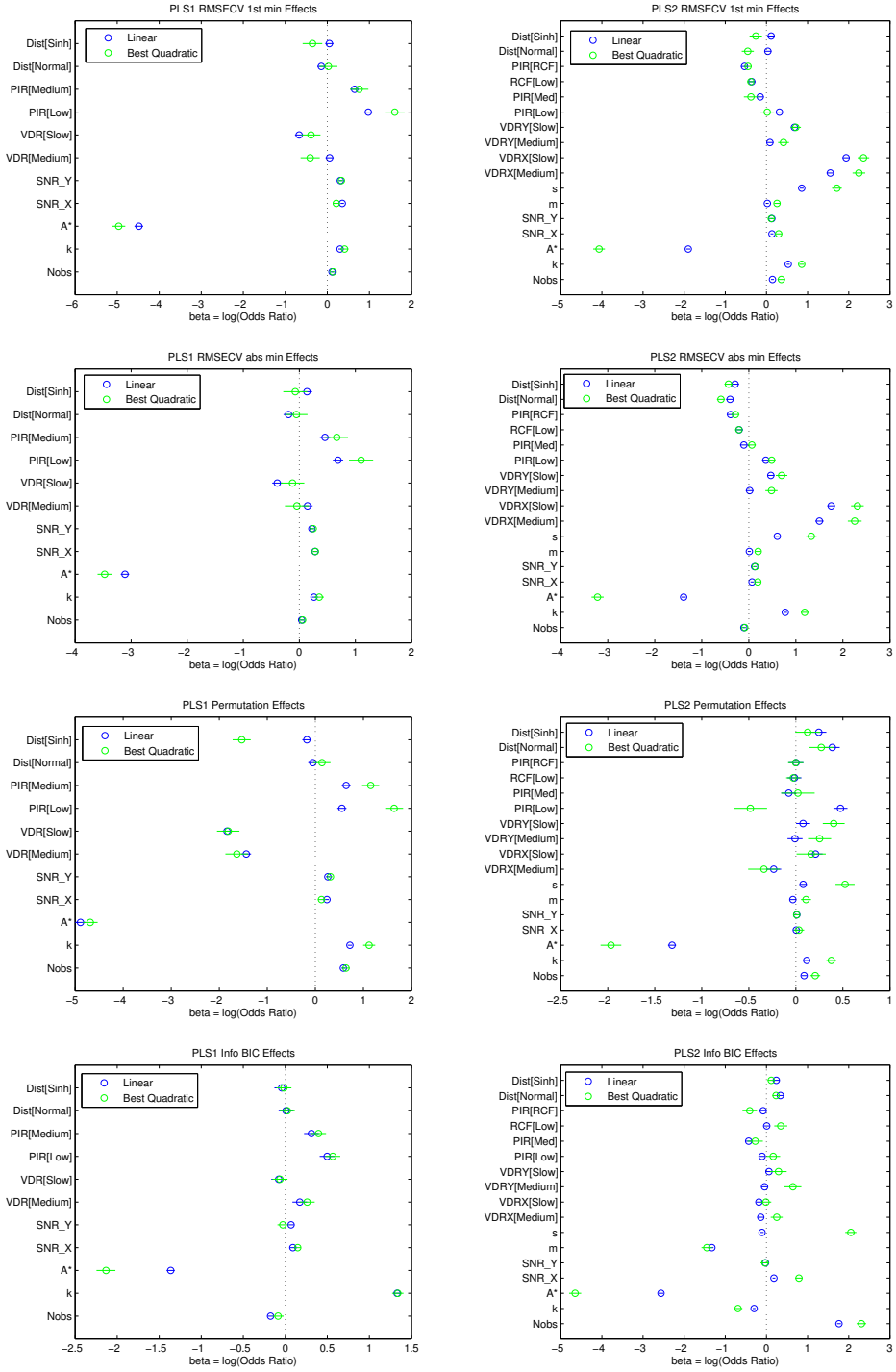

Figure 8.13: Latent Variable Selection Methods Overfitting Effects Plots

	Rank	RMSECV 1st Min.	RMSECV Abs Min	Permutations	Info' Criteria BIC
PLS1	1	High A^*	High A^*	High A^*	High A^*
	2	Low PIR[Medium]	Low PIR[Medium]	High $VDR_X[Low]$	Low k
	3	High $VDR_X[Low]$	Low PIR[Low]	High $VDR_X[Medium]$	Low PIR[Medium]
	4	Low PIR[Low]	High $VDR_X[Low]$	Low k	Low PIR[Low]
	5	Low SNR_X	Low SNR_X	Low PIR[Low]	High Nobs
	6	Low SNR_Y	Low k	Low Nobs	Low $VDR_X[Medium]$
	7	Low k	Low SNR_Y	Low PIR[Medium]	Low SNR_X
	8	High Dist'[Normal]	High Dist'[Normal]	Low SNR_Y	High $VDR_X[Low]$
	9	Low Nobs	Low $VDR_X[Medium]$	Low SNR_X	Low SNR_Y
	10	Low $VDR_X[Medium]$	Low Dist'[logNormal]	High Dist'[logNormal]	High Dist'[logNormal]
	11	Low Dist'[logNormal]	Low Nobs	High Dist'[Normal]	Low Dist'[Normal]
PLS2	1	Low $VDR_X[Slow]$	Low $VDR_X[Slow]$	High A^*	High A^*
	2	High A^*	Low $VDR_X[Medium]$	Low PIR[Slow]	Low Nobs
	3	Low $VDR_X[Medium]$	High A^*	Low Dist'[Normal]	High m
	4	Low s	Low k	Low Dist'[Sinh]	High PIR[medium]
	5	Low $VDR_Y[Slow]$	Low s	High $VDR_X[Medium]$	Low Dist'[Normal]
	6	High RCF[Medium]	Low $VDR_Y[Slow]$	Low $VDR_X[Slow]$	High k
	7	Low k	High Dist'[Normal]	Low k	Low Dist'[Sinh]
	8	High RCF[Slow]	High RCF[Medium]	Low Nobs	High $VDR_X[Slow]$
	9	Low PIR[Slow]	Low PIR[Slow]	Low s	Low SNR_X
	10	High PIR[medium]	High Dist'[Sinh]	Low $VDR_Y[Slow]$	High $VDR_X[Medium]$
	11	Low Nobs	High RCF[Slow]	High PIR[medium]	High s
	12	Low SNR_X	Low SNR_Y	High m	High PIR[Slow]
	13	Low SNR_Y	High Nobs	High RCF[Slow]	High RCF[Medium]
	14	Low Dist'[Sinh]	High PIR[medium]	Low SNR_Y	Low $VDR_Y[Slow]$
	15	Low $VDR_Y[Medium]$	Low SNR_X	High $VDR_Y[Medium]$	High$VDR_Y[Medium]$
	16	Low Dist'[Normal]	Low $VDR_Y[Medium]$	Low SNR_X	High SNR_Y
	17	Low m	Low m	Low RCF[Medium]	Low RCF[Slow]

Table 8.9: Latent Variable Selection Methods Overfitting Effects Summary

be associated with multivariate complexity might be expected. Less apparent is that regressor and response signal to noise ratios SNR_X and SNR_Y do not appear as strong factors for overfitting. So the conclusion must be that the overfitting tendency of PLS latent variable selection methods must be more closely associated with the detail of the correlation and collinear structure rather than overall error level as typical for ordinary multiple regression methods. The role of the number of regressor variables k in this table is not very consistent. So the simulation datasets were split into portrait and landscape subsets and the analysis repeated. No systematic differences were apparent in the PLS1 dataset but in the PLS2 dataset, lower values of k have increase overfitting in the portrait subset while higher values of k have increase overfitting in the landscape subset.

8.7 Model Coefficients Analysis

The sensitivity of the PLS coefficients to latent variable selection has been assessed in two ways. For each simulation sample the coefficients from the base models at the "true" values at A^* latent variables and coefficients from the simulation samples at a range of latent variables was calculated. The correlation between these two sets of coefficients was used to assess simulation samples. As an alternative to assessment by correlation, the second method is an assessment of wether the base coefficients are the same or different to the coefficients from the simulation samples. This second method tests if the coefficients from the base models at A^* latent variables is within the sample coefficient confidence intervals at all latent variable numbers. The coefficient confidence intervals here have been calculated by local linearization.

Figures 8.14 and 8.15 show box plots of the correlation between the sample coefficients and the underlying base PLS1 and PLS2 model. Each box plot is for a subset of samples with a specific number of base latent variables A^* and shows how the distribution of the sample correlations for this subset changes with the number of latent variables in the sample models. The box plots from the PLS1 simulations include all 9000 samples but the box plots from the PLS2 simulations are from a reduced subset of 2400 from 12000 samples. Both sets of plots show that the correlations are maximized at a latent variable number around A^* with the location of the maximum increasing with A^*. But the location of the maximum tends to lag behind A^*, which is more evident in the median correlation coefficient values shown as Tables 8.10 and 8.12. The difference between the maximum median correlation values and the values at A^* is generally quite small, only becoming of practical importance at high numbers of latent variables.

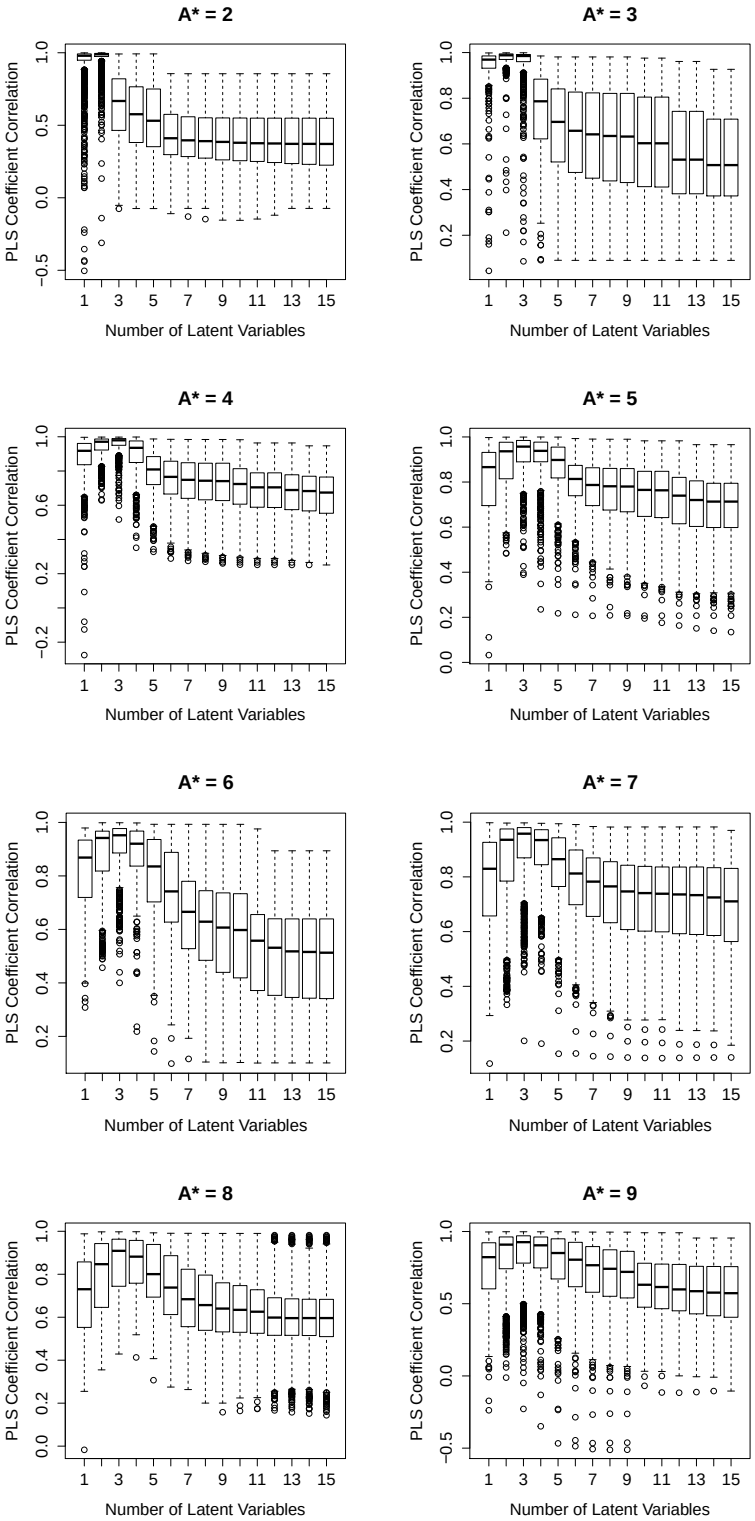

Figure 8.14: PLS1 Coefficient Correlation vs. Latent Variables

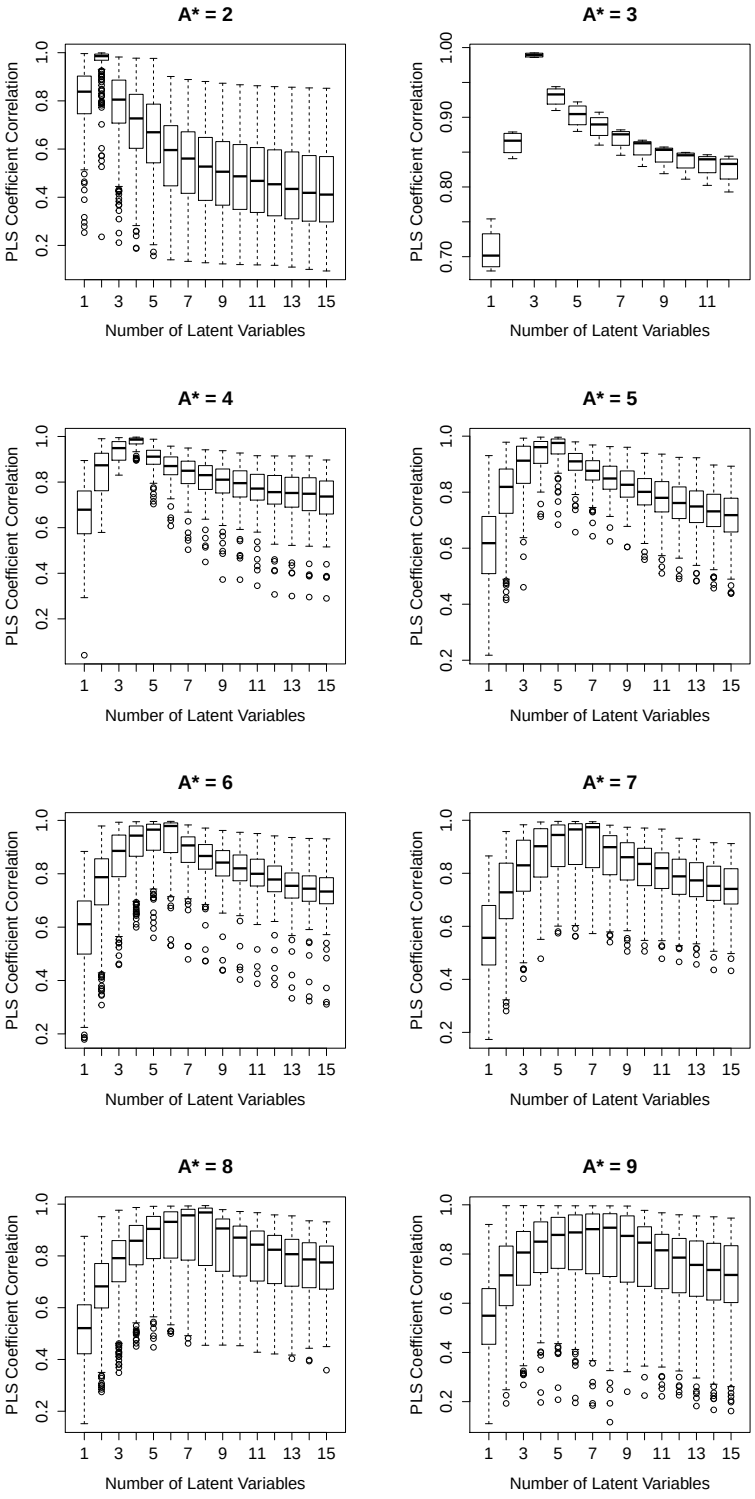

Figure 8.15: PLS2 Coefficient Correlations

LV	$A^* = 2$	$A^* = 3$	$A^* = 4$	$A^* = 5$	$A^* = 6$	$A^* = 7$	$A^* = 8$	$A^* = 9$
1	0.9794	0.9955	0.9937	0.9926	0.9778	0.9885	0.9848	0.9915
2	0.9896	0.9987	0.9981	0.9988	0.9987	0.9964	0.9973	0.9970
3	0.6674	0.9893	0.9907	0.9992	0.9987	0.9971	0.9975	0.9971
4	0.5761	0.8220	0.9812	0.9974	0.9977	0.9947	0.9971	0.9958
5	0.5311	0.7229	0.8759	0.9832	0.9898	0.9897	0.9905	0.9928
6	0.4104	0.5780	0.7876	0.8875	0.9505	0.9645	0.9793	0.9894
7	0.3962	0.5415	0.7663	0.8671	0.8802	0.9143	0.9604	0.9839
8	0.3906	0.5293	0.7564	0.8615	0.8732	0.9008	0.9491	0.9807
9	0.3852	0.5215	0.7522	0.8602	0.8698	0.8891	0.9456	0.9792
10	0.3794	0.4998	0.7241	0.8402	0.849	0.8752	0.9281	0.8733
11	0.3761	0.4942	0.7012	0.8312	0.6977	0.8185	0.8208	0.8142
12	0.3744	0.4502	0.6826	0.7974	0.6544	0.8035	0.7105	0.7417
13	0.3717	0.4489	0.6740	0.7809	0.6485	0.8020	0.7072	0.7218
14	0.3716	0.4393	0.6624	0.7685	0.6458	0.7967	0.7014	0.7129
15	0.3712	0.4375	0.6516	0.7652	0.6451	0.7867	0.6981	0.7104
Best	2	2	2	3	3	3	3	3

Table 8.10: PLS1 Median Coefficient Correlation Table

LV	$A^* = 2$	$A^* = 3$	$A^* = 4$	$A^* = 5$	$A^* = 6$	$A^* = 7$	$A^* = 8$	$A^* = 9$
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	0.96	1	1	1	1	1	1	1
5	0.96	1	1	1	1	1	1	1
6	0.52	1	1	1	1	1	1	1
7	0.48	1	1	1	1	1	1	1
8	0.40	0.8636	1	1	1	1	1	1
9	0.36	0.5164	1	1	1	1	1	1
10	0.32	0.0598	0	0	0	0.0022	0	0
11	0.28	0.0272	0	0	0	0	0	0
12	0.24	0.0075	0	0	0	0	0	0
13	0.24	0	0	0	0	0	0	0
14	0.20	0	0	0	0	0	0	0
15	0.16	0	0	0	0	0	0	0
Best	3	7	8	8	8	8	8	8
Best	2	3	4	4	5	5	7	8

Table 8.11: PLS1 Coefficient Median Inclusion CoverageTable

LV	$A^* = 2$	$A^* = 3$	$A^* = 4$	$A^* = 5$	$A^* = 6$	$A^* = 7$	$A^* = 8$	$A^* = 9$
1	0.8382	0.7542	0.7650	0.7193	0.6811	0.6310	0.5707	0.5761
2	0.9857	0.8789	0.9573	0.9510	0.9417	0.9046	0.8503	0.8834
3	0.8052	0.9924	0.9949	0.9923	0.9916	0.9776	0.9672	0.9795
4	0.7271	0.9416	0.9963	0.9965	0.9949	0.9931	0.9843	0.9899
5	0.6697	0.9144	0.9652	0.9947	0.9956	0.9949	0.9911	0.9931
6	0.5959	0.8916	0.9325	0.9627	0.9943	0.9941	0.9917	0.9937
7	0.5606	0.8764	0.9119	0.9417	0.9665	0.9923	0.9924	0.9941
8	0.5272	0.8627	0.8955	0.9222	0.9479	0.9656	0.9912	0.9945
9	0.5059	0.8533	0.8868	0.9092	0.9302	0.9450	0.9608	0.9918
10	0.4868	0.8452	0.8748	0.8824	0.9104	0.9267	0.9398	0.9488
11	0.4679	0.8394	0.8471	0.8661	0.8898	0.9095	0.9251	0.9261
12	0.4539	0.8316	0.8447	0.8428	0.8564	0.8751	0.9073	0.9079
13	0.4345		0.7447	0.8042	0.8172	0.8536	0.8860	0.8909
14	0.4182		0.7336	0.7900	0.8025	0.8366	0.8705	0.8787
15	0.4110		0.7247	0.7754	0.7936	0.8247	0.8554	0.8683
Best	2	3	4	4	5	5	7	8

Table 8.12: PLS2 Median Coefficient Correlation Table

Coefficient Estimate Intervals by Local Linearization

Confidence intervals for PLS by local linearization were originally from Denham [17], then improved by Serneels, Lemberge and Van Espen [97]. The basic prediction equation assumes normally distributed errors, with the variance of the coefficient estimates given by

$$var[\beta] = \mathbf{J}\mathbf{J}^T\hat{\sigma}^2 \tag{8.56}$$

so that the prediction interval at a point $\mathbf{x}_0$ and at the α significance level is

$$\hat{\beta} \pm t_{\alpha/2,dof}\sqrt{diag(var[\beta])} \tag{8.57}$$

where dof is the degrees of freedom, $\mathbf{J}_0$ is the coefficient Jacobian $\partial\beta/\partial y$ and $\hat{\sigma}^2 = RSS/dof$. For this simulation, the degrees of freedom and the coefficient Jacobian were calculated using the numerical methods described in section 6.4. While the regressor and so response values in the base PLS model are not all normally distributed, the additional random errors added to response and regressor matrix are normally distributed, so this assumption of normally distributed prediction errors is not unreasonable here.

8.7.1 Effect of the Number of Latent Variables on Simulated Coefficients

Figures 8.16 and 8.17 shows box plots of the coverage probabilities for the coefficients from the base model at A^* latent variables appearing between the coefficient confidence intervals from subsets of the simulation samples at specific numbers of base latent variables A^*. These sets of plots show similar patterns with the location of the maximum coverage starting near A^* but lagging behind A^* as the number of latent variables in the model increases. The tabulated median coverage values shown as Table 8.11 for the PLS1 simulation are not so informative as the median values do not resolve the differences, Median coverage values for the PLS2 simulation are not shown because they are either 1 or 0.

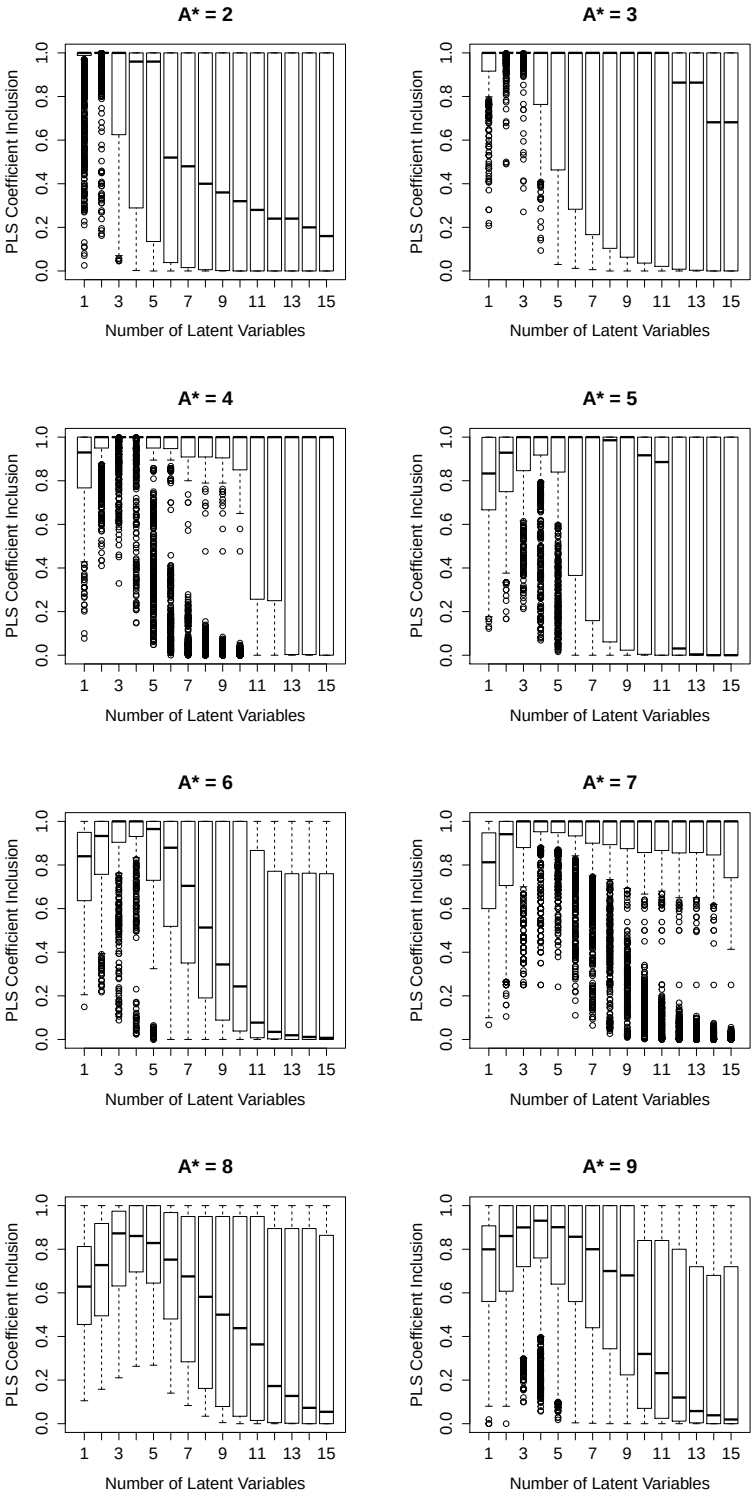

Figure 8.16: PLS1 Coefficient Median Inclusion

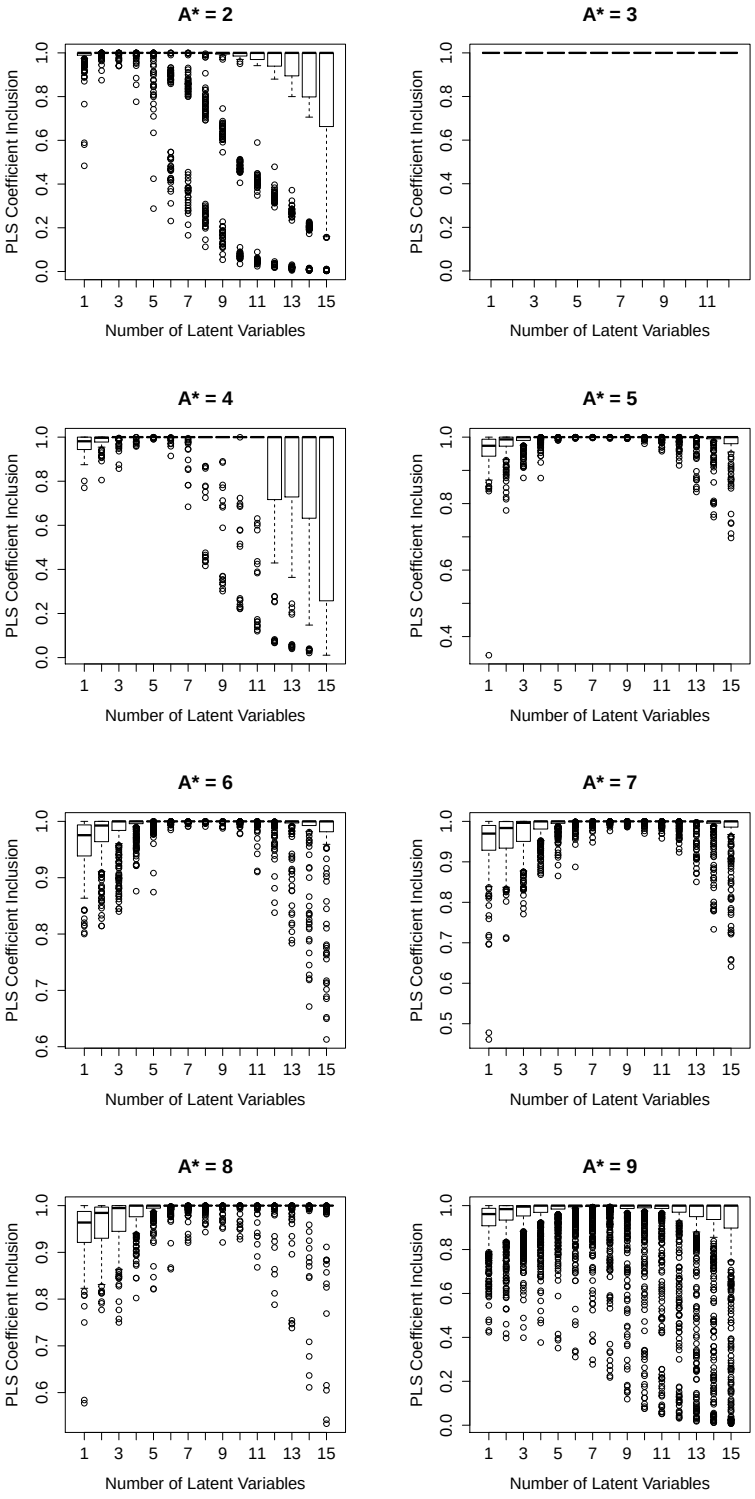

Figure 8.17: PLS2 Coefficient Median Inclusion

8.7.2 Effects of Simulation Factors on Coefficient Identification

The effects of the simulation factors on the tendency to select numbers of latent variables that under fit or overfit the best values for coefficient correlation or coverage has also been analysed by ordinal logistic regression. The response here is the difference between the best number of latent variables that maximize correlation or confidence interval inclusion between the base and fitted simulation models and the number of latent variables in the base model, A^*. The methods used for the ordinal logistic regression here were the same as that for the latent variable selection method described previously is Section 8.6. Details of these ordinal logistic regression models and their confidence interval calculations for the latent variable methods overfitting are shown in Appendix C.

Table 8.13 shows a summary of the analysis of the simulation factors on the overfitting tendency to overfitting of the number of latent variables identified by coefficient correlation and coverage. The factors are ranked by the absolute value of their correlation coefficient in this table. In Figure 8.18, the coefficients for the linear terms in both linear screening and quadratic stepwise models are plotted.

Rank	PLS1 Coefficient Correlation	PLS1 Coefficient Coverage	PLS2 Coefficient Correlation
1	High A^*	High A^*	High A^*
2	High $VDR_X[Slow]$	High $VDR_X[Slow]$	Low $VDR_X[medium]$
3	Low $PIR[Low]$	High $VDR[Medium]$	Low $VDR_X[slow]$
4	High $VDR[Medium]$	Low $PIR[Low]$	Low s
5	Low $PIR[Medium]$	Low $PIR[Medium]$	Low $PIR[Low]$
6	Low SNR_X	Low k	Low N
7	Low N	High $Distribution[Normal]$	Low $Distribution[Sinh]$
8	Low SNR_Y	Low $Distribution[Skew]$	High $VDR_Y[slow]$
9	High $Distribution[Skew]$	Low SNR_Y	High $RCF[Medium]$
10	High k	Low SNR_X	Low $PIR[Medium]$
11	High $Distribution[Normal]$	High N	Low SNR_X
12			Low k
13			Low m
14			High $VDR_Y[medium]$
15			Low $Distribution[Normal]$
16			High $RCF[Low]$
17			Low SNR_Y

Table 8.13: Coefficient Identification Underfitting or Overfitting Factors Table

In Table 8.14, the differences in fit between linear screening and quadratic stepwise models are larger than those observed for the latent variable selection analysis, but this

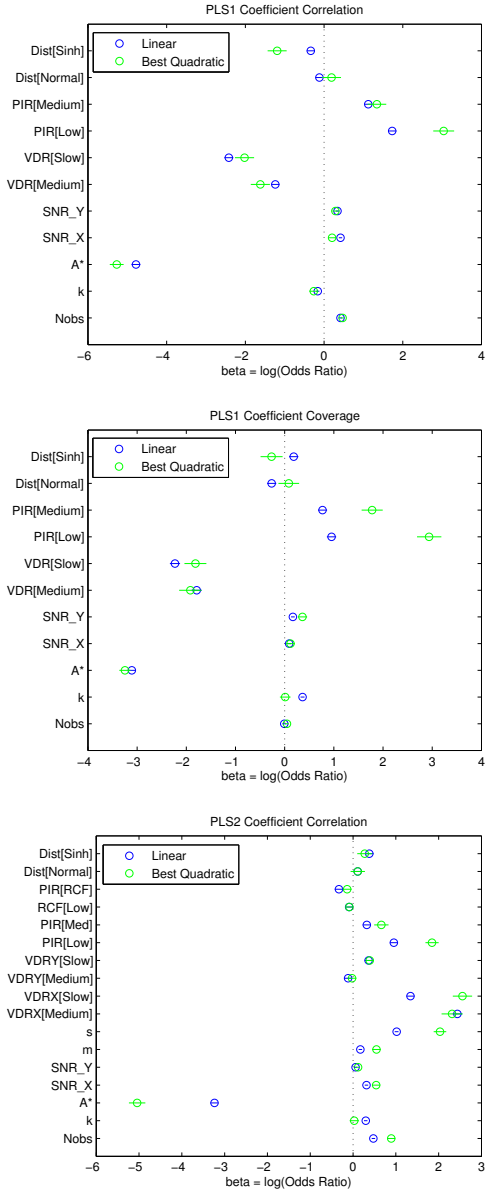

Figure 8.18: Coefficient Effects Plots

	Response	Model	McFadden's R^2
PLS1	Coefficient Correlation	Linear Screening	0.4534
		Quadratic Stepwise	0.5075
PLS1	Coefficient Coverage	Linear Screening	0.3154
		Quadratic Stepwise	0.3856
PLS2	Coefficient Correlation	Linear Screening	0.3079
		Quadratic Stepwise	0.3996

Table 8.14: Coefficient Identification Logistic Model Fit Summary

difference is not so apparent in the coefficient plots. The largest difference between linear and quadratic models is in the PLS2 Coefficient Correlation analysis and is due to a strong interaction between A^* and s, the response matrix rank.

8.7.3 Conclusions on Coefficient Analysis by Simulation

Overall, coefficient correlation appears to be a more sensitive indicator of the differences between the simulation models than coefficient coverage from the examination of the fit box plots Figures 8.14 and 8.15. The base number of latent variables A^* does not appear to be a clear optimum here. The optima number of latent variables for coefficient correlation can be a lot less than A^*. But the difference between the optimal and A^* values of the correlation coefficients can be small and may be of no practical significance. The detrimental effect on both coefficient correlation and inclusion coverage from overfilling with numbers of latent variables greater than A^* is clear in all these plots and data tables.

The factor in the simulation that is having the strongest effect on the under or overfitting tendency is again the number of latent variables in the base model A^*, as observed in the analysis of the latent variable selection methods. After this, the same structural factors in the regressor matrix are again found to have the strongest effects. These as the regressors variance decay rate VDR_X and the pattern in the internal regression, PIR. Response distributions and overall regressors and response error levels have weaker effects. The effects for PLS1 and PLS2 appear very similar, apart from the rank of the response matrix s having a relatively strong effect. There is also a strong interaction here between s and A^*, so the mechanism must be complex.

8.8 Model Prediction Analysis

The strategy here is to use the same base model as the simulation PLS base model then generate a different alternative dataset that has the same perfect fit. This method is more complex than that used for assessing coefficients, but it was considered that the prediction set should be as independent as possible from the base model for a thorough

assessment of prediction performance. A sample of 100 predictions were generated for each simulation sample to give a consistent estimate of coverage probabilities. The details of the method are

- Starting from the set of simulation sample factors n,k,A^*... calculate the base regressor and response matrices and base PLS model using the methods described previously.
- Use the same set of simulation sample factors to generate an second base regressor with 100 observations and the same correlation and covariance structures as the base regressor matrix but different randomisation. These are the "true" regressor values for assessing the prediction performance.
- Use this true regressor matrix and the base PLS model to predict the "true" response values.
- Add the relevant level of regressor and response error levels to make the prediction simulation samples.
- Calculate the fitted response values and their corresponding confidence intervals by local linearization.

As a comparison to the previous section on coefficients, this analysis of predictions in the simulation uses the two methods. Fit between the fitted values from the base models at A^* latent variables with predictions from the simulation samples over a range of numbers of latent variables has been assessed by RMSE. Coverage was assessed by comparing these fitted values to the prediction interval from the simulation samples, where the prediction interval was calculated by local linearization. Only prediction analysis from the PLS1 simulation is reported here.

Prediction Intervals by Local Linearization

Prediction intervals for PLS by local linearization were originally from Denham [17], then improved by Serneels, Lemberge and Van Espen [97] and by Romera [94]. The basic prediction equation assumes normally distributed errors, so that the prediction

LV	$A^* = 2$	$A^* = 3$	$A^* = 4$	$A^* = 5$	$A^* = 6$	$A^* = 7$	$A^* = 8$	$A^* = 9$
1	0.0414	0.0102	0.0034	0.0012	0.0030	0.0030	0.0028	0.0027
2	0.0308	0.0031	0.0020	0.0013	0.0016	0.0017	0.0017	0.0012
3	0.0572	0.0046	0.0025	0.0011	0.0017	0.0016	0.0012	0.0013
4	0.0724	0.0140	0.0040	0.0014	0.0019	0.0026	0.0024	0.0018
5	0.0801	0.0204	0.0078	0.0023	0.0023	0.0040	0.0036	0.0026
6	0.0727	0.0230	0.0086	0.0027	0.0039	0.0080	0.0058	0.0035
7	0.0773	0.0250	0.0091	0.0028	0.0049	0.0105	0.0059	0.0042
8	0.0781	0.0259	0.0100	0.0028	0.0053	0.0116	0.0064	0.0054
9	0.0791	0.0262	0.0101	0.0028	0.0054	0.0116	0.0064	0.0059
10	0.0799	0.0259	0.0087	0.0026	0.0054	0.0118	0.0065	0.0055
11	0.0808	0.0263	0.0078	0.0024	0.0053	0.0119	0.0066	0.0055
12	0.0815	0.0248	0.0077	0.0022	0.0053	0.0118	0.0062	0.0055
13	0.0822	0.0248	0.0069	0.0020	0.0053	0.0114	0.0062	0.0054
14	0.0826	0.0247	0.0069	0.0019	0.0053	0.0111	0.0061	0.0054
15	0.0828	0.0247	0.0067	0.0019	0.0053	0.0094	0.0059	0.0054
Best	2	2	2	3	2	3	3	2

Table 8.15: PLS1 Prediction Median RMSE Table

interval at a point $\mathbf{x}_0$ and at the α significance level is

$$\hat{\mathbf{y}} \pm t_{\alpha/2,dof}\hat{\sigma}\big[\frac{n+1}{n} + \mathbf{x}_0^T\mathbf{J}_0\mathbf{J}_0^T\mathbf{x}_0\big]^{1/2} \tag{8.58}$$

where dof is the degrees of freedom, $\mathbf{J}_0$ is the coefficient Jacobian $\partial\beta/\partial y$ at $\mathbf{y}_0$ and $\hat{\sigma}^2 = RSS/dof$. For this simulation, the degrees of freedom and the coefficient Jacobian were calculated using the numerical methods described in section 6.4.

8.8.1 Effect of the Number of Latent Variables on Simulated Prediction

Sets of box plots for prediction fit RMSE and prediction interval coverage are shown as Figures 8.19 and 8.20. The number of latent variables for the minimum values of RMSE start at A^* for two latent variables and increase with increasing A^* but appear to lag behind just as the coefficient analysis in the previous section. Median values of the prediction RMSE values are shown in Table 8.15. The number of latent variables for the minimum RMSE in this PLS1 simulation is very similar to the number of latent variables for maximum coefficient correlation shown in Table 8.10. But the difference is that the median correlation values at A^* latent variables and their maximum values is quite small, but the corresponding differences in prediction RMSE in Table 8.15 can be more than four times larger.

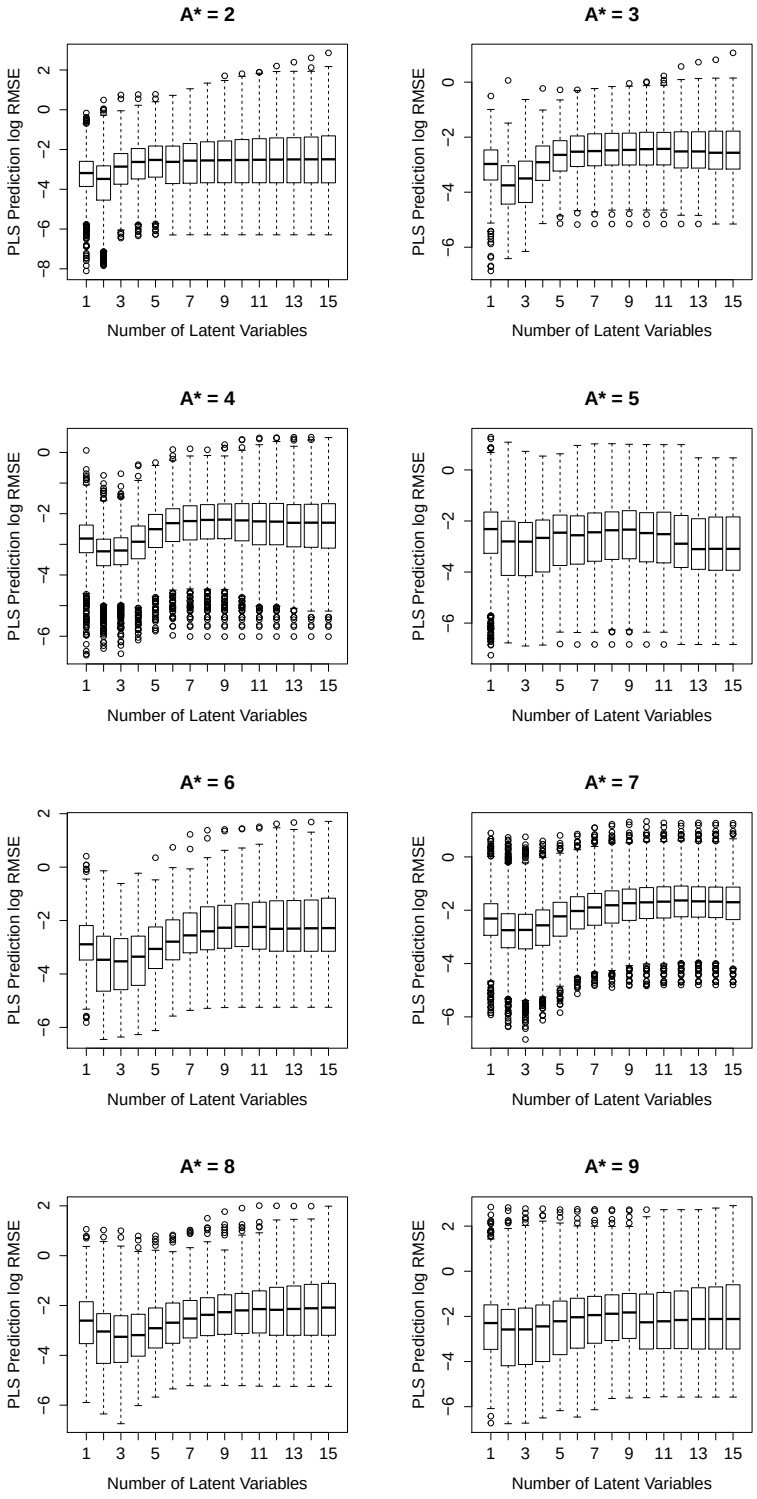

Figure 8.19: PLS1 Prediction RMSE

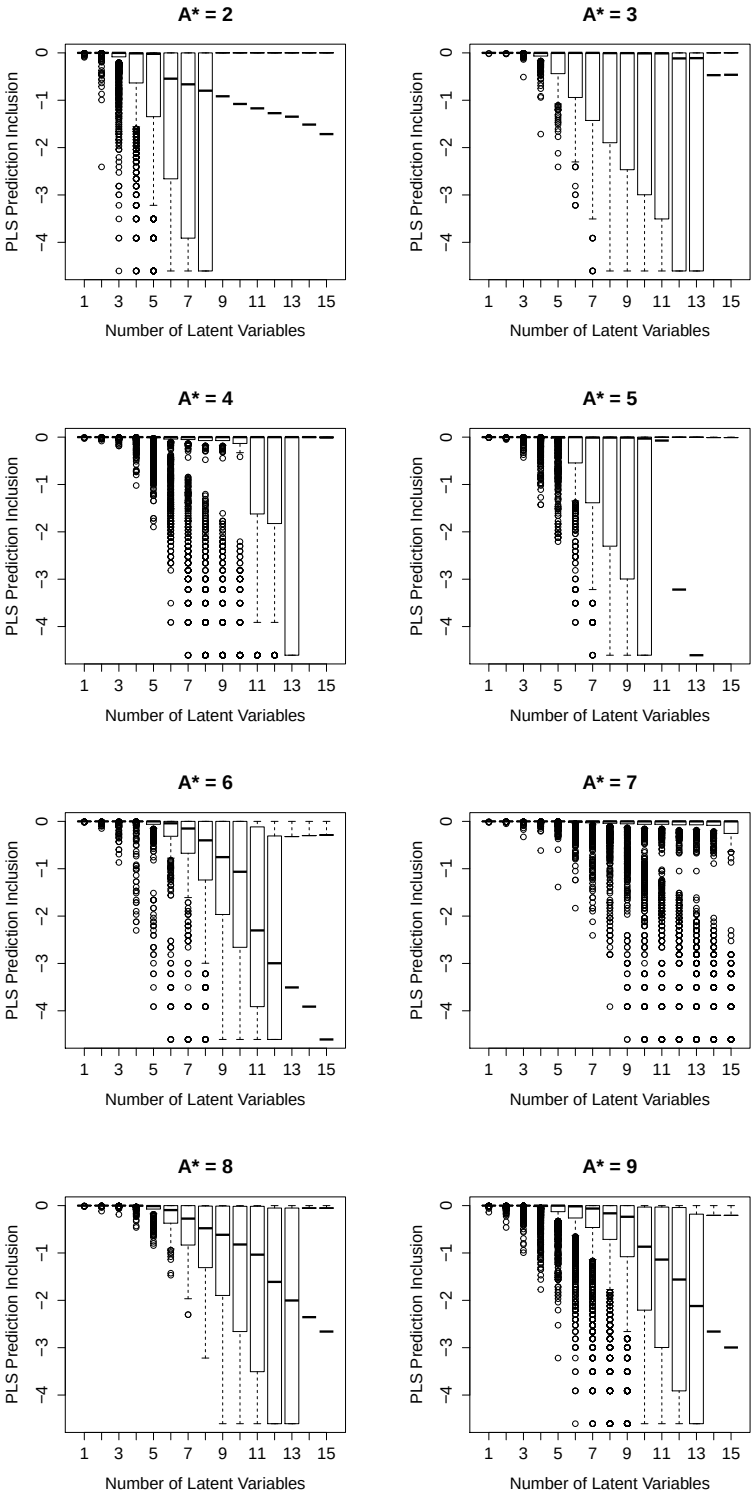

Figure 8.20: PLS1 Prediction Inclusion

LV	$A^* = 2$	$A^* = 3$	$A^* = 4$	$A^* = 5$	$A^* = 6$	$A^* = 7$	$A^* = 8$	$A^* = 9$
	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	0.9900	1	1	1	1	1	1	1
5	0.9800	1	1	1	1	1	1	1
6	0.5800	1	1	1	1	1	1	1
7	0.5150	1	1	1	1	1	1	1
8	0.4500	0.9800	1	1	1	1	1	1
9	0.4000	0.8900	1	1	1	1	1	1
10	0.3400	0.1014						
11	0.3100	0.0500						
12	0.2800	0.0100						
13	0.2600	0.0100						
14	0.2200							
15	0.1800							

Table 8.16: PLS1 Prediction Interval Median Coverage Table

8.8.2 Effect of Simulation Factors on Simulated Prediction

Analysis of the effects of the simulation factors on the effects on the number of latent variables on performance of the predictions has also been made by ordinal logistic regression, using the same methods as described previously. These models are shown in detail in Appendix D. Table 8.17 shows a summary of the analysis of the simulation factors with the strongest effects on the under or overfitting tendency on the number of latent variables identified by prediction RMSE and coverage over the prediction intervals. Figure 8.21 shows a comparison of the effects between linear screening and quadratic stepwise models.

The PLS1 prediction coverage analysis is unusual in that the stepwise regression did not add any terms beyond those in the linear screening model. In this model, the number of latent variables in the simulation base model A^* has an overwhelming effect.

The prediction performance model fits shown in Table 8.18 do not show any large changes from the inclusion of second order effects. This is also apparent in the coefficient plots, Figure 8.21. The factor effect ranking for PLS1 and PLS2 prediction RMSE are in a very similar order, with the structural factors of the regressors matrix VDR_X and PIR having high ranking. The main difference is the strong effect of s, the response matrix rank in the PLS2 prediction RMSE latent variable difference model. This factor also appears in the coefficient identification model where there is a strong interaction between s and A^*. This interaction is also strong in this prediction model,

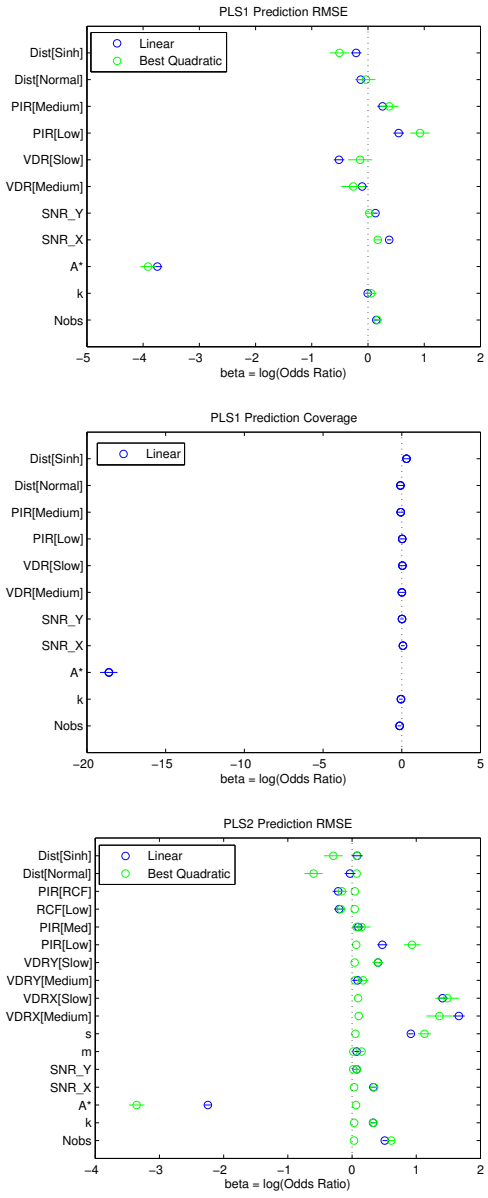

Figure 8.21: Prediction Effects Plots

Rank	PLS1 Prediction RMSE	PLS1 Prediction Coverage	PLS2 Prediction RMSE
1	High A^*	HighA^*	HighA^*
2	Low $PIR[Low]$	Low $Distribution[Sinh]$	Low $VDR_X[Medium]$
3	High $VDR_X[Slow]$	High N	Low $VDR_X[Slow]$
4	Low SNR_X	High $Distribution[Normal]$	Low s
5	Low $PIR[Medium]$	Low SNR_X	Low N
6	High $Distribution[Sinh]$	High $PIR[Medium]$	Low $PIR[Low]$
7	Low N	High k	Low $VDR_Y[Slow]$
8	Low SNR_Y	Low$VDR_X[Slow]$	Low SNR_X
9	High $Distribution[Normal]$	Low $PIR[Low]$	Low k
10	High $VDR_X[Medium]$	Low SNR_Y	Low High$RCF[Medium]$
11	High k	Low $VDR_X[Medium]$	High$RCF[Low]$
12			Low $PIR[Medium]$
13			Low $VDR_Y[Medium]$
14			Low $Distribution[Sinh]$
15			Low m
16			Low SNR_Y
17			High $Distribution[Normal]$

Table 8.17: Prediction Identification Overfitting Factors Table

but is not the strongest interaction as the interactions between s and VDR_X are also of comparable strength.

	Response	Model	McFadden's R^2
PLS1	Prediction RMSE	Linear Screening	0.3427
		Quadratic Stepwise	0.3572
PLS1	Prediction Coverage	Linear Screening	0.9029
		Quadratic Stepwise	As Linear
PLS2	Prediction RMSE	Linear Screening	0.2124
		Quadratic Stepwise	0.2511

Table 8.18: Prediction Performance Logistic Model Fit Summary

8.9 Overall Conclusions from the Simulation

- The choice of latent variable number for the strongest effect interpretation by coefficients and for maximum prediction performance appears to be the same number of latent variables.

- This optimum number of latent variables can be quite low. In the simulation, it is generally lower than the number of latent variables A^* in the base simulation sample where there is an exact solution between regressors and response.

- Overfitting by selecting a number of latent variables higher than the optimal number will have an adverse effect on both coefficient interpretation and prediction performance, even though this optimal number may be less than A^*.

- All the latent variable selection methods in the simulation study tend to overfit by selecting latent variable numbers greater than A^*.

- The latent variable selection method in the simulation study with the least tendency towards overfitting uses the first minimum in the RMSECV plot.

- The different performance of these latent variable selection methods have has some dependence on the structure of the simulation sample.

- The degree of overfitting common to all these latent variable selection methods is mainly associated with factors related to the structure of the regressor matrix. Factors related to the internal regression coefficients, the structure of the response matrix for PLS2 or for the overall error levels appear to have secondary or minor effects on overfitting.

- The distribution of the regressors or responses does not appear to have any strong effect on PLS coefficients or prediction performance in the simulation. It follows that when transforming regressors or responses prior to building PLS models changes the fit or predictions, it is more likely that the transforms have changed the cross-covariance between $\mathbf{X}$ and $\mathbf{Y}$ that have resulted in a different PLS model rather than the nature of the distribution having any direct effect.

- In the overfitting analysis, some response curvature effects were detected and are reported in the effect summary tables. No particularly strong curvature effects were detected in that they only appeared late in the stepwise factor selection after the strongest factors had been included. The inclusion of second order for curvature did not result in large changes in the effects of other factors.

The original objective of this simulation was to compare the performance of the latent variable selection methods and to determine how the structure of simulation sample datasets influenced this performance. These objectives were explicit in the series of questions posed at the start of this chapter, which in the main have been answered in theses conclusions.

Chapter 9

Discussion and Conclusions

9.1 The Consequences of the Simulation

The unexpected conclusion from the simulation was that all the latent variable selection methods in the study had a tendency towards overfitting, to the detriment of model interpretation and prediction performance. The finding here that overfitting in the simulation is more related to the structure of the regressor matrix rather than the response or overall random error levels may not be intuitive. In the simulation, A^* latent variables represent the exact solution to the base PLS model before random errors are added to form the simulation samples. But a number of cases in the simulation shows that the optimal number of latent variables may be less than A^*. For the PLS1 prediction RMSE simulation the optimal numbers of latent variables is far less than A^*. The simulation factors effects analysis consistently shows that the scale of the random errors is not an strong factor for overfitting. The significance of these effects tables is not in what they contain, it is in what they are missing. As this difference in the optimal number of latent variables with and without the random errors is associated with the structure of the regressor matrix, it must be that the effect of the random errors must be by acting specifically on the fit within the space of the model rather than having a more general effect on the overall information content.

This conclusion is consistent with other studies. Control of the variation within and outside of the model space in PLS is known to influence overfitting. In the original orthogonalised variation on PLS known as O-PLS from Trygg and Wold[103]

the variation in the regressor that is orthogonal to the response is removed to reduce model complexity. The review on orthogonal versions of PLS by Fonville et al.[33] claims that the success of this specific version of PLS in genomics applications is due to its reduced tendency to overfitting. A recent paper by Cloarec[15] on overfitting in univariate PLS1 shows how random error terms influence the scores calculations by contributing to the diagonal of the regressor covariance matrix. The conclusion from this paper was that the inflation of these diagonal elements leads directly to overfitting. This simulation starting from the widest credible range of datasets and covered all combinations of structural factors has come to the same conclusion that overfitting in PLS is somehow associated with variation within the model space. This conclusion is consistent for all PLS1 structures and for PLS2 over the restricted range of samples in the simulation.

9.2 The Latent Variable Selection Methods

The conclusion that overfitting may have more to do with the regressor structure and PLS algorithm than the latent variable selection method has reduced the importance of the selection method study as a comparison. The first minimum in RMSECV has the best overall performance in the simulation, but this is probably simply due to the crossvalidation consistently selecting lower latent variable numbers than the other methods and not any enhanced ability at latent variable identification. However, a latent variable selection method is required for any PLS algorithm so the work presented here to to improve selection methods overall and extend them to PLS2 is still of value.

The comparison of the latent variable selection methods applied to the example datasets should be viewed in terms of how consistent are the methods for real datasets rather than the idealised datasets from the simulation. The overall selections are summarised as Table 9.1, but before examining this in detail consider what might be expected here. Each latent variable selection method has used a different criteria,so in terms of “Null Hypothesis Significance Tests” different answers might be anticipated.

But the simulation shows that different aspects of coefficient correlation and prediction RMSE for PLS1 identify the same number of latent variables. In their study of permutation methods, Wiklund et al[109] comment "... indeed it is important to remember that an exact number does not exist, but rather an appropriate interval". In this respect the appropriate interval may depend on the dataset structure rather than the latent variable selection method.

Dataset	Scaling	Max LVs	Equivalent Scaling	Cov' Overfit max LVs	RMSECV Corner	Covariance Permutations	R^2Q^2 Permutations	Information Criteria
WineAroma	Centred	17	OK	max 5	2 or 8?	2	1-17 OK	13
	Scaled	17	NOK	max 16	4	1	1-17 OK	6
Gasoline	Centred	59	OK?	max 7	4	6	1-10+ OK	3
	Scaled	59	NOK?	max 7	5	5	1-10+ OK	3
WasteGlass	Centred	11	NOK	max 9	11	2	1-11 OK	10
	Scaled	11	OK	max 5	11	2	1-11 OK	10
OliveOil	XC YC	5	OK	max 1	2	?	2,3,4 OK	2
	XC YS	5	NOK	max 1	2	?	2,3 OK	2
	XS YC	5	OK	max 5	2	1	2,3 OK	1
	XS YS	5	OK	max 5	2	3	2,3,4 OK	2
Biscuits	XC YC	39	OK	max 10	4	4	1-12 OK	?
	XC YS	39	OK	max 11	4	4	1-12 OK	?
	XS YC	39	NOK	max 9	3	3	1-12 OK	?
	XS YS	39	NOK	max 9	3	4	1-12 OK	?
Abrasive	XC YC	8	OK	max 7	4	5	3-8 OK	4
	XC YS	8	OK	max 7	7	5	3-8 OK	6
	XS YC	8	NOK	max 8	4	?	3-8 OK	2
	XS YS	8	NOK	max 8	5	?	2-8 OK	5

Table 9.1: Latent Variable Selection Summary

The crossvalidation against residual plots clearly identified specific latent variables in all cases, except for the centred Wine Aroma dataset where there are two corners in the plot at 2 and 8 latent variables. The first minimum in RMSECV is at 3 latent variables and the overall RMSECV minimum is at 8 latent variables. The cause of the problem here has been identified, so the conclusion is that the method has functioned as well as might be expected. The crossvalidation method shows reasonable consistency in that only the latent variable selection for the Waste Glass dataset exceeds the over-fitted covariance criterion.

The covariance permutation method failed to select any latent variables for the Olive Oil dataset where the regressor variables are centred but not scaled and for the Abrasives dataset where the regressors are centred and scaled. In both cases, the covariance test statistics are within the permutation distributions for all latent variables. The R^2Q^2 plots do not identify specific latent variables, for most of the datasets and

scalings they do not even identify ranges of latent variables. Where covariance permutations select latent variables, they are all within the over-fitted covariance criterion for these example datasets. Wiklund et al[109] found good agreement between latent variable selection by RMSECV and covariance permutations for their example dataset, which is also observed with these examples. But the way PLS tends to find and inflate spurious covariances makes the application of permutation methods for latent variable selection questionable. The fact that covariance permutations succeed or fail to identify latent variables depending on the scaling is strong evidence for covariance permutation methods being influenced by this spurious covariance effect.

The information criteria have problems selecting the number of latent variables with both landscape datasets. For Gasoline, there is a sharp corner in the BIC values at 3 latent variables followed by a gradual decrease with increasing latent variables. So selecting 3 latent variables is at least reasonable. The Biscuits dataset information criteria plot in Figure 6.8 on page 88 selects very high numbers of latent variables with are clear overfitted. With the exception of the centred Wine Aroma dataset, all the latent variable selections by information criteria are within the over-fitted covariance criterion. If the over-fitted covariance limit of 5 latent variables is applied to this dataset then 2 latent variables are selected, which is the same as the selection by RMSECV and covariance permutations. The section of 2 latent variables for the regressor scaled and response centred Abrasives dataset is also an apparent anomaly. Inspection of the tabulated values here shows that 4 latent variables is also credible as it is the absolute minimum for AIC and a secondary minimum for BIC.

The overall conclusion from the example datasets must be that no single method of latent variable selection has proved to be completely reliable across all these datasets. The spurious covariance effect makes covariance permutations an unreliable method for PLS latent variable selection, because the scale of this effect is uncertain and any influence it may have cannot be detected. The traditional PLS method of crossvalidation has performed well when crossvalidation against residual plots are used to select the number of latent variables. The use of the plots does not make latent variable selection “foolproof”, but it is a lot less subjective than selection from inspection of

RMSECV against latent variable plots. Latent variable selection by RMSECV and information criteria has produced very similar selections. Any variations by more than one latent variable have been easily accounted for. Considering how close van der Voet's pseudo degrees of freedom are to those from numerical derivatives, this apparent close correspondence between latent variable selection from crossvalidation and information criteria may well be due to this connection. This may go some way to explain why the traditional method of crossvalidation is considered reliable in practice when there many good theoretical reasons to suggest that it should not be.

From the comparison of latent variable selection methods in the simulation, Tables 8.6 and 8.7 on pages 133 and 134 show that RMSECV minima methods have lower tendency to overfit than permutation methods. The information criteria method is best overall for PLS1 but worst for PLS2. A comparison of latent variable selection methods for the example datasets is summarized in Table 9.1 on page 160 shows that only latent variable selection for the PLS1 gasoline dataset is entirely consistent with the conclusions from the simulation. The three PLS2 example datasets do not show strong differences between the four latent variable selection methods, so do not provide evidence in support or against the conclusions from the simulation. Retrospectively, the waste glass PLS1 dataset is not typical of PLS datasets in that the regressor matrix is reduced rank only due to the mixtures constant row sum constraint. Otherwise, the regressor columns form a statistically balanced experimental design. This may be the reason why latent variable selection by RMSECV and Information Criteria methods overfit so badly here. The PLS2 mixtures abrasives dataset are also from a designed experiment but does not show this overfitting tendency, possibly because the strong correlations in the response matrix restricts the maximum number of latent variables. Latent variable selection for the wine aroma dataset remains an anomaly when compared to the conclusions of the simulation. Tables 8.6 and 8.7 show that differences between the latent variable selection methods, but there is considerable scope for under or overfitting within each method. Consequently, no conclusion can be inferred from the apparent anomolous behaviour of the wine aroma dataset. To obtain consistent conclusions from a comparison of latent variable selection methods between this simulation and example datasets, many mor samples of example datasets would be required.

Without referring to latent variable selection specifically, it is an open question wether an overall optimal PLS model exists where optimal coefficient identification and prediction performance coincide. The alternative is that PLS models need to be developed with prior knowledge of its application. The evidence from the simulation is that PLS1 coefficient identification and prediction RMSE suggest that the same model is optimal. It is accepted that this single study cannot prove the case, but a single example that shows the opposite would be conclusive.

9.3 Summary of Original Work and Discoveries

9.3.1 Informative Plots

- Crossvalidated Residuals against Fit Residuals Plots

  Plots of crossvalidated residuals deviation against the number of latent variables generally do not show a clear minimum point for selecting the best number of latent variables. Plots of crossvalidated residuals against fit residuals show a definite corner which indicates the best number of latent variables. These plots also have diagnostic value in identifying datasets where latent variable selection is unstable, or where the dataset is well balanced as an experimental design. Introduced in section 4.1 on page 53. Discussion with positive conclusions on page 61.

- Covariance Explained Plots

  The simulation suggests that overfitting may be a serious issue in developing PLS models. The Covariance Explained Plots described in section 7 on page 91 are a diagnostic plot that can detect overfitting. It is not a universal solution in that it can detect overfitting in all cases, but could well be a useful diagnostic tool for some datasets.

9.3.2 Latent Variable Selection Methods

- Randomised F and t-tests for Latent Variable Selection.
  These permutation tests based on crossvalidated sampling need to take the different means for each permutation sample into account, if not then the test is not valid. In section 5.1 on page 62.

- Extension of Permutation Test Methods for PLS2 Multivariate Responses.
  Details of the methods are given for covariance permutation tests in section 5.2 on page 63 and for R^2 and Q^2 permutation plots in section 5.3 on page 66. The conclusions from these tests for PLS2 example datasets on page 62 were that these multivariate results appeared to be more stable than the univariate results.

- Extension of Degrees of Freedom Calculation Methods
  The methods published to date are limited to PLS1 univariate responses with variates scaled to unit variance. These methods have been extended and generalised in a number of ways.

  - In section 6.1 on page 74 a minor correction to the published calculation for the upper bound on degrees of freedom in PLS1 is made.
  - Section 6.2 on page 76 shows the general form for extending the PLS1 univariate response methods to PLS2 multivariate responses.
  - In section 6.3 on page 79, the methods have been extended to any scaling method.
  - In section 6.5 on page 81 these calculated bounds are compared to the degrees of freedom calculated for the example sets. These results are reviewed later on page 86 after the degrees of freedom results have been examined in detail. The conclusion is that these bounds calculations are consistent with the example datasets. These theoretical limits are derived independently from any specific method for estimating the degrees of freedom.

- Estimates of pseudo-degrees of freedom calculated from fit and crossvalidated residuals by van der Voet's method can be very close to the degrees of freedom

calculated the numerical derivatives. Comparison between the two methods in section 6.5.2 on page 86 shows that the degrees of freedom from van der Voets's method tends to be an overestimate. As this could lead to overfitting it was concluded that using van der Voet's pseudo degrees of freedom for information criteria is not reliable. This conclusion is on page 88.

- Using numerical derivatives from fitted values for estimating degrees of freedom. Introduced in Section 6.4 on page 80. Algebraic solutions for degrees of freedom calculations are not available for most classes of PLS model. Numerical solution are proposed as a general solution for all classes of PLS model. For the example datasets where the algebraic solution can be calculated, numerical methods gave the same values as the algebraic solution within rounding error. The results for the example datasets are compared in section 6.5 on page 81 and found to be equivalent. Having a general solution to estimate degrees of freedom for all classes of PLS model allows information criteria to be applied to latent variable selection.

- In practice, the information criteria AIC and BIC appear to be equivalent measures for determining the number of latent variables through degrees of freedom. This suggests that PLS latent variable selection is a fairly coarse procedure compared to selecting terms in OLS stepwise regression for example.

- The best latent variable selection method for practical applications is probably to use the first minimum in RMSECV, due to the increased risk of adverse consequences from overfitting with other methods.

9.3.3 The PLS Simulation

- The methods presented to generate multivariate reduced rank datasets samples are more extensive than any previous published method. These methods have extended published univariate response methods to include more factors to define sample datasets. Dataset correlation and collinearity can be varied independently. The further extension to multivariate responses allows independent control of the structure of the regressor and response matrices. The distribution

of the regressor variables can be changed between three different categories, without changing their covariance matrices. As transforms that change regressor or response distributions usually also change the covariance, the methods presented here mean that regressor distribution is an independent factor. The sample matrices generated could be applied to a wide range of multivariate simulations as there is nothing specific to PLS in their structure.

- The important conclusion from the simulation is that overfitting in PLS is related to the structure of the dataset, specifically how the random errors relate to the space of the PLS model. This has also been concluded by some previous publications on PLS1 that use different approaches. The novel aspect here is that this conclusion comes from a simulation that covers a wide range of dataset characteristics over both PLS1 univariate and PLS2 multivariate response datasets. So this study provides new evidence that this conclusion may be universal to all conventional PLS models based the NIPALS algorithm and is not dependent on any subset of PLS structures.

- The distribution of the regressors and responses does not appear to have influence on the PLS model. This conclusion is contrary to published practice.

9.4 Open Questions and Further Work

Precisely what it is about the relation between the errors and model spaces that is the fundamental cause of overfitting needs to be identified. The paper by Cloarec[15] is a good start but is not a complete solution. Any solution here needs to include PLS2. This could also be investigated be extending this simulation to include the effects of random errors that are either within the model space or orthogonal to it.

Knowledge of the structural cause of over fitting could identify a specific PLS algorithm without the problem of overfitting, and so should replace algorithms like NIPALS as the standard. This is likely to be one of the variations on orthogonal PLS. Unfortunately, it is not possible to select an orthogonal method directly. Comments by Ergon[25] after examining a number of orthogonalization options are that it may not

be possible to combine all positive aspects within the same algorithm. So a "standard" PLS algorithm will have to be a compromise. So understanding the structural cause of overfitting may resolve this issue.

Appendix A

Notation

α	scalar	diagonal terms in diagonalization matrix $\mathbf{B}$
A	scalar	number of latent variables.
$\mathbf{B}$	$(k+1 \times k)$	diagonalisation matrix.
β	$(k \times m)$	regression coefficient matrix between the $\mathbf{X}$-block predictors and $\mathbf{Y}$-block responses.
$\mathbf{c}$	$(A \times 1)$	column vector containing the inner regression coefficients between the scores and response in PLS2.
$\mathbf{C}$	$(A \times A)$	diagonal matrix of inner regression coefficients between the scores and response in PLS2.
DoF	scalar	number of degrees of freedom.
$\mathbf{E}$	$(n \times k)$	$\mathbf{X}$-block error matrix.
$\mathbf{F}$	$(n \times m)$	$\mathbf{Y}$-block error matrix.
γ	scalar	off diagonal terms in diagonalization matrix $\mathbf{B}$
k	scalar	number of predictor variable columns in the $\mathbf{X}$ matrix.
λ	scalar	Lagrange's multiplier
m	scalar	number of response variable columns in the $\mathbf{Y}$ matrix.
$MSEP$	scalar	Mean squared error of prediction.
n	scalar	number of observation rows in both $\mathbf{X}$ and $\mathbf{Y}$ matrices.
$\mathbf{P}$	$(k \times A)$	$\mathbf{X}$-block loadings matrix.
$\mathbf{p}_i$	$(k \times 1)$	$\mathbf{X}$-block loadings column vector for the i^{th} latent variable.
$PRESS$	scalar	Prediction error sum of squares.
$\mathbf{Q}$	$(A \times m)$	$\mathbf{Y}$-block loadings matrix.
$\mathbf{Q}^2$	scalar	Regression coefficient, from response crossvalidated fit residuals.
$\mathbf{q}_i$	$(1 \times m)$	$\mathbf{Y}$-block loadings row vector for the i^{th} latent variable.
$\mathbf{R}^2$	scalar	Regression coefficient, from response fit residuals.
$\mathbf{r}$	$(k \times 1)$	from SIMPLS, left hand singular vector from SVD.
$RMSECV$	scalar	Root mean square error of crossvalidation.
$RMSE_{\mathbf{X}}$	scalar	Regressor residuals root mean square .
$RMSE_{\mathbf{Y}}$	scalar	Response residuals root mean square.
RSS	scalar	response fit residuals sum of squares.
$\mathbf{S_{XY}}$	$(k \times m)$	Sample covariance, $\mathbf{X}^T\mathbf{Y}$
$\mathbf{T}$	$(n \times A)$	$\mathbf{X}$-block predictor scores matrix.
$\mathbf{t}_i$	$(n \times 1)$	$\mathbf{X}$-block predictor scores column vector for the i^{th} latent variable.
$\mathbf{U}$	$(n \times A)$	$\mathbf{Y}$-block response scores matrix.
$\mathbf{u}_i$	$(n \times 1)$	$\mathbf{Y}$-block response scores column vector for the i^{th} latent variable.
$\boldsymbol{v}_1$	$(k \times 1)$	from SIMPLS, left hand column vector from SVD.
$\mathbf{V}_A$	$(k \times A)$	set of orthonormal basis vectors $\mathbf{v}_i$ for LVs 1 to A.
$\mathbf{v}_i$	$(k \times 1)$	orthonormal basis of the $\mathbf{X}$-block loadings $\mathbf{p}_i$
VIP	scalar	Variable importance in projection factor.
$\mathbf{W}$	$(k \times A)$	$\mathbf{X}$-block weighting matrix.
$\mathbf{w}_i$	$(k \times 1)$	$\mathbf{X}$-block weighting column vector for the i^{th} latent variable.
$\mathbf{X}$	$(n \times k)$	predictor variables matrix.
$\mathbf{Y}$	$(n \times m)$	response variables matrix.

Appendix B

Latent Variable Selection Logistic Models

B.1 PLS1 RMSECV 1st minimum Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(RMSECV_first_min_LVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC.DATA)
> fmnull$logLik
[1] -18952.65

> fm1 <- clm(ordered(RMSECV_first_min_LVlessAstar) ~
                  zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,
                  link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm1)
formula:
ordered(RMSECV_first_min_LVlessAstar) ~
                zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  8905 -11245.11 22548.21 9(0)  6.16e-13 3.9e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.11236    0.02055   5.466 4.59e-08 ***
zk                  0.30150    0.02073  14.545  < 2e-16 ***
zAstar             -4.48393    0.05370 -83.498  < 2e-16 ***
zSNR_X              0.35466    0.02077  17.076  < 2e-16 ***
zSNR_Y              0.30868    0.02075  14.878  < 2e-16 ***
VDRMedium           0.05346    0.04953   1.079  0.28040
```

```
VDRSlow            -0.67138    0.05022 -13.369  < 2e-16 ***
PIRLow              0.97337    0.05102  19.080  < 2e-16 ***
PIRMedium           0.64751    0.05010  12.926  < 2e-16 ***
DistributionNormal -0.14376    0.04920  -2.922  0.00348 **
DistributionSkew    0.04776    0.04982   0.959  0.33774
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
       Estimate Std. Error z value
-8|-7 -10.73820    0.22099  -48.59
-7|-6  -8.14401    0.11019  -73.91
-6|-5  -6.33522    0.09261  -68.41
-5|-4  -4.87946    0.08335  -58.54
-4|-3  -3.09767    0.07206  -42.99
-3|-2  -1.24957    0.06574  -19.01
-2|-1   0.73902    0.06391   11.56
-1|0    3.09164    0.07228   42.77
0|1     7.21234    0.09927   72.65
1|2     8.46816    0.11403   74.27
2|3     9.48019    0.14300   66.29
3|4    10.17600    0.17947   56.70
4|5    10.70036    0.22000   48.64
5|6    10.97643    0.24704   44.43
6|7    11.52528    0.31513   36.57
7|8    11.97878    0.38891   30.80
8|9    13.23408    0.71302   18.56
9|11   13.92799    1.00419   13.87
(95 observations deleted due to missingness)
> ci1 <- confint(fm1)
                         2.5 %      97.5 %
zNobs                0.07209052  0.15266625
zk                   0.26091175  0.34216848
zAstar              -4.58999290 -4.37947848
zSNR_X               0.31400792  0.39542441
zSNR_Y               0.26806214  0.34939302
VDRMedium           -0.04360885  0.15053778
VDRSlow             -0.76989980 -0.57303574
PIRLow               0.87350455  1.07348977
PIRMedium            0.54941338  0.74579050
DistributionNormal -0.24020988 -0.04733826
DistributionSkew   -0.04988216  0.14540704

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.4066735

sum(is.na(SIM_DATA_ABC.DATA$RMSECV_first_min_LVlessAstar)==FALSE)
[1] 8095
> klogn <- log(8905)
```

```
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)  % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm2)
formula:
ordered(RMSECV_first_min_LVlessAstar) ~
         zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution + VDR:Distribution + zk:zAstar
         + zAstar:VDR + VDR:PIR + zAstar:zSNR_X + zAstar:Distribution + zNobs:zk + PIR:Distribution
         + zAstar:PIR + zAstar:zSNR_Y + zk:zSNR_X + zSNR_X:PIR + zk:VDR + zNobs:zAstar + zk:PIR
data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  8905 -10739.34 21596.68 9(0)  4.08e-12 6.6e+03

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
zNobs                        0.137364   0.021194   6.481 9.10e-11 ***
zk                           0.409447   0.046922   8.726  < 2e-16 ***
zAstar                      -4.963549   0.079883 -62.135  < 2e-16 ***
zSNR_X                       0.213625   0.036197   5.902 3.60e-09 ***
zSNR_Y                       0.333417   0.021277  15.670  < 2e-16 ***
VDRMedium                   -0.408340   0.116713  -3.499 0.000468 ***
VDRSlow                     -0.389297   0.115391  -3.374 0.000742 ***
PIRLow                       1.603747   0.119818  13.385  < 2e-16 ***
PIRMedium                    0.756911   0.111416   6.794 1.09e-11 ***
DistributionNormal           0.024665   0.109613   0.225 0.821968
DistributionSkew            -0.353602   0.119193  -2.967 0.003011 **
VDRMedium:DistributionNormal 0.284565   0.125742   2.263 0.023630 *
VDRSlow:DistributionNormal  -0.885677   0.123163  -7.191 6.43e-13 ***
VDRMedium:DistributionSkew   0.976406   0.127034   7.686 1.52e-14 ***
VDRSlow:DistributionSkew     1.440886   0.130068  11.078  < 2e-16 ***
zk:zAstar                    0.257871   0.019815  13.014  < 2e-16 ***
zAstar:VDRMedium             0.310766   0.053003   5.863 4.54e-09 ***
zAstar:VDRSlow              -0.174642   0.051968  -3.361 0.000778 ***
VDRMedium:PIRLow            -0.237093   0.125434  -1.890 0.058734 .
VDRSlow:PIRLow              -1.025000   0.126158  -8.125 4.48e-16 ***
VDRMedium:PIRMedium          0.393231   0.125523   3.133 0.001732 **
VDRSlow:PIRMedium           -0.311375   0.124157  -2.508 0.012145 *
zAstar:zSNR_X                0.172647   0.020274   8.516  < 2e-16 ***
zAstar:DistributionNormal   -0.262258   0.051303  -5.112 3.19e-07 ***
zAstar:DistributionSkew      0.086706   0.053818   1.611 0.107159
zNobs:zk                    -0.147177   0.020356  -7.230 4.83e-13 ***
PIRLow:DistributionNormal    0.294715   0.125168   2.355 0.018545 *
PIRMedium:DistributionNormal 0.003564   0.122545   0.029 0.976796
PIRLow:DistributionSkew     -0.715467   0.128380  -5.573 2.50e-08 ***
PIRMedium:DistributionSkew  -0.367126   0.127307  -2.884 0.003929 **
zAstar:PIRLow                0.390593   0.053839   7.255 4.02e-13 ***
zAstar:PIRMedium             0.157907   0.052071   3.033 0.002425 **
zAstar:zSNR_Y                0.127560   0.020197   6.316 2.69e-10 ***
```

```
zk:zSNR_X                    -0.104556   0.020311  -5.148 2.64e-07 ***
zSNR_X:PIRLow                 0.130473   0.052240   2.498 0.012504 *
zSNR_X:PIRMedium              0.256349   0.050294   5.097 3.45e-07 ***
zk:VDRMedium                  0.107325   0.053109   2.021 0.043293 *
zk:VDRSlow                   -0.163099   0.051391  -3.174 0.001505 **
zNobs:zAstar                  0.077282   0.019805   3.902 9.53e-05 ***
zk:PIRLow                    -0.229666   0.052003  -4.416 1.00e-05 ***
zk:PIRMedium                 -0.101124   0.051946  -1.947 0.051571 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -11.6845     0.2429 -48.102
-7|-6  -8.9310     0.1447 -61.720
-6|-5  -6.8004     0.1274 -53.397
-5|-4  -5.0869     0.1176 -43.270
-4|-3  -3.1672     0.1081 -29.298
-3|-2  -1.2094     0.1029 -11.757
-2|-1   0.9172     0.1019   9.005
-1|0    3.4318     0.1091  31.462
0|1     7.6521     0.1297  59.010
1|2     8.9136     0.1414  63.023
2|3     9.9263     0.1658  59.886
3|4    10.6228     0.1981  53.618
4|5    11.1473     0.2355  47.341
5|6    11.4234     0.2609  43.782
6|7    11.9721     0.3261  36.711
7|8    12.4255     0.3979  31.230
8|9    13.6808     0.7179  19.056
9|11   14.3745     1.0077  14.265
(95 observations deleted due to missingness)
> ci2 <- confint(fm2)
                                    2.5 %        97.5 %
zNobs                          0.095838838  0.1789226601
zk                             0.317435326  0.5013785906
zAstar                        -5.120877049 -4.8077254258
zSNR_X                         0.142711589  0.2846081938
zSNR_Y                         0.291760435  0.3751686134
VDRMedium                     -0.637165154 -0.1796355391
VDRSlow                       -0.615558118 -0.1632066578
PIRLow                         1.369005425  1.8387150134
PIRMedium                      0.538607351  0.9753765049
DistributionNormal            -0.190217215  0.2394871747
DistributionSkew              -0.587375803 -0.1201263188
VDRMedium:DistributionNormal   0.038126391  0.5310476438
VDRSlow:DistributionNormal    -1.127221900 -0.6444096622
VDRMedium:DistributionSkew     0.727593881  1.2255789839
VDRSlow:DistributionSkew       1.186185398  1.6960653643
```

```
zk:zAstar                      0.219108707  0.2967883695
zAstar:VDRMedium               0.206934068  0.4147151621
zAstar:VDRSlow                -0.276532792 -0.0728094057
VDRMedium:PIRLow              -0.483000361  0.0087146894
VDRSlow:PIRLow                -1.272416517 -0.7778624777
VDRMedium:PIRMedium            0.147239848  0.6393021957
VDRSlow:PIRMedium             -0.554797321 -0.0680895031
zAstar:zSNR_X                  0.132936596  0.2124140277
zAstar:DistributionNormal     -0.362899971 -0.1617855521
zAstar:DistributionSkew       -0.018764457  0.1922118508
zNobs:zk                      -0.187105560 -0.1073058632
PIRLow:DistributionNormal      0.049453463  0.5401239046
PIRMedium:DistributionNormal -0.236637541  0.2437516949
PIRLow:DistributionSkew       -0.967189571 -0.4639286265
PIRMedium:DistributionSkew    -0.616702767 -0.1176448259
zAstar:PIRLow                  0.285074137  0.4961333096
zAstar:PIRMedium               0.055822921  0.2599483577
zAstar:zSNR_Y                  0.087993327  0.1671682543
zk:zSNR_X                     -0.144372868 -0.0647499553
zSNR_X:PIRLow                  0.028102523  0.2328870664
zSNR_X:PIRMedium               0.157804367  0.3549625549
zk:VDRMedium                   0.003258043  0.2114555199
zk:VDRSlow                    -0.263849379 -0.0623897915
zNobs:zAstar                   0.038476231  0.1161140105
zk:PIRLow                     -0.331634370 -0.1277769189
zk:PIRMedium                  -0.202997605  0.0006391721

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.4333595
```

B.2 PLS1 RMSECV Absolute minimum Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(RMSECV_abs_min_LVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC.DATA)
> fmnull$logLik
[1] -20253.54

> fm1 <- clm(ordered(RMSECV_abs_min_LVlessAstar) ~
                zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,
                link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm1)
formula:
ordered(RMSECV_abs_min_LVlessAstar) ~
                   zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -14737.90 29539.80 10(2) 1.33e-12 7.3e+02

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.04226    0.01905   2.218  0.02656 *
zk                  0.26208    0.01941  13.503  < 2e-16 ***
zAstar             -3.11003    0.03869 -80.391  < 2e-16 ***
zSNR_X              0.27973    0.01917  14.591  < 2e-16 ***
zSNR_Y              0.22379    0.01912  11.707  < 2e-16 ***
VDRMedium           0.14667    0.04619   3.175  0.00150 **
VDRSlow            -0.39238    0.04643  -8.452  < 2e-16 ***
PIRLow              0.69036    0.04720  14.625  < 2e-16 ***
PIRMedium           0.45837    0.04653   9.850  < 2e-16 ***
DistributionNormal -0.19169    0.04613  -4.155 3.25e-05 ***
DistributionSkew    0.13661    0.04653   2.936  0.00333 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -9.31155    0.26812 -34.730
-7|-6 -6.46913    0.09761 -66.276
-6|-5 -4.83747    0.07642 -63.302
-5|-4 -3.64417    0.06829 -53.364
-4|-3 -2.30287    0.06127 -37.586
-3|-2 -1.01664    0.05728 -17.749
-2|-1  0.30883    0.05622   5.493
```

```
-1|0   1.95139    0.06039  32.314
0|1    4.76923    0.07443  64.077
1|2    5.57005    0.08010  69.535
2|3    6.04596    0.08565  70.590
3|4    6.57965    0.09501  69.251
4|5    6.79983    0.10019  67.868
5|6    7.00216    0.10577  66.201
6|7    7.51963    0.12418  60.553
7|8    7.89497    0.14203  55.587
8|9    8.14072    0.15620  52.119
9|10   8.64333    0.19248  44.905
10|11  8.86001    0.21164  41.864
11|12  8.98859    0.22415  40.100
12|13  9.44289    0.27607  34.204

> ci1 <- confint(fm1)
> ci1
                          2.5 %      97.5 %
zNobs                0.004918429  0.07960691
zk                   0.224081697  0.30016563
zAstar              -3.186323276 -3.03466986
zSNR_X               0.242185416  0.31734144
zSNR_Y               0.186346617  0.26128183
VDRMedium            0.056141683  0.23722565
VDRSlow             -0.483401932 -0.30141020
PIRLow               0.597901753  0.78294006
PIRMedium            0.367214908  0.54962832
DistributionNormal -0.282134066 -0.10128282
DistributionSkew     0.045415835  0.22782555

> 1-fm1$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.2723297

> sum(is.na(SIM_DATA_ABC.DATA$RMSECV_abs_min_LVlessAstar)==FALSE)
[1] 9000
> klogn <- log(9000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn) % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm2)
formula:
ordered(RMSECV_abs_min_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
      + zk:zAstar + VDR:Distribution + zAstar:zSNR_X + zAstar:VDR + zAstar:PIR + zAstar:zSNR_Y
      + zk:zSNR_X + zAstar:Distribution + VDR:PIR + zNobs:zk + PIR:Distribution + zNobs:zAstar
      + zk:VDR + zk:PIR
data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -14377.30 28874.60 10(2) 8.57e-13 1.7e+03
```

```
Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
zNobs                         0.05957    0.01942   3.068 0.002155 **
zk                            0.35522    0.04300   8.261  < 2e-16 ***
zAstar                       -3.47266    0.06458 -53.773  < 2e-16 ***
zSNR_X                        0.28538    0.01953  14.616  < 2e-16 ***
zSNR_Y                        0.24766    0.01939  12.772  < 2e-16 ***
VDRMedium                    -0.04416    0.10798  -0.409 0.682545
VDRSlow                      -0.12108    0.10699  -1.132 0.257771
PIRLow                        1.10253    0.10993  10.029  < 2e-16 ***
PIRMedium                     0.66752    0.10270   6.500 8.05e-11 ***
DistributionNormal           -0.05263    0.10134  -0.519 0.603526
DistributionSkew             -0.07292    0.10910  -0.668 0.503909
zk:zAstar                     0.25761    0.01800  14.313  < 2e-16 ***
VDRMedium:DistributionNormal  0.13955    0.11638   1.199 0.230483
VDRSlow:DistributionNormal   -0.52784    0.11415  -4.624 3.77e-06 ***
VDRMedium:DistributionSkew    0.57265    0.11688   4.899 9.61e-07 ***
VDRSlow:DistributionSkew      0.87667    0.11876   7.382 1.56e-13 ***
zAstar:zSNR_X                 0.16827    0.01854   9.077  < 2e-16 ***
zAstar:VDRMedium              0.23350    0.04820   4.844 1.27e-06 ***
zAstar:VDRSlow               -0.16656    0.04743  -3.512 0.000445 ***
zAstar:PIRLow                 0.37468    0.04909   7.632 2.31e-14 ***
zAstar:PIRMedium              0.14146    0.04743   2.982 0.002859 **
zAstar:zSNR_Y                 0.10688    0.01840   5.807 6.35e-09 ***
zk:zSNR_X                    -0.10646    0.01891  -5.629 1.81e-08 ***
zAstar:DistributionNormal    -0.13343    0.04703  -2.837 0.004551 **
zAstar:DistributionSkew       0.10851    0.04886   2.221 0.026358 *
VDRMedium:PIRLow             -0.16447    0.11623  -1.415 0.157064
VDRSlow:PIRLow               -0.75785    0.11680  -6.489 8.66e-11 ***
VDRMedium:PIRMedium           0.13021    0.11649   1.118 0.263677
VDRSlow:PIRMedium            -0.41123    0.11522  -3.569 0.000358 ***
zNobs:zk                     -0.08386    0.01896  -4.422 9.76e-06 ***
PIRLow:DistributionNormal     0.23335    0.11724   1.990 0.046539 *
PIRMedium:DistributionNormal -0.16297    0.11428  -1.426 0.153850
PIRLow:DistributionSkew      -0.43742    0.11930  -3.667 0.000246 ***
PIRMedium:DistributionSkew   -0.21591    0.11700  -1.845 0.064972 .
zNobs:zAstar                  0.06180    0.01807   3.420 0.000627 ***
zk:VDRMedium                  0.14098    0.04926   2.862 0.004212 **
zk:VDRSlow                   -0.08599    0.04732  -1.817 0.069189 .
zk:PIRLow                    -0.20680    0.04879  -4.238 2.25e-05 ***
zk:PIRMedium                 -0.12773    0.04785  -2.669 0.007599 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -9.86316    0.28181 -34.999
-7|-6 -6.89777    0.12723 -54.216
```

```
-6|-5 -5.03854    0.10851 -46.432
-5|-4 -3.67464    0.10115 -36.329
-4|-3 -2.23089    0.09527 -23.417
-3|-2 -0.87436    0.09221  -9.482
-2|-1  0.51555    0.09181   5.615
-1|0   2.22984    0.09538  23.379
0|1    5.08150    0.10532  48.248
1|2    5.88620    0.10941  53.799
2|3    6.36411    0.11351  56.066
3|4    6.89903    0.12071  57.155
4|5    7.11889    0.12481  57.036
5|6    7.32096    0.12932  56.609
6|7    7.83778    0.14475  54.147
7|8    8.21265    0.16031  51.229
8|9    8.45813    0.17298  48.895
9|10   8.96044    0.20633  43.428
10|11  9.17712    0.22430  40.914
11|12  9.30580    0.23615  39.407
12|13  9.76068    0.28589  34.141

> ci2 <- confint(fm2)
> ci2
                                      2.5 %       97.5 %
zNobs                          0.021516687  0.097628260
zk                             0.270978097  0.439545041
zAstar                        -3.599609972 -3.346447336
zSNR_X                         0.247143303  0.323684524
zSNR_Y                         0.209681830  0.285692316
VDRMedium                     -0.255831813  0.167475241
VDRSlow                       -0.330832641  0.088602788
PIRLow                         0.887093515  1.318038114
PIRMedium                      0.466215491  0.868815637
DistributionNormal            -0.251344868  0.145916385
DistributionSkew              -0.286867526  0.140836409
zk:zAstar                      0.222385083  0.292943035
VDRMedium:DistributionNormal -0.088557634  0.367644251
VDRSlow:DistributionNormal   -0.751659424 -0.304164513
VDRMedium:DistributionSkew    0.343641783  0.801823523
VDRSlow:DistributionSkew      0.643983546  1.109542787
zAstar:zSNR_X                  0.131962731  0.204630510
zAstar:VDRMedium               0.139035503  0.327995460
zAstar:VDRSlow                -0.259552010 -0.073637717
zAstar:PIRLow                  0.278498365  0.470946423
zAstar:PIRMedium               0.048494052  0.234426425
zAstar:zSNR_Y                  0.070821804  0.142967705
zk:zSNR_X                     -0.143543937 -0.069404551
zAstar:DistributionNormal     -0.225629984 -0.041276959
zAstar:DistributionSkew        0.012758330  0.204291121
VDRMedium:PIRLow              -0.392323550  0.063306048
```

```
VDRSlow:PIRLow               -0.986849841 -0.528998605
VDRMedium:PIRMedium          -0.098138064  0.358524573
VDRSlow:PIRMedium            -0.637127247 -0.185457029
zNobs:zk                     -0.121032902 -0.046697825
PIRLow:DistributionNormal     0.003617685  0.463193222
PIRMedium:DistributionNormal -0.386983893  0.061007830
PIRLow:DistributionSkew      -0.671291297 -0.203635060
PIRMedium:DistributionSkew   -0.445249619  0.013394109
zNobs:zAstar                  0.026384451  0.097224194
zk:VDRMedium                  0.044474831  0.237591802
zk:VDRSlow                   -0.178741533  0.006761162
zk:PIRLow                    -0.302461913 -0.111182876
zk:PIRMedium                 -0.221557382 -0.033972102

> 1-fm1$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.290134
```

B.3 PLS1 Permutation minimum Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(perms_best_LVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC.DATA)
> fmnull$logLik
[1] -20040.99

> fm1 <- clm(ordered(perms_best_LVlessAstar) ~
                zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,
                link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm1)
formula:
ordered(perms_best_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  8975 -11478.19 23008.38 9(0)  4.26e-13 4.0e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.58279    0.02102  27.730  < 2e-16 ***
zk                  0.72018    0.02302  31.280  < 2e-16 ***
zAstar             -4.88730    0.05618 -86.991  < 2e-16 ***
zSNR_X              0.24279    0.02020  12.020  < 2e-16 ***
zSNR_Y              0.26060    0.02023  12.883  < 2e-16 ***
VDRMedium          -1.43766    0.05126 -28.045  < 2e-16 ***
VDRSlow            -1.83783    0.05306 -34.638  < 2e-16 ***
PIRLow              0.55193    0.04896  11.273  < 2e-16 ***
PIRMedium           0.64185    0.04844  13.250  < 2e-16 ***
DistributionNormal -0.05449    0.04811  -1.132 0.257452
DistributionSkew   -0.17511    0.04872  -3.594 0.000325 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
       Estimate Std. Error z value
-9|-8 -12.56901    0.20488 -61.349
-8|-7  -9.22382    0.11734 -78.607
-7|-6  -7.35380    0.10255 -71.709
-6|-5  -5.92724    0.09109 -65.069
-5|-4  -4.31141    0.07918 -54.451
-4|-3  -2.48147    0.06902 -35.952
-3|-2  -0.62357    0.06374  -9.784
-2|-1   1.23229    0.06452  19.099
-1|0    3.27169    0.07493  43.662
0|1     7.43459    0.10211  72.807
1|2    10.12239    0.18549  54.571
```

```
2|3    11.30308    0.30347  37.246
3|4    12.17857    0.45690  26.655
4|5    13.09487    0.71328  18.359
5|6    13.78804    1.00437  13.728
(25 observations deleted due to missingness)

> ci1 <- confint(fm1)
> ci1
                        2.5 %      97.5 %
zNobs                0.5416772  0.62406408
zk                   0.6751846  0.76544106
zAstar              -4.9981988 -4.77796266
zSNR_X               0.2032292  0.28240813
zSNR_Y               0.2209877  0.30028225
VDRMedium           -1.5383374 -1.33738067
VDRSlow             -1.9420807 -1.73408890
PIRLow               0.4560377  0.64796027
PIRMedium            0.5469884  0.73688060
DistributionNormal -0.1487875  0.03982164
DistributionSkew   -0.2706262 -0.07963713

> 1-fm1$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.4272643

> sum(is.na(SIM_DATA_ABC.DATA$perms_best_LVlessAstar)==FALSE)
[1] 8975
> klogn <- log(8975)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn) % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm2)
formula:
ordered(perms_best_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
   + zAstar:VDR + VDR:Distribution + zk:zAstar + zNobs:zAstar + VDR:PIR + zk:VDR + zk:zSNR_X + zk:PIR
   + zk:Distribution + zAstar:Distribution + zAstar:zSNR_X + zAstar:zSNR_Y + zSNR_X:Distribution
    + zNobs:zk + zNobs:Distribution + zk:zSNR_Y + zAstar:PIR + zNobs:zSNR_X
data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  8975 -10437.32 20990.63 9(0)  3.88e-13 6.1e+03

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
zNobs                       0.63831    0.03853  16.567  < 2e-16 ***
zk                          1.11946    0.06406  17.476  < 2e-16 ***
zAstar                     -4.68450    0.07849 -59.686  < 2e-16 ***
zSNR_X                      0.12793    0.03667   3.488 0.000486 ***
zSNR_Y                      0.31398    0.02145  14.638  < 2e-16 ***
VDRMedium                  -1.63405    0.12075 -13.533  < 2e-16 ***
```

```
VDRSlow                       -1.81282    0.11954 -15.164  < 2e-16 ***
PIRLow                         1.63955    0.09515  17.231  < 2e-16 ***
PIRMedium                      1.15142    0.09089  12.668  < 2e-16 ***
DistributionNormal             0.13841    0.09081   1.524 0.127470
DistributionSkew              -1.53225    0.09625 -15.919  < 2e-16 ***
zAstar:VDRMedium              -0.98818    0.05607 -17.625  < 2e-16 ***
zAstar:VDRSlow                -1.32901    0.05651 -23.516  < 2e-16 ***
VDRMedium:DistributionNormal -0.13004    0.12670  -1.026 0.304744
VDRSlow:DistributionNormal    -0.62779    0.12444  -5.045 4.54e-07 ***
VDRMedium:DistributionSkew     1.39475    0.12800  10.897  < 2e-16 ***
VDRSlow:DistributionSkew       2.29518    0.13317  17.235  < 2e-16 ***
zk:zAstar                      0.45003    0.02157  20.864  < 2e-16 ***
zNobs:zAstar                   0.29650    0.02068  14.338  < 2e-16 ***
VDRMedium:PIRLow              -1.07150    0.12698  -8.438  < 2e-16 ***
VDRSlow:PIRLow                -1.66756    0.13008 -12.820  < 2e-16 ***
VDRMedium:PIRMedium           -0.27528    0.12519  -2.199 0.027884 *
VDRSlow:PIRMedium             -0.90322    0.12660  -7.134 9.74e-13 ***
zk:VDRMedium                  -0.52637    0.05971  -8.815  < 2e-16 ***
zk:VDRSlow                    -0.53208    0.05693  -9.346  < 2e-16 ***
zk:zSNR_X                     -0.19191    0.02176  -8.820  < 2e-16 ***
zk:PIRLow                      0.09345    0.05483   1.704 0.088312 .
zk:PIRMedium                   0.39048    0.05640   6.923 4.42e-12 ***
zk:DistributionNormal         -0.29396    0.05316  -5.530 3.20e-08 ***
zk:DistributionSkew            0.03467    0.05691   0.609 0.542373
zAstar:DistributionNormal     -0.23031    0.05216  -4.415 1.01e-05 ***
zAstar:DistributionSkew        0.11631    0.05466   2.128 0.033360 *
zAstar:zSNR_X                  0.12266    0.02044   6.001 1.96e-09 ***
zAstar:zSNR_Y                  0.11045    0.02067   5.343 9.16e-08 ***
zSNR_X:DistributionNormal      0.26284    0.05070   5.184 2.17e-07 ***
zSNR_X:DistributionSkew        0.04210    0.05251   0.802 0.422744
zNobs:zk                       0.10629    0.02190   4.853 1.21e-06 ***
zNobs:DistributionNormal      -0.12617    0.05110  -2.469 0.013550 *
zNobs:DistributionSkew         0.16060    0.05388   2.981 0.002874 **
zk:zSNR_Y                      0.08330    0.02259   3.687 0.000227 ***
zAstar:PIRLow                 -0.15516    0.05475  -2.834 0.004598 **
zAstar:PIRMedium               0.09191    0.05262   1.747 0.080716 .
zNobs:zSNR_X                  -0.06072    0.02003  -3.031 0.002435 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
       Estimate Std. Error z value
-9|-8 -14.92436    0.24156 -61.783
-8|-7 -10.95353    0.15805 -69.306
-7|-6  -8.38229    0.13564 -61.799
-6|-5  -6.60076    0.12266 -53.815
-5|-4  -4.71443    0.11159 -42.249
-4|-3  -2.61308    0.09964 -26.226
-3|-2  -0.54166    0.09494  -5.706
```

```
-2|-1   1.62122    0.09607  16.875
-1|0    4.01244    0.10660  37.639
0|1     8.01990    0.12787  62.719
1|2    10.70672    0.20121  53.211
2|3    11.88829    0.31333  37.942
3|4    12.76377    0.46351  27.537
4|5    13.68008    0.71752  19.066
5|6    14.37325    1.00739  14.268
(25 observations deleted due to missingness)

> ci2 <- confint(fm2)
> ci2
                                      2.5 %      97.5 %
zNobs                          0.562899631  0.71394038
zk                             0.994382152  1.24546974
zAstar                        -4.839037509 -4.53136630
zSNR_X                         0.056056833  0.19982656
zSNR_Y                         0.271985357  0.35607150
VDRMedium                     -1.871033107 -1.39768108
VDRSlow                       -2.047523169 -1.57889690
PIRLow                         1.453230751  1.82623272
PIRMedium                      0.973429609  1.32974277
DistributionNormal            -0.039599262  0.31640413
DistributionSkew              -1.721114065 -1.34379386
zAstar:VDRMedium              -1.098197052 -0.87840426
zAstar:VDRSlow                -1.439966998 -1.21842323
VDRMedium:DistributionNormal -0.378401456  0.11829344
VDRSlow:DistributionNormal   -0.871775566 -0.38395867
VDRMedium:DistributionSkew    1.144054377  1.64582254
VDRSlow:DistributionSkew      2.034464766  2.55650657
zk:zAstar                      0.407804229  0.49235874
zNobs:zAstar                   0.256011793  0.33708002
VDRMedium:PIRLow              -1.320514338 -0.82273081
VDRSlow:PIRLow                -1.922726181 -1.41280342
VDRMedium:PIRMedium           -0.520687342 -0.02993741
VDRSlow:PIRMedium             -1.151511250 -0.65521143
zk:VDRMedium                  -0.643592562 -0.40950690
zk:VDRSlow                    -0.643848753 -0.42066763
zk:zSNR_X                     -0.234576840 -0.14928344
zk:PIRLow                     -0.014008021  0.20093604
zk:PIRMedium                   0.279973857  0.50108248
zk:DistributionNormal         -0.398211433 -0.18982502
zk:DistributionSkew           -0.076834404  0.14625384
zAstar:DistributionNormal     -0.332610188 -0.12811797
zAstar:DistributionSkew        0.009187442  0.22347282
zAstar:zSNR_X                  0.082632959  0.16275785
zAstar:zSNR_Y                  0.069956294  0.15099489
zSNR_X:DistributionNormal      0.163506376  0.36226985
zSNR_X:DistributionSkew       -0.060816690  0.14504073
```

```
zNobs:zk                       0.063384982  0.14923766
zNobs:DistributionNormal      -0.226347865 -0.02602908
zNobs:DistributionSkew         0.055008075  0.26620661
zk:zSNR_Y                      0.039041571  0.12761707
zAstar:PIRLow                 -0.262542467 -0.04790917
zAstar:PIRMedium              -0.011258530  0.19503432
zNobs:zSNR_X                  -0.099989638 -0.02146699

> 1-fm1$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.4792014
```

B.4 PLS1 Information Criteria Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(Info_BIC_min_LVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC.DATA)
> fmnull$logLik
[1] -24726.88

> fm1 <- clm(ordered(Info_BIC_min_LVlessAstar)) ~
                zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,
                link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm1)
formula:
ordered(Info_BIC_min_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -21651.17 43366.34 7(0)  2.71e-09 8.5e+02

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs              -0.17527    0.01857  -9.436  < 2e-16 ***
zk                  1.32924    0.02403  55.311  < 2e-16 ***
zAstar             -1.36357    0.02475 -55.097  < 2e-16 ***
zSNR_X              0.08978    0.01837   4.886 1.03e-06 ***
zSNR_Y              0.06817    0.01834   3.717 0.000202 ***
VDRMedium           0.17177    0.04446   3.864 0.000112 ***
VDRSlow            -0.07882    0.04459  -1.768 0.077113 .
PIRLow              0.49743    0.04514  11.018  < 2e-16 ***
PIRMedium           0.31070    0.04481   6.934 4.09e-12 ***
DistributionNormal  0.00837    0.04452   0.188 0.850876
DistributionSkew   -0.04221    0.04454  -0.948 0.343292
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -7.27447    0.23057 -31.550
-7|-6 -4.66903    0.08475 -55.094
-6|-5 -3.45335    0.06615 -52.202
-5|-4 -2.64582    0.05926 -44.646
-4|-3 -1.73560    0.05424 -31.999
-3|-2 -1.05790    0.05195 -20.365
-2|-1 -0.37094    0.05082  -7.300
-1|0   0.44128    0.05132   8.599
0|1    1.57735    0.05397  29.227
1|2    1.87543    0.05502  34.086
2|3    2.23200    0.05642  39.559
```

```
3|4    2.65380    0.05828  45.536
4|5    3.01840    0.06012  50.209
5|6    3.39314    0.06239  54.385
6|7    3.92825    0.06672  58.876
7|8    4.24355    0.07005  60.576
8|9    4.60311    0.07490  61.459
9|10   4.97217    0.08136  61.112
10|11  5.33485    0.08953  59.585
11|12  5.68022    0.09949  57.095
12|13  6.24998    0.12171  51.351

> ci1 <- confint(fm1)
> ci1
                          2.5 %       97.5 %
zNobs               -0.21169677 -0.138884940
zk                   1.28234471  1.376552045
zAstar              -1.41225301 -1.315237690
zSNR_X               0.05377384  0.125801983
zSNR_Y               0.03222945  0.104119510
VDRMedium            0.08464695  0.258926236
VDRSlow             -0.16621897  0.008580214
PIRLow               0.40897590  0.585946821
PIRMedium            0.22289943  0.398551584
DistributionNormal -0.07888706  0.095629915
DistributionSkew   -0.12951203  0.045091552

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.1243873

> sum(is.na(SIM_DATA_ABC.DATA$Info_BIC_min_LVlessAstar)==FALSE)
[1] 9000
> klogn <- log(9000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)  % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC.DATA)

> summary(fm2)
formula:
ordered(Info_BIC_min_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
   + zk:zAstar + zNobs:zk + zAstar:VDR + zAstar:zSNR_X + zNobs:zAstar + zSNR_Y:Distribution
   + zAstar:Distribution + zk:zSNR_X + zk:VDR + zNobs:VDR + zAstar:PIR

data:    SIM_DATA_ABC.DATA

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -20726.67 41551.33 7(0)  1.83e-08 9.1e+02

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
zNobs                   -0.08541    0.03307  -2.583  0.00979 **
```

```
zk                          1.33748    0.03466  38.589  < 2e-16 ***
zAstar                     -2.13325    0.05748 -37.111  < 2e-16 ***
zSNR_X                      0.14788    0.01874   7.891 3.01e-15 ***
zSNR_Y                     -0.02961    0.03246  -0.912  0.36163
VDRMedium                   0.26032    0.04567   5.701 1.19e-08 ***
VDRSlow                    -0.06370    0.04569  -1.394  0.16328
PIRLow                      0.56433    0.04592  12.288  < 2e-16 ***
PIRMedium                   0.39501    0.04561   8.661  < 2e-16 ***
DistributionNormal          0.02459    0.04525   0.543  0.58685
DistributionSkew           -0.01610    0.04521  -0.356  0.72176
zk:zAstar                   0.58240    0.01961  29.701  < 2e-16 ***
zNobs:zk                    0.43939    0.01867  23.539  < 2e-16 ***
zAstar:VDRMedium            0.43734    0.04814   9.084  < 2e-16 ***
zAstar:VDRSlow              0.09810    0.04727   2.075  0.03796 *
zAstar:zSNR_X               0.16040    0.01833   8.749  < 2e-16 ***
zNobs:zAstar               -0.15368    0.01904  -8.072 6.92e-16 ***
zSNR_Y:DistributionNormal   0.27879    0.04548   6.129 8.83e-10 ***
zSNR_Y:DistributionSkew     0.06147    0.04642   1.324  0.18545
zAstar:DistributionNormal   0.18488    0.04688   3.943 8.03e-05 ***
zAstar:DistributionSkew     0.34005    0.04784   7.108 1.18e-12 ***
zk:zSNR_X                  -0.11342    0.01810  -6.267 3.67e-10 ***
zk:VDRMedium                0.17896    0.04593   3.897 9.76e-05 ***
zk:VDRSlow                  0.22189    0.04499   4.932 8.13e-07 ***
zNobs:VDRMedium            -0.10351    0.04685  -2.209  0.02714 *
zNobs:VDRSlow              -0.22685    0.04669  -4.859 1.18e-06 ***
zAstar:PIRLow               0.21493    0.04834   4.446 8.74e-06 ***
zAstar:PIRMedium            0.12492    0.04717   2.649  0.00808 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -8.37384    0.23729 -35.290
-7|-6 -5.62648    0.09761 -57.644
-6|-5 -4.12689    0.07543 -54.712
-5|-4 -3.08330    0.06514 -47.333
-4|-3 -1.91075    0.05764 -33.148
-3|-2 -1.02958    0.05427 -18.972
-2|-1 -0.17113    0.05308  -3.224
-1|0   0.79515    0.05441  14.615
0|1    2.00578    0.05715  35.100
1|2    2.29458    0.05797  39.582
2|3    2.64246    0.05915  44.677
3|4    3.05963    0.06082  50.309
4|5    3.42641    0.06257  54.759
5|6    3.80599    0.06478  58.755
6|7    4.33918    0.06884  63.037
7|8    4.64933    0.07193  64.638
8|9    5.00381    0.07650  65.412
```

```
9|10   5.36841    0.08267  64.936
10|11  5.72895    0.09061  63.227
11|12  6.07494    0.10040  60.505
12|13  6.64449    0.12236  54.301
```

```
> ci2 <- confint(fm2)
> ci2
                                 2.5 %      97.5 %
zNobs                      -0.150257572 -0.02063276
zk                          1.269795590  1.40567487
zAstar                     -2.246114434 -2.02077267
zSNR_X                      0.111163936  0.18463111
zSNR_Y                     -0.093233912  0.03399792
VDRMedium                   0.170843250  0.34985295
VDRSlow                    -0.153255110  0.02586687
PIRLow                      0.474350749  0.65437443
PIRMedium                   0.305644052  0.48443056
DistributionNormal         -0.064104746  0.11329485
DistributionSkew           -0.104700227  0.07251646
zk:zAstar                   0.544017789  0.62088737
zNobs:zk                    0.402879960  0.47605661
zAstar:VDRMedium            0.343041602  0.53177040
zAstar:VDRSlow              0.005466906  0.19076752
zAstar:zSNR_X               0.124502795  0.19637377
zNobs:zAstar               -0.191042515 -0.11641003
zSNR_Y:DistributionNormal   0.189678448  0.36798180
zSNR_Y:DistributionSkew    -0.029510215  0.15247738
zAstar:DistributionNormal   0.093006792  0.27679689
zAstar:DistributionSkew     0.246322314  0.43386033
zk:zSNR_X                  -0.148918221 -0.07797098
zk:VDRMedium                0.088996091  0.26904280
zk:VDRSlow                  0.133709473  0.31006958
zNobs:VDRMedium            -0.195354773 -0.01171178
zNobs:VDRSlow              -0.318408460 -0.13539064
zAstar:PIRLow               0.120190811  0.30968734
zAstar:PIRMedium            0.032488974  0.21738176
```

```
> 1-fm1$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.1617758
```

B.5 PLS2 RMSECV 1st minimum Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(RMSECV_first_min_LVlessAstar) ~ 1,
              link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)
> fmnull$logLik
[1] -22359.62

> fm1 <- clm(ordered(RMSECV_first_min_LVlessAstar) ~
               zNobs+zk+zAstar+zSNR_X+zSNR_Y+zm+zs+VDR_X+VDR_Y+PIR+RCF+Distribution,
               link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)

> summary(fm1)
formula:
ordered(RMSECV_first_min_LVlessAstar) ~
     zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs + VDR_X + VDR_Y + PIR + RCF + Distribution

data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  11662 -18075.94 36225.88 9(0)  3.17e-11 1.0e+04

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.14583    0.01783   8.178 2.89e-16 ***
zk                  0.52802    0.01856  28.446  < 2e-16 ***
zAstar             -1.90031    0.02678 -70.966  < 2e-16 ***
zSNR_X              0.13276    0.01768   7.507 6.05e-14 ***
zSNR_Y              0.13216    0.01776   7.440 1.01e-13 ***
zm                  0.02120    0.01749   1.212 0.225530
zs                  0.85770    0.02305  37.214  < 2e-16 ***
VDR_XMedium         1.55122    0.04629  33.510  < 2e-16 ***
VDR_XSlow           1.93677    0.04407  43.948  < 2e-16 ***
VDR_YMedium         0.08405    0.04346   1.934 0.053133 .
VDR_YSlow           0.68053    0.04105  16.577  < 2e-16 ***
PIRLow              0.31430    0.04064   7.734 1.04e-14 ***
PIRMedium          -0.15211    0.04362  -3.487 0.000488 ***
RCFLow             -0.35587    0.04106  -8.667  < 2e-16 ***
RCFMedium          -0.53375    0.04389 -12.160  < 2e-16 ***
DistributionNormal  0.03470    0.04374   0.793 0.427576
DistributionSinh    0.11355    0.04503   2.522 0.011673 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -5.68707    0.13025 -43.663
-7|-6 -4.46290    0.08818 -50.610
```

```
-6|-5 -3.38382    0.07203 -46.979
-5|-4 -2.31477    0.06440 -35.945
-4|-3 -1.49082    0.06114 -24.385
-3|-2 -0.71522    0.05945 -12.032
-2|-1  0.01690    0.05896   0.287
-1|0   0.89800    0.05993  14.983
0|1    4.34885    0.07300  59.575
1|2    5.88265    0.08484  69.335
2|3    6.93417    0.10534  65.829
3|4    7.71935    0.13469  57.313
4|5    8.29902    0.16811  49.367
5|6    8.88821    0.21607  41.135
6|7    9.58360    0.29724  32.242
7|8   10.12420    0.38455  26.327
8|9   10.68465    0.50500  21.158
9|10  10.97246    0.58168  18.863
10|12 11.37809    0.71065  16.011
12|13 12.07143    1.00251  12.041
(338 observations deleted due to missingness)

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.1915811

> ci1 <- confint(fm1)
> ci1
                          2.5 %      97.5 %
zNobs                0.110891930  0.18079901
zk                   0.491713688  0.56447852
zAstar              -1.953002971 -1.84803313
zSNR_X               0.098106490  0.16743050
zSNR_Y               0.097361616  0.16699619
zm                  -0.013080726  0.05548982
zs                   0.812658259  0.90300830
VDR_XMedium          1.460644970  1.64210941
VDR_XSlow            1.850603413  2.02335746
VDR_YMedium         -0.001120361  0.16924732
VDR_YSlow            0.600132083  0.76106428
PIRLow               0.234674185  0.39397794
PIRMedium           -0.237615309 -0.06661220
RCFLow              -0.436381606 -0.27542545
RCFMedium           -0.619825126 -0.44775519
DistributionNormal -0.051047794  0.12042454
DistributionSinh    0.025296673  0.20180181

> sum(is.na(SIM2_DATA_AtoS_DOE_04March2015$RMSECV_first_min_LVlessAstar)==FALSE)
[1] 11662
> klogn <- log(11662)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn) % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)
```

```
> summary(fm2)
formula:
ordered(RMSECV_first_min_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs
   + VDR_X + VDR_Y + PIR + RCF + Distribution + zs:VDR_X + zAstar:zs + zk:zAstar + zNobs:zAstar
   + zAstar:VDR_X + zNobs:zk + zm:zs + zSNR_X:RCF + zAstar:VDR_Y + zAstar:zSNR_X + zk:VDR_X
   + zk:zSNR_X + zNobs:Distribution + zSNR_X:PIR + zNobs:VDR_X + zAstar:PIR + zAstar:zSNR_Y
   + zk:zs + zNobs:VDR_Y + VDR_X:VDR_Y + PIR:Distribution + zs:PIR + zm:RCF + zSNR_Y:RCF
   + zSNR_X:zSNR_Y

data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  11662 -16373.31 32908.62 9(0)  1.26e-10 1.4e+04

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
zNobs                       0.361664   0.047295   7.647 2.06e-14 ***
zk                          0.857005   0.030541  28.061  < 2e-16 ***
zAstar                     -4.066915   0.072749 -55.904  < 2e-16 ***
zSNR_X                      0.298175   0.040637   7.338 2.18e-13 ***
zSNR_Y                      0.116447   0.032610   3.571 0.000356 ***
zm                          0.257148   0.033915   7.582 3.40e-14 ***
zs                          1.711548   0.060851  28.127  < 2e-16 ***
VDR_XMedium                 2.245603   0.078541  28.591  < 2e-16 ***
VDR_XSlow                   2.356931   0.072121  32.680  < 2e-16 ***
VDR_YMedium                 0.414753   0.067840   6.114 9.73e-10 ***
VDR_YSlow                   0.712188   0.063836  11.157  < 2e-16 ***
PIRLow                      0.020226   0.084416   0.240 0.810637
PIRMedium                  -0.376525   0.087589  -4.299 1.72e-05 ***
RCFLow                     -0.377094   0.042179  -8.940  < 2e-16 ***
RCFMedium                  -0.447078   0.045821  -9.757  < 2e-16 ***
DistributionNormal         -0.453534   0.074491  -6.088 1.14e-09 ***
DistributionSinh           -0.258992   0.077289  -3.351 0.000805 ***
zs:VDR_XMedium              0.955316   0.051810  18.439  < 2e-16 ***
zs:VDR_XSlow                1.315319   0.054602  24.089  < 2e-16 ***
zAstar:zs                  -1.352652   0.044414 -30.456  < 2e-16 ***
zk:zAstar                   0.240988   0.017723  13.598  < 2e-16 ***
zNobs:zAstar                0.166259   0.015146  10.977  < 2e-16 ***
zAstar:VDR_XMedium          0.640319   0.050823  12.599  < 2e-16 ***
zAstar:VDR_XSlow            0.466586   0.050618   9.218  < 2e-16 ***
zNobs:zk                   -0.168473   0.016030 -10.510  < 2e-16 ***
zm:zs                       0.214657   0.022557   9.516  < 2e-16 ***
zSNR_X:RCFLow              -0.364451   0.044184  -8.248  < 2e-16 ***
zSNR_X:RCFMedium           -0.100678   0.045069  -2.234 0.025491 *
zAstar:VDR_YMedium          0.173356   0.045062   3.847 0.000120 ***
zAstar:VDR_YSlow            0.299918   0.045190   6.637 3.21e-11 ***
zAstar:zSNR_X               0.099892   0.014947   6.683 2.34e-11 ***
zk:VDR_XMedium             -0.287396   0.044754  -6.422 1.35e-10 ***
```

```
zk:VDR_XSlow                -0.234562   0.044449  -5.277 1.31e-07 ***
zk:zSNR_X                   -0.091711   0.015886  -5.773 7.79e-09 ***
zNobs:DistributionNormal    -0.173680   0.044491  -3.904 9.47e-05 ***
zNobs:DistributionSinh      -0.278126   0.045720  -6.083 1.18e-09 ***
zSNR_X:PIRLow                0.157306   0.045106   3.487 0.000488 ***
zSNR_X:PIRMedium            -0.079248   0.044492  -1.781 0.074883 .
zNobs:VDR_XMedium            0.220039   0.044999   4.890 1.01e-06 ***
zNobs:VDR_XSlow             -0.033062   0.044320  -0.746 0.455685
zAstar:PIRLow                0.240225   0.050236   4.782 1.74e-06 ***
zAstar:PIRMedium            -0.089333   0.050338  -1.775 0.075953 .
zAstar:zSNR_Y                0.062187   0.014847   4.189 2.81e-05 ***
zk:zs                        0.077689   0.018002   4.316 1.59e-05 ***
zNobs:VDR_YMedium            0.003524   0.045674   0.077 0.938502
zNobs:VDR_YSlow             -0.219673   0.044467  -4.940 7.80e-07 ***
VDR_XMedium:VDR_YMedium     -0.647294   0.116411  -5.560 2.69e-08 ***
VDR_XSlow:VDR_YMedium        0.155024   0.106007   1.462 0.143632
VDR_XMedium:VDR_YSlow       -0.046070   0.106523  -0.432 0.665382
VDR_XSlow:VDR_YSlow          0.044897   0.098066   0.458 0.647076
PIRLow:DistributionNormal    0.621676   0.107263   5.796 6.80e-09 ***
PIRMedium:DistributionNormal 0.584286   0.113094   5.166 2.39e-07 ***
PIRLow:DistributionSinh      0.470827   0.111807   4.211 2.54e-05 ***
PIRMedium:DistributionSinh   0.468053   0.115980   4.036 5.45e-05 ***
zs:PIRLow                   -0.225559   0.051476  -4.382 1.18e-05 ***
zs:PIRMedium                 0.029934   0.051011   0.587 0.557329
zm:RCFLow                   -0.174639   0.043966  -3.972 7.12e-05 ***
zm:RCFMedium                -0.013452   0.044508  -0.302 0.762474
zSNR_Y:RCFLow                0.178787   0.045205   3.955 7.65e-05 ***
zSNR_Y:RCFMedium             0.006380   0.045666   0.140 0.888891
zSNR_X:zSNR_Y                0.047111   0.015258   3.088 0.002018 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -7.56604    0.14942 -50.635
-7|-6 -6.30433    0.11394 -55.329
-6|-5 -5.12555    0.10018 -51.164
-5|-4 -3.83326    0.09164 -41.831
-4|-3 -2.72515    0.08650 -31.503
-3|-2 -1.61867    0.08325 -19.444
-2|-1 -0.57011    0.08171  -6.977
-1|0   0.61388    0.08219   7.469
0|1    4.31834    0.09106  47.425
1|2    5.87788    0.10049  58.490
2|3    6.94126    0.11832  58.664
3|4    7.73460    0.14515  53.287
4|5    8.31586    0.17662  47.084
5|6    8.90594    0.22277  39.978
6|7    9.60395    0.30216  31.784
```

```
7|8   10.14542    0.38837  26.123
8|9   10.70582    0.50791  21.078
9|10  10.99378    0.58421  18.818
10|12 11.39968    0.71272  15.995
12|13 12.09340    1.00398  12.045
(338 observations deleted due to missingness)

> ci2 <- confint(fm2)
> ci2
                                    2.5 %       97.5 %
zNobs                          0.26900123  0.454401802
zk                             0.79727627  0.917003709
zAstar                        -4.20994886 -3.924770113
zSNR_X                         0.21852678  0.377827519
zSNR_Y                         0.05255407  0.180388114
zm                             0.19068785  0.323637727
zs                             1.59247364  1.831015458
VDR_XMedium                    2.09190507  2.399794757
VDR_XSlow                      2.21583446  2.498555489
VDR_YMedium                    0.28183384  0.547771824
VDR_YSlow                      0.58714391  0.837386143
PIRLow                        -0.14524987  0.185670153
PIRMedium                     -0.54829461 -0.204935992
RCFLow                        -0.45980696 -0.294463419
RCFMedium                     -0.53692812 -0.357309490
DistributionNormal            -0.59958773 -0.307577244
DistributionSinh              -0.41050389 -0.107524668
zs:VDR_XMedium                 0.85385862  1.056958176
zs:VDR_XSlow                   1.20842949  1.422475414
zAstar:zs                     -1.43988498 -1.265778793
zk:zAstar                      0.20630276  0.275776895
zNobs:zAstar                   0.13659459  0.195967081
zAstar:VDR_XMedium             0.54076683  0.739995740
zAstar:VDR_XSlow               0.36742628  0.565853341
zNobs:zk                      -0.19991837 -0.137080941
zm:zs                          0.17046869  0.258894140
zSNR_X:RCFLow                 -0.45109592 -0.277890427
zSNR_X:RCFMedium              -0.18902539 -0.012353746
zAstar:VDR_YMedium             0.08504728  0.261694748
zAstar:VDR_YSlow               0.21135776  0.388507267
zAstar:zSNR_X                  0.07060265  0.129196850
zk:VDR_XMedium                -0.37518204 -0.199742495
zk:VDR_XSlow                  -0.32170952 -0.147464162
zk:zSNR_X                     -0.12285719 -0.060582300
zNobs:DistributionNormal      -0.26089939 -0.086493010
zNobs:DistributionSinh        -0.36776799 -0.188540534
zSNR_X:PIRLow                  0.06892225  0.245741835
zSNR_X:PIRMedium              -0.16644524  0.007966726
zNobs:VDR_XMedium              0.13186888  0.308266824
```

```
zNobs:VDR_XSlow              -0.11991954  0.053818693
zAstar:PIRLow                 0.14177416  0.338704415
zAstar:PIRMedium             -0.18801239  0.009316110
zAstar:zSNR_Y                 0.03308888  0.091289200
zk:zs                         0.04239594  0.112965748
zNobs:VDR_YMedium            -0.08599392  0.093049401
zNobs:VDR_YSlow              -0.30684063 -0.132528540
VDR_XMedium:VDR_YMedium      -0.87553134 -0.419189452
VDR_XSlow:VDR_YMedium        -0.05274141  0.362811701
VDR_XMedium:VDR_YSlow        -0.25487484  0.162700940
VDR_XSlow:VDR_YSlow          -0.14727614  0.237147123
PIRLow:DistributionNormal     0.41150324  0.831981429
PIRMedium:DistributionNormal  0.36270285  0.806039862
PIRLow:DistributionSinh       0.25173438  0.690024513
PIRMedium:DistributionSinh    0.24080150  0.695452508
zs:PIRLow                    -0.32648241 -0.124689300
zs:PIRMedium                 -0.07002967  0.129939782
zm:RCFLow                    -0.26083285 -0.088481643
zm:RCFMedium                 -0.10069717  0.073778549
zSNR_Y:RCFLow                 0.09019120  0.267396705
zSNR_Y:RCFMedium             -0.08312616  0.095887861
zSNR_X:zSNR_Y                 0.01720955  0.077022292

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.2677286
```

B.6 PLS2 RMSECV Absolute minimum Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(RMSECV_abs_min_LVlessAstar) ~ 1,
                        link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)
> fmnull$logLik
[1] -24992.35

> fm1 <- clm(ordered(RMSECV_abs_min_LVlessAstar) ~
                        zNobs+zk+zAstar+zSNR_X+zSNR_Y+zm+zs+VDR_X+VDR_Y+PIR+RCF+Distribution,
                        link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)

> summary(fm1)
formula:
ordered(RMSECV_abs_min_LVlessAstar) ~
                   zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs  + VDR_X + VDR_Y + PIR + RCF + Distribution

data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -21450.46 42976.92 9(1)  1.67e-11 1.2e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs              -0.10535    0.01711  -6.159 7.31e-10 ***
zk                  0.77510    0.01921  40.343  < 2e-16 ***
zAstar             -1.38518    0.02361 -58.660  < 2e-16 ***
zSNR_X              0.06919    0.01700   4.070 4.70e-05 ***
zSNR_Y              0.12398    0.01701   7.288 3.15e-13 ***
zm                  0.01107    0.01716   0.645   0.5189
zs                  0.60619    0.02097  28.908  < 2e-16 ***
VDR_XMedium         1.50154    0.04448  33.757  < 2e-16 ***
VDR_XSlow           1.75133    0.04206  41.639  < 2e-16 ***
VDR_YMedium         0.01762    0.04196   0.420   0.6746
VDR_YSlow           0.46821    0.03941  11.881  < 2e-16 ***
PIRLow              0.36079    0.03911   9.225  < 2e-16 ***
PIRMedium          -0.10502    0.04209  -2.495   0.0126 *
RCFLow             -0.20760    0.03952  -5.252 1.50e-07 ***
RCFMedium          -0.38793    0.04195  -9.247  < 2e-16 ***
DistributionNormal -0.39802    0.04250  -9.364  < 2e-16 ***
DistributionSinh   -0.29242    0.04343  -6.732 1.67e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
```

```
      Estimate Std. Error z value
-8|-7 -6.23515    0.16981 -36.718
-7|-6 -4.90217    0.10065 -48.705
-6|-5 -3.73349    0.07435 -50.215
-5|-4 -2.66720    0.06385 -41.770
-4|-3 -1.86665    0.05992 -31.155
-3|-2 -1.16243    0.05814 -19.994
-2|-1 -0.50314    0.05766  -8.725
-1|0   0.44409    0.05829   7.619
0|1    3.10208    0.06436  48.198
1|2    4.03148    0.06806  59.237
2|3    4.56753    0.07181  63.609
3|4    4.95686    0.07572  65.464
4|5    5.25827    0.07966  66.011
5|6    5.59614    0.08529  65.612
6|7    6.33542    0.10364  61.131
7|8    6.71971    0.11739  57.242
8|9    7.19143    0.13934  51.610
9|10   7.57367    0.16215  46.707
10|11  8.06393    0.19959  40.403
11|12  8.26674    0.21822  37.882
12|13  8.77351    0.27497  31.907

> ci1 <- confint(fm1)
> ci1
                          2.5 %      97.5 %
zNobs               -0.13889174 -0.07183822
zk                   0.73754255  0.81285821
zAstar              -1.43157810 -1.33901208
zSNR_X               0.03587461  0.10251288
zSNR_Y               0.09064570  0.15733287
zm                  -0.02256769  0.04471638
zs                   0.56514782  0.64735043
VDR_XMedium          1.41448868  1.58885679
VDR_XSlow            1.66905185  1.83392698
VDR_YMedium         -0.06462141  0.09987353
VDR_YSlow            0.39100422  0.54549059
PIRLow               0.28416343  0.43747881
PIRMedium           -0.18752787 -0.02251637
RCFLow              -0.28508904 -0.13015065
RCFMedium           -0.47018176 -0.30572316
DistributionNormal -0.48137678 -0.31475710
DistributionSinh   -0.37758848 -0.20732562

> 1-fm1$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.141719

> sum(is.na(SIM2_DATA_AtoS_DOE_04March2015$RMSECV_abs_min_LVlessAstar)==FALSE)
[1] 12000
```

```
> klogn <- log(12000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn) % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)

> summary(fm2)
formula:
ordered(RMSECV_abs_min_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs + VDR_X
    + VDR_Y + PIR + RCF + Distribution + zAstar:VDR_X + zk:zAstar + zAstar:zs + zNobs:zk + zs:VDR_X
    + zk:Distribution + zSNR_X:VDR_X + zm:RCF + zSNR_X:zm + zAstar:zSNR_X + zAstar:PIR + zs:PIR
    + zm:zs + VDR_X:VDR_Y + zk:zSNR_X + zAstar:VDR_Y + zk:PIR + zAstar:zSNR_Y + zk:zs

data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -19824.61 39787.23 9(1)  1.43e-10 2.0e+03

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
zNobs                 -0.08682    0.01727  -5.027 4.98e-07 ***
zk                     1.18736    0.04099  28.966  < 2e-16 ***
zAstar                -3.21598    0.06595 -48.761  < 2e-16 ***
zSNR_X                 0.19122    0.02821   6.779 1.21e-11 ***
zSNR_Y                 0.13769    0.01703   8.087 6.12e-16 ***
zm                     0.19848    0.03179   6.244 4.26e-10 ***
zs                     1.32938    0.05622  23.647  < 2e-16 ***
VDR_XMedium            2.25176    0.07487  30.075  < 2e-16 ***
VDR_XSlow              2.30864    0.06882  33.546  < 2e-16 ***
VDR_YMedium            0.48160    0.06605   7.291 3.07e-13 ***
VDR_YSlow              0.69754    0.06258  11.147  < 2e-16 ***
PIRLow                 0.48679    0.03999  12.174  < 2e-16 ***
PIRMedium              0.06498    0.04313   1.507 0.131914
RCFLow                -0.21102    0.03989  -5.290 1.23e-07 ***
RCFMedium             -0.28430    0.04328  -6.569 5.05e-11 ***
DistributionNormal    -0.59207    0.04333 -13.663  < 2e-16 ***
DistributionSinh      -0.43204    0.04415  -9.786  < 2e-16 ***
zAstar:VDR_XMedium     0.89580    0.04927  18.180  < 2e-16 ***
zAstar:VDR_XSlow       0.75919    0.04968  15.281  < 2e-16 ***
zk:zAstar              0.45754    0.01800  25.419  < 2e-16 ***
zAstar:zs             -0.94384    0.03913 -24.120  < 2e-16 ***
zNobs:zk              -0.31225    0.01574 -19.842  < 2e-16 ***
zs:VDR_XMedium         0.45718    0.04712   9.702  < 2e-16 ***
zs:VDR_XSlow           0.63295    0.04957  12.768  < 2e-16 ***
zk:DistributionNormal -0.56880    0.04385 -12.973  < 2e-16 ***
zk:DistributionSinh   -0.30694    0.04435  -6.921 4.50e-12 ***
zSNR_X:VDR_XMedium    -0.10651    0.04150  -2.566 0.010280 *
zSNR_X:VDR_XSlow      -0.25098    0.04172  -6.016 1.79e-09 ***
zm:RCFLow             -0.24144    0.04185  -5.769 7.98e-09 ***
zm:RCFMedium          -0.01891    0.04273  -0.443 0.658105
zSNR_X:zm             -0.10357    0.01446  -7.163 7.90e-13 ***
```

```
zAstar:zSNR_X             0.08226    0.01395   5.896 3.73e-09 ***
zAstar:PIRLow             0.36434    0.04888   7.454 9.02e-14 ***
zAstar:PIRMedium         -0.05817    0.04941  -1.177 0.239085
zs:PIRLow                -0.33263    0.04770  -6.973 3.10e-12 ***
zs:PIRMedium              0.01342    0.04822   0.278 0.780833
zm:zs                     0.10350    0.01950   5.309 1.10e-07 ***
VDR_XMedium:VDR_YMedium -0.67020    0.10746  -6.237 4.47e-10 ***
VDR_XSlow:VDR_YMedium   -0.30749    0.09927  -3.098 0.001951 **
VDR_XMedium:VDR_YSlow   -0.37037    0.10103  -3.666 0.000246 ***
VDR_XSlow:VDR_YSlow     -0.33881    0.09293  -3.646 0.000267 ***
zk:zSNR_X                -0.07532    0.01524  -4.944 7.66e-07 ***
zAstar:VDR_YMedium        0.08228    0.04152   1.982 0.047518 *
zAstar:VDR_YSlow          0.21850    0.04209   5.191 2.09e-07 ***
zk:PIRLow                 0.20524    0.04238   4.843 1.28e-06 ***
zk:PIRMedium              0.07808    0.04489   1.739 0.081962 .
zAstar:zSNR_Y             0.05174    0.01367   3.783 0.000155 ***
zk:zs                    -0.05699    0.01727  -3.300 0.000967 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -7.50239    0.17816 -42.111
-7|-6 -6.13732    0.11353 -54.058
-6|-5 -4.86232    0.08899 -54.639
-5|-4 -3.57465    0.07718 -46.314
-4|-3 -2.52124    0.07107 -35.477
-3|-2 -1.56167    0.06786 -23.013
-2|-1 -0.67065    0.06690 -10.025
-1|0   0.52768    0.06770   7.794
0|1    3.42710    0.07271  47.132
1|2    4.44474    0.07651  58.093
2|3    5.04011    0.08055  62.568
3|4    5.47680    0.08477  64.608
4|5    5.81405    0.08890  65.397
5|6    6.18584    0.09457  65.410
6|7    6.96141    0.11197  62.174
7|8    7.34696    0.12483  58.856
8|9    7.81747    0.14568  53.662
9|10   8.20025    0.16769  48.900
10|11  8.69002    0.20417  42.563
11|12  8.89123    0.22242  39.975
12|13  9.39498    0.27831  33.758

> ci2 <- confint(fm2)
> ci2
                                2.5 %      97.5 %
zNobs                   -0.1206837901 -0.05297869
zk                       1.1071324930  1.26782785
```

```
zAstar                  -3.3455725991 -3.08703051
zSNR_X                   0.1359467450  0.24652385
zSNR_Y                   0.1043290793  0.17107275
zm                       0.1361947982  0.26079659
zs                       1.2193378190  1.43971989
VDR_XMedium              2.1051870028  2.39868917
VDR_XSlow                2.1739608119  2.44374422
VDR_YMedium              0.3521895670  0.61111570
VDR_YSlow                0.5749455035  0.82026120
PIRLow                   0.4084510252  0.56520242
PIRMedium               -0.0195476779  0.14952389
RCFLow                  -0.2892339397 -0.13285110
RCFMedium               -0.3691419289 -0.19949754
DistributionNormal      -0.6770537769 -0.50718922
DistributionSinh        -0.5186089997 -0.34554472
zAstar:VDR_XMedium       0.7993044040  0.99246366
zAstar:VDR_XSlow         0.6619070626  0.85666798
zk:zAstar                0.4223129207  0.49287350
zAstar:zs               -1.0206895901 -0.86729157
zNobs:zk                -0.3431299324 -0.28143949
zs:VDR_XMedium           0.3648648048  0.54959273
zs:VDR_XSlow             0.5358427281  0.73017181
zk:DistributionNormal   -0.6547807546 -0.48290445
zk:DistributionSinh     -0.3939005081 -0.22003486
zSNR_X:VDR_XMedium      -0.1878684647 -0.02517577
zSNR_X:VDR_XSlow        -0.3327758333 -0.16922197
zm:RCFLow               -0.3234851201 -0.15942542
zm:RCFMedium            -0.1026612865  0.06483403
zSNR_X:zm               -0.1319154054 -0.07523705
zAstar:zSNR_X            0.0549194223  0.10961313
zAstar:PIRLow            0.2685769964  0.46017442
zAstar:PIRMedium        -0.1550065689  0.03867323
zs:PIRLow               -0.4261564124 -0.23915374
zs:PIRMedium            -0.0810796319  0.10793294
zm:zs                    0.0652998297  0.14172566
VDR_XMedium:VDR_YMedium -0.8808841278 -0.45962837
VDR_XSlow:VDR_YMedium   -0.5020809512 -0.11294497
VDR_XMedium:VDR_YSlow   -0.5684223087 -0.17238583
VDR_XSlow:VDR_YSlow     -0.5209763498 -0.15667225
zk:zSNR_X               -0.1051936031 -0.04546866
zAstar:VDR_YMedium       0.0009031504  0.16367528
zAstar:VDR_YSlow         0.1360188349  0.30100911
zk:PIRLow                0.1221794534  0.28830496
zk:PIRMedium            -0.0099044777  0.16605369
zAstar:zSNR_Y            0.0249350404  0.07854033
zk:zs                   -0.0908510637 -0.02315774

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.2067729
```

B.7 PLS2 Permutation minimum Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(perms_best_LVlessAstar) ~ 1,
                link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)
> fmnull$logLik
[1] -29189.31

> fm1 <- clm(ordered(perms_best_LVlessAstar) ~
                zNobs+zk+zAstar+zSNR_X+zSNR_Y+zm+zs+VDR_X+VDR_Y+PIR+RCF+Distribution,
                link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)

> summary(fm1)
formula:
ordered(perms_best_LVlessAstar) ~
                zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs  + VDR_X + VDR_Y + PIR + RCF + Distribution

data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -26488.09 53052.17 8(0)  8.14e-09 1.7e+04

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
zNobs               0.087505   0.016091   5.438 5.38e-08 ***
zk                  0.114934   0.016374   7.019 2.23e-12 ***
zAstar             -1.316023   0.022059 -59.659  < 2e-16 ***
zSNR_X              0.004144   0.016072   0.258   0.7965
zSNR_Y              0.010773   0.016070   0.670   0.5026
zm                 -0.032071   0.016272  -1.971   0.0487 *
zs                  0.077991   0.019632   3.973 7.11e-05 ***
VDR_XMedium        -0.235396   0.040494  -5.813 6.13e-09 ***
VDR_XSlow           0.210124   0.037785   5.561 2.68e-08 ***
VDR_YMedium        -0.008991   0.040557  -0.222   0.8246
VDR_YSlow           0.076867   0.037754   2.036   0.0417 *
PIRLow              0.474041   0.038061  12.455  < 2e-16 ***
PIRMedium          -0.075819   0.039940  -1.898   0.0577 .
RCFLow             -0.014532   0.037875  -0.384   0.7012
RCFMedium           0.001592   0.040393   0.039   0.9686
DistributionNormal  0.387645   0.040184   9.647  < 2e-16 ***
DistributionSinh    0.244010   0.040801   5.980 2.23e-09 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-9|-8 -4.46812    0.09206 -48.537
-8|-7 -2.59937    0.06178 -42.073
```

```
-7|-6 -1.73728    0.05719 -30.376
-6|-5 -1.04822    0.05515 -19.006
-5|-4 -0.44060    0.05416  -8.135
-4|-3  0.15736    0.05388   2.921
-3|-2  0.68639    0.05423  12.656
-2|-1  1.17506    0.05503  21.352
-1|0   1.62389    0.05622  28.886
0|1    3.71190    0.06582  56.394
1|2    4.04605    0.06854  59.029
2|3    4.28018    0.07100  60.284
3|4    4.50638    0.07389  60.989
4|5    4.78696    0.07830  61.139
5|6    5.17000    0.08608  60.061
6|7    5.68301    0.10050  56.547
7|8    6.10845    0.11684  52.280
8|9    7.16067    0.18098  39.566
9|10   8.29108    0.30700  27.006
10|12  9.59165    0.58024  16.531
12|13 10.69079    1.00167  10.673

> ci1 <- confint(fm1)
> ci1
                          2.5 %        97.5 %
zNobs                0.055973842  0.1190500586
zk                   0.082853985  0.1470426918
zAstar              -1.359356420 -1.2728837012
zSNR_X              -0.027355907  0.0356457796
zSNR_Y              -0.020723793  0.0422695620
zm                  -0.063967950 -0.0001806509
zs                   0.039513854  0.1164722900
VDR_XMedium         -0.314761629 -0.1560226913
VDR_XSlow            0.136094273  0.2842122831
VDR_YMedium         -0.088481790  0.0705023026
VDR_YSlow            0.002871556  0.1508657850
PIRLow               0.399478480  0.5486772994
PIRMedium           -0.154095130  0.0024696722
RCFLow              -0.088767015  0.0597026608
RCFMedium           -0.077575619  0.0807655298
DistributionNormal   0.308903783  0.4664267917
DistributionSinh     0.164052804  0.3239943386

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.0925414

> sum(is.na(SIM2_DATA_AtoS_DOE_04March2015$perms_best_LVlessAstar)==FALSE)
[1] 12000
> klogn <- log(12000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn) % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)
```

```
> summary(fm2)
formula:
ordered(perms_best_LVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs
    + VDR_X + VDR_Y + PIR + RCF + Distribution + VDR_X:PIR + zk:VDR_X + zAstar:PIR + PIR:Distribution
    + zk:zs + zm:VDR_X + zs:PIR + zAstar:zs + zNobs:VDR_X + VDR_X:VDR_Y + zAstar:VDR_X
    + zAstar:Distribution + zNobs:zk + zSNR_X:RCF + zs:VDR_X + zk:zSNR_X + zNobs:zSNR_X

data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -25978.32 52102.65 8(0)  1.02e-08 1.6e+04

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
zNobs                         0.206232   0.026677   7.731 1.07e-14 ***
zk                            0.377023   0.026916  14.008  < 2e-16 ***
zAstar                       -1.965864   0.056193 -34.984  < 2e-16 ***
zSNR_X                        0.032998   0.028052   1.176 0.239475
zSNR_Y                        0.008477   0.016254   0.522 0.601991
zm                            0.108210   0.027157   3.985 6.76e-05 ***
zs                            0.522990   0.051730  10.110  < 2e-16 ***
VDR_XMedium                  -0.339144   0.085648  -3.960 7.50e-05 ***
VDR_XSlow                     0.162734   0.079401   2.050 0.040412 *
VDR_YMedium                   0.252246   0.062597   4.030 5.58e-05 ***
VDR_YSlow                     0.402316   0.059227   6.793 1.10e-11 ***
PIRLow                       -0.483170   0.090349  -5.348 8.90e-08 ***
PIRMedium                     0.021938   0.091047   0.241 0.809590
RCFLow                       -0.027426   0.038395  -0.714 0.475041
RCFMedium                    -0.001639   0.041320  -0.040 0.968364
DistributionNormal            0.270602   0.065960   4.103 4.09e-05 ***
DistributionSinh              0.125975   0.067669   1.862 0.062656 .
VDR_XMedium:PIRLow            1.242773   0.097637  12.728  < 2e-16 ***
VDR_XSlow:PIRLow              1.043601   0.091683  11.383  < 2e-16 ***
VDR_XMedium:PIRMedium        -0.134337   0.099006  -1.357 0.174829
VDR_XSlow:PIRMedium           0.214819   0.096432   2.228 0.025902 *
zk:VDR_XMedium               -0.231847   0.039445  -5.878 4.16e-09 ***
zk:VDR_XSlow                 -0.490901   0.040068 -12.252  < 2e-16 ***
zAstar:PIRLow                 0.556800   0.047339  11.762  < 2e-16 ***
zAstar:PIRMedium             -0.001121   0.045831  -0.024 0.980491
PIRLow:DistributionNormal     0.748738   0.097389   7.688 1.49e-14 ***
PIRMedium:DistributionNormal -0.272480   0.099789  -2.731 0.006322 **
PIRLow:DistributionSinh       0.370057   0.099273   3.728 0.000193 ***
PIRMedium:DistributionSinh   -0.001623   0.102647  -0.016 0.987386
zk:zs                         0.116040   0.014333   8.096 5.67e-16 ***
zm:VDR_XMedium               -0.205806   0.040009  -5.144 2.69e-07 ***
zm:VDR_XSlow                 -0.223547   0.040529  -5.516 3.47e-08 ***
zs:PIRLow                    -0.312093   0.048093  -6.489 8.62e-11 ***
zs:PIRMedium                 -0.005243   0.046460  -0.113 0.910154
```

```
zAstar:zs                    -0.242779   0.034638  -7.009 2.40e-12 ***
zNobs:VDR_XMedium            -0.171739   0.039563  -4.341 1.42e-05 ***
zNobs:VDR_XSlow              -0.237366   0.039962  -5.940 2.85e-09 ***
VDR_XMedium:VDR_YMedium      -0.432736   0.101850  -4.249 2.15e-05 ***
VDR_XSlow:VDR_YMedium        -0.420607   0.096952  -4.338 1.44e-05 ***
VDR_XMedium:VDR_YSlow        -0.407949   0.095657  -4.265 2.00e-05 ***
VDR_XSlow:VDR_YSlow          -0.607293   0.090675  -6.697 2.12e-11 ***
zAstar:VDR_XMedium            0.055454   0.045731   1.213 0.225280
zAstar:VDR_XSlow              0.280782   0.046577   6.028 1.66e-09 ***
zAstar:DistributionNormal     0.210254   0.039281   5.353 8.67e-08 ***
zAstar:DistributionSinh       0.062180   0.039288   1.583 0.113497
zNobs:zk                      0.056830   0.014117   4.026 5.68e-05 ***
zSNR_X:RCFLow                 0.034800   0.039731   0.876 0.381087
zSNR_X:RCFMedium             -0.156742   0.040464  -3.874 0.000107 ***
zs:VDR_XMedium               -0.176047   0.046311  -3.801 0.000144 ***
zs:VDR_XSlow                 -0.235757   0.049092  -4.802 1.57e-06 ***
zk:zSNR_X                    -0.048019   0.014179  -3.387 0.000708 ***
zNobs:zSNR_X                 -0.045851   0.013632  -3.364 0.000769 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-9|-8 -4.95365    0.10822 -45.774
-8|-7 -3.03694    0.08334 -36.442
-7|-6 -2.11025    0.07922 -26.639
-6|-5 -1.35548    0.07705 -17.593
-5|-4 -0.69503    0.07581  -9.169
-4|-3 -0.05343    0.07528  -0.710
-3|-2  0.50631    0.07542   6.713
-2|-1  1.01473    0.07601  13.350
-1|0   1.47385    0.07703  19.135
0|1    3.57286    0.08461  42.226
1|2    3.91228    0.08672  45.114
2|3    4.14944    0.08866  46.801
3|4    4.37802    0.09098  48.121
4|5    4.66095    0.09459  49.277
5|6    5.04664    0.10112  49.909
6|7    5.56247    0.11364  48.947
7|8    5.98980    0.12832  46.678
8|9    7.04460    0.18859  37.353
9|10   8.17616    0.31156  26.243
10|12  9.47733    0.58266  16.266
12|13 10.57668    1.00308  10.544

> ci2 <- confint(fm2)
> ci2
                                 2.5 %      97.5 %
zNobs                        0.153951929  0.25852791
```

zk	0.324317790	0.42983298
zAstar	-2.076166080	-1.85588319
zSNR_X	-0.021975657	0.08799189
zSNR_Y	-0.023379812	0.04033506
zm	0.054989445	0.16145021
zs	0.421667782	0.62445578
VDR_XMedium	-0.506949792	-0.17119865
VDR_XSlow	0.007209051	0.31846963
VDR_YMedium	0.129560122	0.37494628
VDR_YSlow	0.286244775	0.51842176
PIRLow	-0.660382988	-0.30620592
PIRMedium	-0.156543010	0.20037482
RCFLow	-0.102681348	0.04782912
RCFMedium	-0.082621462	0.07935540
DistributionNormal	0.141320447	0.39989070
DistributionSinh	-0.006658152	0.25861230
VDR_XMedium:PIRLow	1.051491016	1.43423323
VDR_XSlow:PIRLow	0.863989399	1.22338901
VDR_XMedium:PIRMedium	-0.328392231	0.05971875
VDR_XSlow:PIRMedium	0.025860515	0.40387604
zk:VDR_XMedium	-0.309182863	-0.15455362
zk:VDR_XSlow	-0.569474901	-0.41240718
zAstar:PIRLow	0.464068819	0.64964300
zAstar:PIRMedium	-0.090945669	0.08871754
PIRLow:DistributionNormal	0.557924865	0.93969414
PIRMedium:DistributionNormal	-0.468080783	-0.07690444
PIRLow:DistributionSinh	0.175523638	0.56467647
PIRMedium:DistributionSinh	-0.202809754	0.19957192
zk:zs	0.087966104	0.14415336
zm:VDR_XMedium	-0.284244615	-0.12740724
zm:VDR_XSlow	-0.303008408	-0.14413240
zs:PIRLow	-0.406414341	-0.21788362
zs:PIRMedium	-0.096313581	0.08581829
zAstar:zs	-0.310728416	-0.17494446
zNobs:VDR_XMedium	-0.249293166	-0.09420392
zNobs:VDR_XSlow	-0.315705402	-0.15905278
VDR_XMedium:VDR_YMedium	-0.632387639	-0.23312941
VDR_XSlow:VDR_YMedium	-0.610657583	-0.23060172
VDR_XMedium:VDR_YSlow	-0.595471096	-0.22048972
VDR_XSlow:VDR_YSlow	-0.785070524	-0.42962233
zAstar:VDR_XMedium	-0.034160633	0.14511103
zAstar:VDR_XSlow	0.189514825	0.37210054
zAstar:DistributionNormal	0.133276008	0.28726135
zAstar:DistributionSinh	-0.014828561	0.13918420
zNobs:zk	0.029165930	0.08450573
zSNR_X:RCFLow	-0.043072477	0.11267439
zSNR_X:RCFMedium	-0.236068956	-0.07744848
zs:VDR_XMedium	-0.266838318	-0.08529079
zs:VDR_XSlow	-0.332009825	-0.13956491

```
zk:zSNR_X                   -0.075815991 -0.02023286
zNobs:zSNR_X                -0.072572814 -0.01913559

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.1100057
```

B.8 PLS2 Information Criteria Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(Info_BIC_min_Lvless.Astar) ~ 1,
              link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)
> fmnull$logLik
[1] -29834.41

> fm1 <- clm(ordered(Info_BIC_min_LvlessAstar) ~
               zNobs+zk+zAstar+zSNR_X+zSNR_Y+zm+zs+VDR_X+VDR_Y+PIR+RCF+Distribution,
              link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)

> summary(fm1)
formula:
ordered(Info_BIC_min_LvlessAstar) ~
        zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs + VDR_X + VDR_Y + PIR + RCF + Distribution
data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -21644.18 43364.36 8(0)  8.15e-11 2.2e+03

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
zNobs               1.761244   0.023931  73.597  < 2e-16 ***
zk                 -0.299391   0.016630 -18.003  < 2e-16 ***
zAstar             -2.564086   0.030489 -84.097  < 2e-16 ***
zSNR_X              0.181833   0.016865  10.782  < 2e-16 ***
zSNR_Y             -0.023786   0.016753  -1.420 0.155664
zm                 -1.327031   0.021266 -62.403  < 2e-16 ***
zs                 -0.108107   0.020029  -5.398 6.76e-08 ***
VDR_XMedium        -0.138410   0.041402  -3.343 0.000829 ***
VDR_XSlow          -0.183034   0.037136  -4.929 8.28e-07 ***
VDR_YMedium        -0.045410   0.040667  -1.117 0.264153
VDR_YSlow           0.059315   0.037362   1.588 0.112384
PIRLow             -0.105998   0.037377  -2.836 0.004569 **
PIRMedium          -0.431424   0.040269 -10.714  < 2e-16 ***
RCFLow              0.005615   0.037656   0.149 0.881462
RCFMedium          -0.081147   0.040143  -2.021 0.043232 *
DistributionNormal  0.339359   0.040272   8.427  < 2e-16 ***
DistributionSinh    0.239582   0.040248   5.953 2.64e-09 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -5.60431    0.08238 -68.026
-7|-6 -4.55482    0.07232 -62.983
-6|-5 -3.80658    0.06661 -57.150
```

```
-5|-4 -2.65478    0.06070 -43.734
-4|-3 -1.43743    0.05695 -25.241
-3|-2 -0.49956    0.05572  -8.965
-2|-1  0.42359    0.05605   7.558
-1|0   1.77419    0.05906  30.042
0|1    2.82407    0.06297  44.851
1|2    4.77470    0.07452  64.075
2|3    6.10426    0.08660  70.485
3|4    7.12725    0.10283  69.314
4|5    7.50826    0.11254  66.719
5|6    7.90720    0.12501  63.253
6|7    8.87383    0.15905  55.793
7|8    9.15829    0.16711  54.803
8|9    9.64474    0.17997  53.592
9|10   9.99115    0.19018  52.536
10|11 10.78072    0.22414  48.098
11|12 12.21481    0.35921  34.005
12|13 13.07013    0.51755  25.254

> ci1 <- confint(fm1)
> ci1
                         2.5 %       97.5 %
zNobs                1.71450797  1.808316800
zk                  -0.33202147 -0.266830668
zAstar              -2.62408412 -2.504564783
zSNR_X               0.14879026  0.214900557
zSNR_Y              -0.05662451  0.009046405
zm                  -1.36883312 -1.285471816
zs                  -0.14737820 -0.068862763
VDR_XMedium         -0.21958814 -0.057290562
VDR_XSlow           -0.25582962 -0.110256034
VDR_YMedium         -0.12512982  0.034285496
VDR_YSlow           -0.01390933  0.132551475
PIRLow              -0.17926697 -0.032749535
PIRMedium           -0.51039576 -0.352540236
RCFLow              -0.06819317  0.079418260
RCFMedium           -0.15984252 -0.002482188
DistributionNormal   0.26045361  0.418319812
DistributionSinh     0.16071899  0.318490607

> 1-fm1$logLik/fmnull$logLik  #McFadden's pseudo Rsquared
[1] 0.2745229

> sum(is.na(SIM2_DATA_AtoS_DOE_04March2015$Info_BIC_min_LvlessAstar)==FALSE)
[1] 12000
> klogn <- log(12000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn) % This k factor is for BIC
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM2_DATA_AtoS_DOE_04March2015)
```

```
> summary(fm2)
formula:
ordered(Info_BIC_min_LvlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs
    + VDR_X + VDR_Y + PIR + RCF + Distribution + zNobs:zAstar + zAstar:zs + zAstar:zm + zk:zm
    + zNobs:VDR_X + zNobs:zSNR_X + zm:Distribution + zNobs:Distribution + zm:PIR + zNobs:PIR
    + zNobs:zm + zk:zs + zAstar:VDR_X + zSNR_X:zm + VDR_X:VDR_Y + zk:zAstar + zm:VDR_X
    + zNobs:zs + VDR_Y:RCF + zm:RCF + zNobs:RCF + zs:Distribution + zm:VDR_Y + zk:Distribution
    + zSNR_X:Distribution + zSNR_Y:RCF + zSNR_Y:VDR_X + zSNR_X:PIR + VDR_Y:PIR
    + zSNR_X:RCF + zAstar:zSNR_Y + zm:zs + zs:PIR + zAstar:PIR + zs:VDR_X + PIR:RCF
    + zSNR_X:zs + zk:PIR + zk:RCF + zAstar:RCF + zSNR_Y:PIR + zk:VDR_Y + zSNR_Y:VDR_Y

data:    SIM2_DATA_AtoS_DOE_04March2015

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -18709.71 37657.43 9(2)  4.17e-11 4.9e+03

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
zNobs                     2.307667   0.060052  38.428  < 2e-16 ***
zk                       -0.691540   0.054585 -12.669  < 2e-16 ***
zAstar                   -4.648916   0.075239 -61.789  < 2e-16 ***
zSNR_X                    0.789000   0.046930  16.812  < 2e-16 ***
zSNR_Y                   -0.037957   0.053910  -0.704 0.481380
zm                       -1.449825   0.065403 -22.168  < 2e-16 ***
zs                        2.050037   0.068569  29.898  < 2e-16 ***
VDR_XMedium               0.249418   0.074295   3.357 0.000788 ***
VDR_XSlow                -0.012637   0.065279  -0.194 0.846506
VDR_YMedium               0.642724   0.104440   6.154 7.56e-10 ***
VDR_YSlow                 0.293080   0.102001   2.873 0.004062 **
PIRLow                    0.164434   0.085107   1.932 0.053349 .
PIRMedium                -0.273106   0.095315  -2.865 0.004166 **
RCFLow                    0.347964   0.081013   4.295 1.75e-05 ***
RCFMedium                -0.406419   0.089726  -4.530 5.91e-06 ***
DistributionNormal        0.232589   0.043799   5.310 1.09e-07 ***
DistributionSinh          0.115605   0.043448   2.661 0.007796 **
zNobs:zAstar              1.132378   0.022394  50.566  < 2e-16 ***
zAstar:zs                -1.793825   0.044386 -40.414  < 2e-16 ***
zAstar:zm                -0.647027   0.019546 -33.103  < 2e-16 ***
zk:zm                     0.116709   0.016238   7.187 6.60e-13 ***
zNobs:VDR_XMedium         0.313165   0.047463   6.598 4.17e-11 ***
zNobs:VDR_XSlow           0.599575   0.045304  13.234  < 2e-16 ***
zNobs:zSNR_X              0.298119   0.016598  17.961  < 2e-16 ***
zm:DistributionNormal    -0.693619   0.047456 -14.616  < 2e-16 ***
zm:DistributionSinh      -0.378676   0.045749  -8.277  < 2e-16 ***
zNobs:DistributionNormal  0.466842   0.047513   9.826  < 2e-16 ***
zNobs:DistributionSinh    0.406058   0.046326   8.765  < 2e-16 ***
zm:PIRLow                -0.150201   0.045751  -3.283 0.001027 **
zm:PIRMedium              0.615541   0.046997  13.098  < 2e-16 ***
zNobs:PIRLow             -0.122517   0.045340  -2.702 0.006888 **
```

```
zNobs:PIRMedium           -0.513081   0.048477 -10.584  < 2e-16 ***
zNobs:zm                   0.256314   0.017869  14.344  < 2e-16 ***
zk:zs                      0.324388   0.019753  16.422  < 2e-16 ***
zAstar:VDR_XMedium        -0.106560   0.051408  -2.073 0.038187 *
zAstar:VDR_XSlow          -0.457189   0.053070  -8.615  < 2e-16 ***
zSNR_X:zm                 -0.220570   0.016283 -13.546  < 2e-16 ***
VDR_XMedium:VDR_YMedium   -0.951558   0.114029  -8.345  < 2e-16 ***
VDR_XSlow:VDR_YMedium     -0.136406   0.099218  -1.375 0.169190
VDR_XMedium:VDR_YSlow     -0.578644   0.102796  -5.629 1.81e-08 ***
VDR_XSlow:VDR_YSlow       -0.875855   0.093780  -9.340  < 2e-16 ***
zk:zAstar                 -0.187893   0.017906 -10.493  < 2e-16 ***
zm:VDR_XMedium            -0.275811   0.046270  -5.961 2.51e-09 ***
zm:VDR_XSlow              -0.287674   0.045734  -6.290 3.17e-10 ***
zNobs:zs                  -0.229374   0.023559  -9.736  < 2e-16 ***
VDR_YMedium:RCFLow        -0.352668   0.101955  -3.459 0.000542 ***
VDR_YSlow:RCFLow          -0.083040   0.095088  -0.873 0.382503
VDR_YMedium:RCFMedium     -0.205770   0.110598  -1.861 0.062810 .
VDR_YSlow:RCFMedium        0.691091   0.102405   6.749 1.49e-11 ***
zm:RCFLow                 -0.321524   0.045576  -7.055 1.73e-12 ***
zm:RCFMedium              -0.006657   0.046198  -0.144 0.885430
zNobs:RCFLow               0.110420   0.045821   2.410 0.015960 *
zNobs:RCFMedium           -0.415661   0.047456  -8.759  < 2e-16 ***
zs:DistributionNormal     -0.311053   0.048896  -6.362 2.00e-10 ***
zs:DistributionSinh       -0.311709   0.046961  -6.638 3.19e-11 ***
zm:VDR_YMedium             0.251673   0.044321   5.678 1.36e-08 ***
zm:VDR_YSlow              -0.062689   0.045071  -1.391 0.164255
zk:DistributionNormal      0.317123   0.043310   7.322 2.44e-13 ***
zk:DistributionSinh        0.232781   0.044471   5.234 1.66e-07 ***
zSNR_X:DistributionNormal -0.251084   0.045359  -5.535 3.10e-08 ***
zSNR_X:DistributionSinh   -0.250841   0.044524  -5.634 1.76e-08 ***
zSNR_Y:RCFLow              0.360687   0.045853   7.866 3.66e-15 ***
zSNR_Y:RCFMedium           0.229934   0.045633   5.039 4.68e-07 ***
zSNR_Y:VDR_XMedium        -0.170390   0.046086  -3.697 0.000218 ***
zSNR_Y:VDR_XSlow          -0.291103   0.045661  -6.375 1.83e-10 ***
zSNR_X:PIRLow             -0.345624   0.044067  -7.843 4.40e-15 ***
zSNR_X:PIRMedium          -0.216637   0.044580  -4.860 1.18e-06 ***
VDR_YMedium:PIRLow        -0.811244   0.102989  -7.877 3.35e-15 ***
VDR_YSlow:PIRLow          -0.115989   0.094210  -1.231 0.218258
VDR_YMedium:PIRMedium     -0.154194   0.109417  -1.409 0.158769
VDR_YSlow:PIRMedium        0.055179   0.103138   0.535 0.592652
zSNR_X:RCFLow             -0.296612   0.044783  -6.623 3.51e-11 ***
zSNR_X:RCFMedium          -0.186299   0.044355  -4.200 2.67e-05 ***
zAstar:zSNR_Y             -0.091322   0.015208  -6.005 1.92e-09 ***
zm:zs                      0.201645   0.029729   6.783 1.18e-11 ***
zs:PIRLow                 -0.318981   0.057523  -5.545 2.93e-08 ***
zs:PIRMedium              -0.531289   0.059305  -8.959  < 2e-16 ***
zAstar:PIRLow              0.221169   0.052623   4.203 2.64e-05 ***
zAstar:PIRMedium           0.429118   0.052763   8.133 4.19e-16 ***
zs:VDR_XMedium            -0.336898   0.056366  -5.977 2.27e-09 ***
```

```
zs:VDR_XSlow              -0.079946   0.059974  -1.333 0.182531
PIRLow:RCFLow             -0.117224   0.093866  -1.249 0.211724
PIRMedium:RCFLow          -0.565450   0.103140  -5.482 4.20e-08 ***
PIRLow:RCFMedium           0.037822   0.101129   0.374 0.708408
PIRMedium:RCFMedium        0.064216   0.113766   0.564 0.572444
zSNR_X:zs                  0.061193   0.016190   3.780 0.000157 ***
zk:PIRLow                 -0.149505   0.041999  -3.560 0.000371 ***
zk:PIRMedium               0.070242   0.043647   1.609 0.107549
zk:RCFLow                 -0.027663   0.042425  -0.652 0.514364
zk:RCFMedium               0.172617   0.043434   3.974 7.06e-05 ***
zAstar:RCFLow             -0.070027   0.043435  -1.612 0.106910
zAstar:RCFMedium           0.168977   0.044110   3.831 0.000128 ***
zSNR_Y:PIRLow              0.198590   0.044691   4.444 8.85e-06 ***
zSNR_Y:PIRMedium          -0.014378   0.045406  -0.317 0.751499
zk:VDR_YMedium             0.170703   0.044132   3.868 0.000110 ***
zk:VDR_YSlow               0.198184   0.042905   4.619 3.85e-06 ***
zSNR_Y:VDR_YMedium        -0.189104   0.044808  -4.220 2.44e-05 ***
zSNR_Y:VDR_YSlow          -0.189152   0.045804  -4.130 3.63e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -9.03257    0.14074 -64.178
-7|-6 -7.03822    0.11758 -59.859
-6|-5 -5.72381    0.10502 -54.500
-5|-4 -3.91094    0.09428 -41.483
-4|-3 -2.05567    0.08822 -23.302
-3|-2 -0.69040    0.08610  -8.018
-2|-1  0.55926    0.08607   6.498
-1|0   2.15876    0.08803  24.523
0|1    3.26730    0.09068  36.030
1|2    5.17001    0.09816  52.669
2|3    6.55899    0.10971  59.784
3|4    7.79449    0.12791  60.937
4|5    8.21296    0.13629  60.262
5|6    8.60573    0.14581  59.021
6|7    9.43048    0.17070  55.245
7|8    9.65386    0.17802  54.228
8|9   10.04596    0.19143  52.477
9|10  10.34036    0.20257  51.045
10|11 11.06073    0.23766  46.541
11|12 12.43220    0.36970  33.628
12|13 13.26910    0.52523  25.263

> ci2 <- confint(fm2)
> ci2
                               2.5 %      97.5 %
zNobs                     2.190144846  2.42555511
```

zk	-0.798703066	-0.58471759
zAstar	-4.796839298	-4.50189775
zSNR_X	0.697115298	0.88108599
zSNR_Y	-0.143618248	0.06771454
zm	-1.578136188	-1.32175136
zs	1.915958597	2.18475818
VDR_XMedium	0.103783622	0.39503022
VDR_XSlow	-0.140574782	0.11532445
VDR_YMedium	0.438148837	0.84756955
VDR_YSlow	0.093161992	0.49301523
PIRLow	-0.002407431	0.33122149
PIRMedium	-0.459919892	-0.08627242
RCFLow	0.189225136	0.50680504
RCFMedium	-0.582258752	-0.23052183
DistributionNormal	0.146759915	0.31845398
DistributionSinh	0.030453841	0.20077145
zNobs:zAstar	1.088591123	1.17637679
zAstar:zs	-1.881051343	-1.70705383
zAstar:zm	-0.685413507	-0.60879314
zk:zm	0.084886501	0.14854091
zNobs:VDR_XMedium	0.220186742	0.40624639
zNobs:VDR_XSlow	0.510837454	0.68843385
zNobs:zSNR_X	0.265616405	0.33068260
zm:DistributionNormal	-0.786712935	-0.60068155
zm:DistributionSinh	-0.468380563	-0.28903949
zNobs:DistributionNormal	0.373778538	0.56003370
zNobs:DistributionSinh	0.315310832	0.49691231
zm:PIRLow	-0.239870821	-0.06052166
zm:PIRMedium	0.523492260	0.70772410
zNobs:PIRLow	-0.211389406	-0.03365592
zNobs:PIRMedium	-0.608124735	-0.41809285
zNobs:zm	0.221331510	0.29138137
zk:zs	0.285706511	0.36314197
zAstar:VDR_XMedium	-0.207323441	-0.00579870
zAstar:VDR_XSlow	-0.561200654	-0.35315784
zSNR_X:zm	-0.252508983	-0.18867886
VDR_XMedium:VDR_YMedium	-1.175184551	-0.72818113
VDR_XSlow:VDR_YMedium	-0.330874139	0.05806528
VDR_XMedium:VDR_YSlow	-0.780182875	-0.37721431
VDR_XSlow:VDR_YSlow	-1.059759924	-0.69214019
zk:zAstar	-0.222999376	-0.15280647
zm:VDR_XMedium	-0.366556023	-0.18517497
zm:VDR_XSlow	-0.377360475	-0.19807843
zNobs:zs	-0.275602303	-0.18325031
VDR_YMedium:RCFLow	-0.552542477	-0.15287314
VDR_YSlow:RCFLow	-0.269417462	0.10333214
VDR_YMedium:RCFMedium	-0.422596501	0.01095477
VDR_YSlow:RCFMedium	0.490471887	0.89190447
zm:RCFLow	-0.410877575	-0.23221746

zm:RCFMedium	-0.097167605	0.08393210
zNobs:RCFLow	0.020625424	0.20024583
zNobs:RCFMedium	-0.508727192	-0.32269549
zs:DistributionNormal	-0.407050873	-0.21537392
zs:DistributionSinh	-0.403807255	-0.21971063
zm:VDR_YMedium	0.164844978	0.33858882
zm:VDR_YSlow	-0.151046014	0.02563606
zk:DistributionNormal	0.232253333	0.40203152
zk:DistributionSinh	0.145650893	0.31998307
zSNR_X:DistributionNormal	-0.340010074	-0.16219860
zSNR_X:DistributionSinh	-0.338130471	-0.16359489
zSNR_Y:RCFLow	0.270838479	0.45058423
zSNR_Y:RCFMedium	0.140523386	0.31940620
zSNR_Y:VDR_XMedium	-0.260744894	-0.08008550
zSNR_Y:VDR_XSlow	-0.380614145	-0.20161926
zSNR_X:PIRLow	-0.432026946	-0.25928088
zSNR_X:PIRMedium	-0.304032882	-0.12927790
VDR_YMedium:PIRLow	-1.013178767	-0.60945397
VDR_YSlow:PIRLow	-0.300634969	0.06867262
VDR_YMedium:PIRMedium	-0.368704364	0.06021985
VDR_YSlow:PIRMedium	-0.146952839	0.25735525
zSNR_X:RCFLow	-0.384401577	-0.20884898
zSNR_X:RCFMedium	-0.273269218	-0.09939344
zAstar:zSNR_Y	-0.121141767	-0.06152431
zm:zs	0.143293081	0.25982832
zs:PIRLow	-0.431752877	-0.20625357
zs:PIRMedium	-0.647637202	-0.41515379
zAstar:PIRLow	0.118029679	0.32431755
zAstar:PIRMedium	0.325730832	0.53256699
zs:VDR_XMedium	-0.447482782	-0.22652089
zs:VDR_XSlow	-0.197633306	0.03747493
PIRLow:RCFLow	-0.301210759	0.06674862
PIRMedium:RCFLow	-0.767696095	-0.36338012
PIRLow:RCFMedium	-0.160388988	0.23604324
PIRMedium:RCFMedium	-0.158853866	0.28711618
zSNR_X:zs	0.029462384	0.09292953
zk:PIRLow	-0.231824539	-0.06718592
zk:PIRMedium	-0.015297499	0.15580375
zk:RCFLow	-0.110812901	0.05549672
zk:RCFMedium	0.087513337	0.25778058
zAstar:RCFLow	-0.155164143	0.01510404
zAstar:RCFMedium	0.082537766	0.25545356
zSNR_Y:PIRLow	0.111002353	0.28619526
zSNR_Y:PIRMedium	-0.103380446	0.07461210
zk:VDR_YMedium	0.084218077	0.25722096
zk:VDR_YSlow	0.114110422	0.28230070
zSNR_Y:VDR_YMedium	-0.276945412	-0.10129716
zSNR_Y:VDR_YSlow	-0.278951781	-0.09939772

```
> 1-fm2$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.3728815
```

Appendix C

Coefficient Identification Logistic Models

C.1 PLS1 Coefficient Correlation Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(COEF_COR_BestLVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC_COEFS)
> fmnull$logLik
[1] -18094.54

> fm1 <- clm(ordered(COEF_COR_BestLVlessAstar) ~ zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,link="logit",d
> summary(fm1)
formula:
ordered(COEF_COR_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC_COEFS

 link  threshold nobs logLik   AIC      niter max.grad cond.H
 logit flexible  9000 -9818.12 19686.24 11(0) 1.73e-09 4.5e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.41823    0.02195  19.057  < 2e-16 ***
zk                 -0.15895    0.02085  -7.623 2.48e-14 ***
zAstar             -4.77891    0.05689 -84.005  < 2e-16 ***
zSNR_X              0.41923    0.02172  19.305  < 2e-16 ***
zSNR_Y              0.34229    0.02170  15.770  < 2e-16 ***
VDRMedium          -1.23773    0.05367 -23.064  < 2e-16 ***
VDRSlow            -2.41833    0.05801 -41.686  < 2e-16 ***
PIRLow              1.73308    0.05504  31.488  < 2e-16 ***
PIRMedium           1.12373    0.05155  21.797  < 2e-16 ***
DistributionNormal -0.11806    0.05111  -2.310   0.0209 *
```

```
DistributionSkew   -0.33569    0.05112  -6.566 5.16e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
       Estimate Std. Error z value
-8|-7 -10.98079    0.15620  -70.30
-7|-6  -8.62676    0.11460  -75.28
-6|-5  -6.94488    0.10042  -69.16
-5|-4  -5.27624    0.08827  -59.78
-4|-3  -3.34424    0.07544  -44.33
-3|-2  -1.18447    0.06745  -17.56
-2|-1   0.93004    0.06551   14.20
-1|0    4.05929    0.07851   51.70
0|1    11.35130    0.26390   43.01
1|2    11.78743    0.31895   36.96
2|3    12.57675    0.45915   27.39
3|5    13.08759    0.58665   22.31
5|6    13.49305    0.71472   18.88
6|8    14.18620    1.00540   14.11

> ci1 <- confint(fm1)
> ci1
                         2.5 %     97.5 %
zNobs                0.3752741  0.4613047
zk                  -0.1998091 -0.1180695
zAstar              -4.8912761 -4.6682688
zSNR_X               0.3767303  0.4618575
zSNR_Y               0.2997912  0.3848766
VDRMedium           -1.3431217 -1.1327496
VDRSlow             -2.5324282 -2.3050121
PIRLow               1.6254659  1.8412267
PIRMedium            1.0228536  1.2249469
DistributionNormal -0.2182381 -0.0178948
DistributionSkew   -0.4359508 -0.2355414

> 1-fm1$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.4573988

> sum(is.na(SIM_DATA_ABC_COEFS$COEF_COR_BestLVlessAstar)==FALSE)
[1] 9000
> klogn<-log(9000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC_COEFS)

> summary(fm2)
formula:
ordered(COEF_COR_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution + VDR:Distrib
data:    SIM_DATA_ABC_COEFS
```

```
 link  threshold nobs logLik   AIC      niter max.grad cond.H
 logit flexible  9000 -8912.05 17944.09 11(0) 2.18e-08 8.5e+03

Coefficients:
                             Estimate Std. Error z value Pr(>|z|)
zNobs                        0.470408   0.023554  19.971  < 2e-16 ***
zk                          -0.260437   0.058458  -4.455 8.38e-06 ***
zAstar                      -5.266719   0.089595 -58.783  < 2e-16 ***
zSNR_X                       0.205771   0.051224   4.017 5.89e-05 ***
zSNR_Y                       0.287995   0.040294   7.147 8.84e-13 ***
VDRMedium                   -1.617600   0.123954 -13.050  < 2e-16 ***
VDRSlow                     -2.018637   0.124092 -16.267  < 2e-16 ***
PIRLow                       3.040020   0.135402  22.452  < 2e-16 ***
PIRMedium                    1.344616   0.120684  11.142  < 2e-16 ***
DistributionNormal           0.190986   0.123209   1.550 0.121119
DistributionSkew            -1.190774   0.123310  -9.657  < 2e-16 ***
VDRMedium:DistributionNormal 0.406071   0.141903   2.862 0.004215 **
VDRSlow:DistributionNormal  -1.175365   0.138262  -8.501  < 2e-16 ***
VDRMedium:DistributionSkew   1.047474   0.132588   7.900 2.78e-15 ***
VDRSlow:DistributionSkew     2.043758   0.135572  15.075  < 2e-16 ***
zAstar:DistributionNormal   -0.932666   0.063258 -14.744  < 2e-16 ***
zAstar:DistributionSkew      0.375452   0.057094   6.576 4.83e-11 ***
VDRMedium:PIRLow            -0.989878   0.137434  -7.203 5.91e-13 ***
VDRSlow:PIRLow              -2.301467   0.137435 -16.746  < 2e-16 ***
VDRMedium:PIRMedium          0.280255   0.136190   2.058 0.039608 *
VDRSlow:PIRMedium           -0.647137   0.133821  -4.836 1.33e-06 ***
zk:VDRMedium                 0.476014   0.057406   8.292  < 2e-16 ***
zk:VDRSlow                   0.599232   0.055954  10.709  < 2e-16 ***
zAstar:PIRLow                0.586187   0.060837   9.635  < 2e-16 ***
zAstar:PIRMedium             0.172392   0.058470   2.948 0.003194 **
zAstar:VDRMedium            -0.033113   0.064149  -0.516 0.605719
zAstar:VDRSlow              -0.448183   0.063067  -7.107 1.19e-12 ***
zk:DistributionNormal        0.083238   0.056825   1.465 0.142968
zk:DistributionSkew         -0.301005   0.051618  -5.831 5.50e-09 ***
zk:zSNR_Y                    0.149801   0.022737   6.588 4.45e-11 ***
zSNR_X:PIRLow                0.273750   0.055826   4.904 9.41e-07 ***
zSNR_X:PIRMedium             0.409658   0.053637   7.638 2.21e-14 ***
zk:PIRLow                   -0.412504   0.054808  -7.526 5.22e-14 ***
zk:PIRMedium                -0.295309   0.054803  -5.389 7.10e-08 ***
zAstar:zSNR_Y                0.141672   0.023005   6.158 7.36e-10 ***
zAstar:zSNR_X                0.126851   0.022646   5.602 2.12e-08 ***
zNobs:zAstar                 0.143655   0.022510   6.382 1.75e-10 ***
PIRLow:DistributionNormal    0.496627   0.134786   3.685 0.000229 ***
PIRMedium:DistributionNormal 0.391971   0.134953   2.904 0.003678 **
PIRLow:DistributionSkew     -0.472256   0.131652  -3.587 0.000334 ***
PIRMedium:DistributionSkew  -0.341681   0.128942  -2.650 0.008052 **
zSNR_X:VDRMedium             0.100619   0.056983   1.766 0.077432 .
zSNR_X:VDRSlow              -0.197180   0.056203  -3.508 0.000451 ***
```

```
zSNR_Y:DistributionNormal     0.233854   0.057163   4.091 4.29e-05 ***
zSNR_Y:DistributionSkew      -0.007391   0.056516  -0.131 0.895946
zk:zAstar                     0.081534   0.022815   3.574 0.000352 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -12.6372     0.2013 -62.776
-7|-6  -9.7180     0.1565 -62.111
-6|-5  -7.6548     0.1406 -54.436
-5|-4  -5.6233     0.1284 -43.786
-4|-3  -3.4067     0.1170 -29.127
-3|-2  -1.0547     0.1107  -9.529
-2|-1   1.2989     0.1081  12.018
-1|0    4.8565     0.1218  39.884
0|1    13.0809     0.3102  42.164
1|2    13.5178     0.3582  37.734
2|3    14.3080     0.4873  29.363
3|5    14.8189     0.6089  24.337
5|6    15.2243     0.7331  20.767
6|8    15.9175     1.0185  15.628

> ci2 <- confint(fm2)
> ci2
                                    2.5 %      97.5 %
zNobs                          0.42432918  0.51666508
zk                            -0.37490134 -0.14573008
zAstar                        -5.44344055 -5.09221272
zSNR_X                         0.10537970  0.30619001
zSNR_Y                         0.20906678  0.36702848
VDRMedium                     -1.86091109 -1.37497994
VDRSlow                       -2.26233822 -1.77586799
PIRLow                         2.77518255  3.30599501
PIRMedium                      1.10831280  1.58142845
DistributionNormal            -0.05037858  0.43263974
DistributionSkew              -1.43272740 -0.94931879
VDRMedium:DistributionNormal   0.12788079  0.68416913
VDRSlow:DistributionNormal    -1.44679647 -0.90477927
VDRMedium:DistributionSkew     0.78769596  1.30745963
VDRSlow:DistributionSkew       1.77833516  2.30979703
zAstar:DistributionNormal     -1.05714416 -0.80915134
zAstar:DistributionSkew        0.26365240  0.48747011
VDRMedium:PIRLow              -1.25948201 -0.72071939
VDRSlow:PIRLow                -2.57128298 -2.03251843
VDRMedium:PIRMedium            0.01328222  0.54716704
VDRSlow:PIRMedium             -0.90968200 -0.38508534
zk:VDRMedium                   0.36354257  0.58858329
zk:VDRSlow                     0.48955613  0.70890329
```

```
zAstar:PIRLow                 0.46701592  0.70550658
zAstar:PIRMedium              0.05774775  0.28696248
zAstar:VDRMedium             -0.15868098  0.09279937
zAstar:VDRSlow               -0.57171992 -0.32448208
zk:DistributionNormal        -0.02800861  0.19475618
zk:DistributionSkew          -0.40229842 -0.19994660
zk:zSNR_Y                     0.10531808  0.19445369
zSNR_X:PIRLow                 0.16438559  0.38323019
zSNR_X:PIRMedium              0.30458994  0.51485453
zk:PIRLow                    -0.51996549 -0.30510587
zk:PIRMedium                 -0.40276990 -0.18793220
zAstar:zSNR_Y                 0.09660398  0.18679004
zAstar:zSNR_X                 0.08249818  0.17127387
zNobs:zAstar                  0.09954748  0.18779418
PIRLow:DistributionNormal     0.23249606  0.76087910
PIRMedium:DistributionNormal  0.12772172  0.65676146
PIRLow:DistributionSkew      -0.73041698 -0.21432050
PIRMedium:DistributionSkew   -0.59450447 -0.08903336
zSNR_X:VDRMedium             -0.01103230  0.21234784
zSNR_X:VDRSlow               -0.30736087 -0.08703503
zSNR_Y:DistributionNormal     0.12187116  0.34595887
zSNR_Y:DistributionSkew      -0.11814790  0.10340319
zk:zAstar                     0.03671873  0.12615768

> 1-fm2$logLik/fmnull$logLik #McFadden's pseudo Rsquared
[1] 0.5074733
```

C.2 PLS1 Coefficient Coverage Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(COEF_inCI_BestLVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC_COEFS)
> fmnull$logLik
[1] -19049.4

> fm1 <- clm(ordered(COEF_inCI_BestLVlessAstar) ~
         zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,link="logit",data=SIM_DATA_ABC_COEFS)

> summary(fm1)
formula:
ordered(COEF_inCI_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC_COEFS

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -13040.99 26135.98 9(0)  6.37e-13 4.7e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs              -0.007802   0.019519  -0.400 0.689375
zk                  0.363553   0.020742  17.527  < 2e-16 ***
zAstar             -3.106558   0.037457 -82.937  < 2e-16 ***
zSNR_X              0.091376   0.019433   4.702 2.57e-06 ***
zSNR_Y              0.167262   0.019641   8.516  < 2e-16 ***
VDRMedium          -1.787941   0.051426 -34.767  < 2e-16 ***
VDRSlow            -2.229677   0.052873 -42.170  < 2e-16 ***
PIRLow              0.952256   0.048338  19.700  < 2e-16 ***
PIRMedium           0.770102   0.047221  16.308  < 2e-16 ***
DistributionNormal -0.264416   0.047051  -5.620 1.91e-08 ***
DistributionSkew    0.183906   0.048248   3.812 0.000138 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -6.73852    0.09233  -72.98
-7|-6 -5.78079    0.08426  -68.60
-6|-5 -4.67622    0.07599  -61.54
-5|-4 -3.41728    0.06832  -50.02
-4|-3 -1.89687    0.06039  -31.41
-3|-2 -0.41841    0.05662   -7.39
-2|-1  1.08594    0.05854   18.55
-1|0   4.11998    0.07196   57.26
0|1    6.95321    0.12509   55.59
1|2    7.45885    0.15113   49.35
2|3    8.00061    0.18949   42.22
3|4    8.57609    0.24538   34.95
```

```
4|5    9.26932    0.34025   27.24
5|6   10.36794    0.58137   17.83
6|7   10.77341    0.71039   15.16
7|8   11.46656    1.00233   11.44

> ci1
                          2.5 %      97.5 %
zNobs              -0.04606461  0.03045169
zk                  0.32296957  0.40428128
zAstar             -3.18039890 -3.03356691
zSNR_X              0.05329919  0.12947559
zSNR_Y              0.12878830  0.20578081
VDRMedium          -1.88894401 -1.68735224
VDRSlow            -2.33357381 -2.12630837
PIRLow              0.85762454  1.04711059
PIRMedium           0.67763206  0.86274227
DistributionNormal -0.35667180 -0.17222915
DistributionSkew    0.08935867  0.27849296


> 1-fm1$logLik/fmnull$logLik # McFadden's pseudo Rsquared
[1] 0.3154122

> sum(is.na(SIM_DATA_ABC_COEFS$COEF_inCI_BestLVlessAstar)==FALSE)
[1] 9000
> klogn<-log(9000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC_COEFS)
> summary(fm2)
formula:
ordered(COEF_inCI_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
   + zAstar:VDR + zk:VDR + VDR:PIR + zNobs:zk + VDR:Distribution + zAstar:PIR + zSNR_Y:VDR + zk:zAstar
   + zk:PIR + zSNR_X:PIR + zk:Distribution + zAstar:zSNR_X + zSNR_X:VDR + zSNR_Y:Distribution
   + zk:zSNR_Y + PIR:Distribution + zAstar:zSNR_Y

data:    SIM_DATA_ABC_COEFS

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -11704.41 23532.82 9(0)  1.38e-12 7.1e+03

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
zNobs                       0.04673    0.02047   2.283 0.022446 *
zk                          0.01086    0.05513   0.197 0.843766
zAstar                     -3.24738    0.05787 -56.114  < 2e-16 ***
zSNR_X                      0.11873    0.04464   2.660 0.007824 **
zSNR_Y                      0.36170    0.04795   7.543 4.59e-14 ***
VDRMedium                  -1.91767    0.11554 -16.598  < 2e-16 ***
VDRSlow                    -1.81368    0.11358 -15.969  < 2e-16 ***
```

```
PIRLow                          2.93679    0.12598  23.312  < 2e-16 ***
PIRMedium                       1.77840    0.10985  16.189  < 2e-16 ***
DistributionNormal              0.08404    0.10706   0.785 0.432458
DistributionSkew               -0.26614    0.11470  -2.320 0.020328 *
zAstar:VDRMedium               -1.05341    0.05509 -19.121  < 2e-16 ***
zAstar:VDRSlow                 -1.40050    0.05533 -25.312  < 2e-16 ***
zk:VDRMedium                    1.14427    0.05590  20.471  < 2e-16 ***
zk:VDRSlow                      1.26156    0.05260  23.983  < 2e-16 ***
VDRMedium:PIRLow               -1.55167    0.12755 -12.165  < 2e-16 ***
VDRSlow:PIRLow                 -2.57941    0.12702 -20.307  < 2e-16 ***
VDRMedium:PIRMedium            -0.34754    0.12157  -2.859 0.004252 **
VDRSlow:PIRMedium              -1.35825    0.12048 -11.273  < 2e-16 ***
zNobs:zk                        0.41386    0.02091  19.791  < 2e-16 ***
VDRMedium:DistributionNormal    0.29574    0.12574   2.352 0.018674 *
VDRSlow:DistributionNormal     -0.22742    0.12136  -1.874 0.060931 .
VDRMedium:DistributionSkew      0.87107    0.12487   6.976 3.04e-12 ***
VDRSlow:DistributionSkew        1.66402    0.12699  13.103  < 2e-16 ***
zAstar:PIRLow                   0.61661    0.05267  11.707  < 2e-16 ***
zAstar:PIRMedium                0.53535    0.05121  10.453  < 2e-16 ***
zSNR_Y:VDRMedium               -0.36190    0.05117  -7.072 1.53e-12 ***
zSNR_Y:VDRSlow                 -0.48015    0.05171  -9.285  < 2e-16 ***
zk:zAstar                      -0.19449    0.01995  -9.751  < 2e-16 ***
zk:PIRLow                      -0.53518    0.05295 -10.107  < 2e-16 ***
zk:PIRMedium                   -0.37202    0.05190  -7.168 7.61e-13 ***
zSNR_X:PIRLow                   0.37038    0.05135   7.213 5.47e-13 ***
zSNR_X:PIRMedium                0.20128    0.04808   4.187 2.83e-05 ***
zk:DistributionNormal          -0.02191    0.05207  -0.421 0.673889
zk:DistributionSkew            -0.37077    0.05217  -7.107 1.18e-12 ***
zAstar:zSNR_X                   0.14651    0.01947   7.526 5.25e-14 ***
zSNR_X:VDRMedium               -0.27530    0.05055  -5.446 5.14e-08 ***
zSNR_X:VDRSlow                 -0.32735    0.05033  -6.505 7.79e-11 ***
zSNR_Y:DistributionNormal       0.13620    0.05011   2.718 0.006565 **
zSNR_Y:DistributionSkew         0.30231    0.05187   5.829 5.59e-09 ***
zk:zSNR_Y                      -0.07738    0.02146  -3.606 0.000311 ***
PIRLow:DistributionNormal      -0.34877    0.12452  -2.801 0.005096 **
PIRMedium:DistributionNormal   -0.53030    0.12020  -4.412 1.03e-05 ***
PIRLow:DistributionSkew        -0.62468    0.12665  -4.932 8.13e-07 ***
PIRMedium:DistributionSkew     -0.57039    0.12118  -4.707 2.52e-06 ***
zAstar:zSNR_Y                   0.06060    0.01971   3.074 0.002112 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -7.74752    0.13253 -58.459
-7|-6 -6.53215    0.12312 -53.054
-6|-5 -5.10589    0.11378 -44.876
-5|-4 -3.43047    0.10557 -32.494
-4|-3 -1.52150    0.09921 -15.337
```

```
-3|-2  0.19831    0.09741   2.036
-2|-1  1.97034    0.09984  19.735
-1|0   5.30554    0.11259  47.122
0|1    8.30948    0.15855  52.409
1|2    8.81581    0.17981  49.029
2|3    9.35776    0.21306  43.920
3|4    9.93335    0.26401  37.625
4|5   10.62664    0.35391  30.026
5|6   11.72532    0.58947  19.891
6|7   12.13081    0.71704  16.918
7|8   12.82398    1.00705  12.734

> ci2 <- confint(fm2)
> ci2
                                    2.5 %      97.5 %
zNobs                          0.006605809  0.08685005
zk                            -0.097287660  0.11881998
zAstar                        -3.361189224 -3.13432576
zSNR_X                         0.031240405  0.20624711
zSNR_Y                         0.267802655  0.45577422
VDRMedium                     -2.144387900 -1.69146331
VDRSlow                       -2.036575978 -1.59133809
PIRLow                         2.690380066  3.18421407
PIRMedium                      1.563302677  1.99393844
DistributionNormal            -0.125850023  0.29384037
DistributionSkew              -0.491034879 -0.04137238
zAstar:VDRMedium              -1.161562365 -0.94559642
zAstar:VDRSlow                -1.509186278 -1.29228706
zk:VDRMedium                   1.034992923  1.25411536
zk:VDRSlow                     1.158645353  1.36485473
VDRMedium:PIRLow              -1.801960708 -1.30194935
VDRSlow:PIRLow                -2.828735587 -2.33081391
VDRMedium:PIRMedium           -0.585829728 -0.10927413
VDRSlow:PIRMedium             -1.594582027 -1.12227157
zNobs:zk                       0.372915146  0.45489144
VDRMedium:DistributionNormal   0.049316748  0.54222838
VDRSlow:DistributionNormal    -0.465335003  0.01039877
VDRMedium:DistributionSkew     0.626441592  1.11593818
VDRSlow:DistributionSkew       1.415327252  1.91315308
zAstar:PIRLow                  0.513461679  0.71994301
zAstar:PIRMedium               0.435027523  0.63579433
zSNR_Y:VDRMedium              -0.462253743 -0.26165492
zSNR_Y:VDRSlow                -0.581577150 -0.37885539
zk:zAstar                     -0.233636088 -0.15543942
zk:PIRLow                     -0.639027696 -0.43144055
zk:PIRMedium                  -0.473765882 -0.27030760
zSNR_X:PIRLow                  0.269797772  0.47108405
zSNR_X:PIRMedium               0.107075772  0.29553751
zk:DistributionNormal         -0.123921526  0.08020500
```

```
zk:DistributionSkew          -0.473085807 -0.26857722
zAstar:zSNR_X                 0.108375397  0.18469516
zSNR_X:VDRMedium             -0.374410803 -0.17626075
zSNR_X:VDRSlow               -0.426024396 -0.22874122
zSNR_Y:DistributionNormal     0.038002371  0.23442751
zSNR_Y:DistributionSkew       0.200695388  0.40401760
zk:zSNR_Y                    -0.119439108 -0.03530729
PIRLow:DistributionNormal    -0.592924255 -0.10478987
PIRMedium:DistributionNormal -0.766000563 -0.29478669
PIRLow:DistributionSkew      -0.873060828 -0.37656913
PIRMedium:DistributionSkew   -0.808011479 -0.33296167
zAstar:zSNR_Y                 0.021971877  0.09924493

> 1-fm2$logLik/fmnull$logLik # McFadden's pseudo Rsquared

[1] 0.3855761
```

C.3 PLS2 Coefficient Correlation Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(COEF_COR_BestLVlessAstar) ~ 1,link="logit",data=SIM2_COEF_DATA_AtoT)
> fmnull$logLik
[1] -17947.28

> fm1 <- clm(ordered(COEF_COR_BestLVlessAstar) ~ zNobs+zk+zAstar+zSNR_X+zSNR_Y+zm+zs+VDR_X+VDR_Y+PIR+RCF+Distribut
> summary(fm1)
formula:
ordered(COEF_COR_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs + VDR_X + VDR_Y + PIR + RCF +
data:    SIM2_COEF_DATA_AtoT

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -12420.99 24901.98 11(0) 7.11e-09 4.8e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.47609    0.02089  22.790  < 2e-16 ***
zk                  0.29705    0.02056  14.448  < 2e-16 ***
zAstar             -3.23798    0.04593 -70.501  < 2e-16 ***
zSNR_X              0.31408    0.02049  15.330  < 2e-16 ***
zSNR_Y              0.05848    0.02030   2.881  0.00396 **
zm                  0.17072    0.02135   7.995 1.30e-15 ***
zs                  1.01833    0.02300  44.273  < 2e-16 ***
VDR_XMedium         2.43750    0.05649  43.152  < 2e-16 ***
VDR_XSlow           1.34291    0.04663  28.796  < 2e-16 ***
VDR_YMedium        -0.11504    0.04899  -2.348  0.01885 *
VDR_YSlow           0.36209    0.04578   7.910 2.58e-15 ***
PIRLow              0.95341    0.04679  20.377  < 2e-16 ***
PIRMedium           0.32215    0.04883   6.597 4.19e-11 ***
RCFLow             -0.09134    0.04625  -1.975  0.04828 *
RCFMedium          -0.32935    0.04910  -6.708 1.98e-11 ***
DistributionNormal  0.10798    0.04867   2.219  0.02651 *
DistributionSinh    0.38004    0.05034   7.550 4.37e-14 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
       Estimate Std. Error z value
-8|-7 -11.05451    1.00310 -11.020
-7|-6  -7.65014    0.20143 -37.980
-6|-5  -5.56970    0.10397 -53.569
-5|-4  -3.67150    0.07898 -46.484
-4|-3  -2.49862    0.07149 -34.949
-3|-2  -1.34381    0.06711 -20.025
-2|-1  -0.23088    0.06563  -3.518
-1|0    1.54523    0.06681  23.129
```

```
0|1    11.29928    0.19256  58.679
1|2    12.06658    0.24017  50.241
2|3    13.12747    0.36102  36.363
3|4    13.93841    0.51886  26.863
4|5    14.63156    0.72057  20.306


> ci1 <- confint(fm1)
> ci1
                          2.5 %        97.5 %
zNobs                0.43521343  0.5171040063
zk                   0.25682869  0.3374271014
zAstar              -3.32865159 -3.1486162098
zSNR_X               0.27396905  0.3542849319
zSNR_Y               0.01870702  0.0982689072
zm                   0.12889789  0.2126060931
zs                   0.97341668  1.0635849143
VDR_XMedium          2.32716218  2.5485920501
VDR_XSlow            1.25168590  1.4344962408
VDR_YMedium         -0.21104443 -0.0190155378
VDR_YSlow            0.27240429  0.4518576334
PIRLow               0.86183494  1.0452526597
PIRMedium            0.22651686  0.4179404156
RCFLow              -0.18200532 -0.0007003298
RCFMedium           -0.42560987 -0.2331301175
DistributionNormal   0.01255865  0.2033473571
DistributionSinh     0.28139274  0.4787251420

> 1-fm1$logLik/fmnull$logLik
[1] 0.3079182

> sum(is.na(SIM2_COEF_DATA_AtoT$COEF_COR_BestLVlessAstar)==FALSE)
[1] 12000
> klogn<-log(12000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)

> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM2_COEF_DATA_AtoT)
> summary(fm2)
formula:
ordered(COEF_COR_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs + VDR_X + VDR_Y + PIR + RCF +
data:    SIM2_COEF_DATA_AtoT

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -10774.97 21693.93 11(0) 6.78e-09 7.4e+03

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
zNobs                         0.891572   0.048697  18.309  < 2e-16 ***
zk                            0.028908   0.055312   0.523 0.601228
```

zAstar	-5.043175	0.097742	-51.597	< 2e-16	***
zSNR_X	0.539447	0.048260	11.178	< 2e-16	***
zSNR_Y	0.118520	0.021459	5.523	3.33e-08	***
zm	0.546637	0.051573	10.599	< 2e-16	***
zs	2.031082	0.074738	27.176	< 2e-16	***
VDR_XMedium	2.314404	0.125712	18.410	< 2e-16	***
VDR_XSlow	2.555504	0.115089	22.204	< 2e-16	***
VDR_YMedium	-0.022448	0.051589	-0.435	0.663461	
VDR_YSlow	0.390171	0.048412	8.059	7.67e-16	***
PIRLow	1.845726	0.078236	23.592	< 2e-16	***
PIRMedium	0.661412	0.085312	7.753	8.99e-15	***
RCFLow	-0.085987	0.048558	-1.771	0.076594	.
RCFMedium	-0.136130	0.052166	-2.610	0.009066	**
DistributionNormal	0.111433	0.085479	1.304	0.192360	
DistributionSinh	0.274333	0.089502	3.065	0.002176	**
zs:VDR_XMedium	1.034007	0.052567	19.670	< 2e-16	***
zs:VDR_XSlow	1.824027	0.057115	31.936	< 2e-16	***
zAstar:zs	-1.526507	0.061360	-24.878	< 2e-16	***
VDR_XMedium:DistributionNormal	1.483913	0.135329	10.965	< 2e-16	***
VDR_XSlow:DistributionNormal	-1.126078	0.120646	-9.334	< 2e-16	***
VDR_XMedium:DistributionSinh	1.259404	0.136641	9.217	< 2e-16	***
VDR_XSlow:DistributionSinh	-0.396003	0.126316	-3.135	0.001719	**
VDR_XMedium:PIRLow	-1.512837	0.129466	-11.685	< 2e-16	***
VDR_XSlow:PIRLow	-1.555295	0.111223	-13.984	< 2e-16	***
VDR_XMedium:PIRMedium	-0.337382	0.134311	-2.512	0.012006	*
VDR_XSlow:PIRMedium	-0.084255	0.120895	-0.697	0.485849	
zk:zs	0.220375	0.018497	11.914	< 2e-16	***
zk:DistributionNormal	0.570876	0.052000	10.978	< 2e-16	***
zk:DistributionSinh	0.690467	0.053809	12.832	< 2e-16	***
zAstar:VDR_XMedium	0.969348	0.079942	12.126	< 2e-16	***
zAstar:VDR_XSlow	0.353000	0.073801	4.783	1.73e-06	***
zk:VDR_XMedium	-0.356265	0.053952	-6.603	4.02e-11	***
zk:VDR_XSlow	0.019040	0.053841	0.354	0.723615	
zAstar:DistributionNormal	-0.432440	0.071930	-6.012	1.83e-09	***
zAstar:DistributionSinh	-0.423255	0.072891	-5.807	6.37e-09	***
zAstar:PIRLow	0.588733	0.074836	7.867	3.63e-15	***
zAstar:PIRMedium	-0.087976	0.076810	-1.145	0.252060	
zk:VDR_YMedium	-0.121220	0.052653	-2.302	0.021322	*
zk:VDR_YSlow	0.231546	0.052547	4.406	1.05e-05	***
zs:PIRLow	-0.421089	0.052892	-7.961	1.70e-15	***
zs:PIRMedium	-0.069734	0.052025	-1.340	0.180113	
zNobs:VDR_YMedium	-0.270715	0.053437	-5.066	4.06e-07	***
zNobs:VDR_YSlow	-0.353425	0.052873	-6.684	2.32e-11	***
zk:zSNR_X	0.106992	0.018883	5.666	1.46e-08	***
zNobs:zk	0.102091	0.018761	5.442	5.28e-08	***
zm:RCFLow	-0.339869	0.055194	-6.158	7.38e-10	***
zm:RCFMedium	-0.089156	0.055020	-1.620	0.105144	
zSNR_X:VDR_XMedium	-0.009139	0.052213	-0.175	0.861049	
zSNR_X:VDR_XSlow	-0.291520	0.052419	-5.561	2.68e-08	***

```
zm:DistributionNormal          -0.283927   0.054392  -5.220 1.79e-07 ***
zm:DistributionSinh            -0.187354   0.053320  -3.514 0.000442 ***
zm:zs                           0.081208   0.023686   3.429 0.000607 ***
zSNR_Y:zm                      -0.066302   0.018577  -3.569 0.000358 ***
zSNR_X:RCFLow                  -0.234852   0.052622  -4.463 8.08e-06 ***
zSNR_X:RCFMedium               -0.015537   0.054008  -0.288 0.773587
zNobs:DistributionNormal       -0.239751   0.053199  -4.507 6.58e-06 ***
zNobs:DistributionSinh         -0.176795   0.054090  -3.269 0.001081 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
       Estimate Std. Error z value
-8|-7 -12.84686    1.00713 -12.756
-7|-6  -9.43214    0.22051 -42.774
-6|-5  -7.28653    0.13572 -53.689
-5|-4  -5.13949    0.11322 -45.395
-4|-3  -3.60702    0.10356 -34.831
-3|-2  -1.99904    0.09625 -20.769
-2|-1  -0.47497    0.09274  -5.121
-1|0    1.79520    0.09347  19.205
0|1    10.77083    0.20676  52.094
1|2    11.53834    0.25168  45.845
2|3    12.59925    0.36877  34.165
3|4    13.41020    0.52429  25.578
4|5    14.10334    0.72449  19.467

> ci2 <- confint(fm2)
> ci2
                                    2.5 %       97.5 %
zNobs                          0.79625652  0.987155719
zk                            -0.07954517  0.137297909
zAstar                        -5.23573598 -4.852575554
zSNR_X                         0.44489270  0.634082902
zSNR_Y                         0.07648361  0.160607261
zm                             0.44563553  0.647815530
zs                             1.88473385  2.177726311
VDR_XMedium                    2.06864556  2.561478408
VDR_XSlow                      2.33045553  2.781642374
VDR_YMedium                   -0.12353869  0.078693897
VDR_YSlow                      0.29533969  0.485119189
PIRLow                         1.69276478  1.999464216
PIRMedium                      0.49441963  0.828862012
RCFLow                        -0.18117796  0.009174205
RCFMedium                     -0.23836352 -0.033866954
DistributionNormal            -0.05631937  0.278778805
DistributionSinh               0.09878888  0.449654298
zs:VDR_XMedium                 0.93124943  1.137324289
zs:VDR_XSlow                   1.71245755  1.936360448
```

```
zAstar:zs                      -1.64674480 -1.406187985
VDR_XMedium:DistributionNormal  1.21903393  1.749551126
VDR_XSlow:DistributionNormal   -1.36282508 -0.889873406
VDR_XMedium:DistributionSinh    0.99190466  1.527565400
VDR_XSlow:DistributionSinh     -0.64368188 -0.148499945
VDR_XMedium:PIRLow             -1.76674700 -1.259214986
VDR_XSlow:PIRLow               -1.77354557 -1.337537330
VDR_XMedium:PIRMedium          -0.60067234 -0.074145695
VDR_XSlow:PIRMedium            -0.32119521  0.152727913
zk:zs                           0.18427803  0.256794797
zk:DistributionNormal           0.46907296  0.672921348
zk:DistributionSinh             0.58515627  0.796098845
zAstar:VDR_XMedium              0.81253010  1.125977487
zAstar:VDR_XSlow                0.20842022  0.497752689
zk:VDR_XMedium                 -0.46193970 -0.250439090
zk:VDR_XSlow                   -0.08635677  0.124711467
zAstar:DistributionNormal      -0.57373818 -0.291758284
zAstar:DistributionSinh        -0.56667826 -0.280906967
zAstar:PIRLow                   0.44213523  0.735517282
zAstar:PIRMedium               -0.23887172  0.062262750
zk:VDR_YMedium                 -0.22442459 -0.018015050
zk:VDR_YSlow                    0.12864400  0.334640652
zs:PIRLow                      -0.52482442 -0.317474794
zs:PIRMedium                   -0.17171861  0.032230410
zNobs:VDR_YMedium              -0.37551041 -0.166029685
zNobs:VDR_YSlow                -0.45712176 -0.249853400
zk:zSNR_X                       0.07001119  0.144036714
zNobs:zk                        0.06533836  0.138885372
zm:RCFLow                      -0.44807759 -0.231708294
zm:RCFMedium                   -0.19701641  0.018673318
zSNR_X:VDR_XMedium             -0.11144960  0.093234184
zSNR_X:VDR_XSlow               -0.39429061 -0.188799436
zm:DistributionNormal          -0.39060807 -0.177381247
zm:DistributionSinh            -0.29190687 -0.082881440
zm:zs                           0.03483408  0.127688051
zSNR_Y:zm                      -0.10274554 -0.029919039
zSNR_X:RCFLow                  -0.33802305 -0.131737160
zSNR_X:RCFMedium               -0.12139054  0.090328250
zNobs:DistributionNormal       -0.34408812 -0.135542105
zNobs:DistributionSinh         -0.28288858 -0.070848662

> 1-fm2$logLik/fmnull$logLik
[1] 0.3996325
```

Appendix D

Prediction RMSE Logistic Models

D.1 PLS1 Prediction RMSE Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(PRED_RMSE_BestLVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC_PRED)
> fmnull$logLik
[1] -20052.88

> fm1 <- clm(ordered(PRED_RMSE_BestLVlessAstar) ~ zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,link="logit",
> summary(fm1)
formula:
ordered(PRED_RMSE_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC_PRED

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -13181.21 26422.42 9(0)  2.36e-12 2.0e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.147797   0.019621   7.533 4.98e-14 ***
zk                 -0.005655   0.019529  -0.290  0.77212
zAstar             -3.747262   0.043584 -85.978  < 2e-16 ***
zSNR_X              0.377115   0.019845  19.003  < 2e-16 ***
zSNR_Y              0.132185   0.019616   6.739 1.60e-11 ***
VDRMedium          -0.103835   0.048006  -2.163  0.03054 *
VDRSlow            -0.515323   0.047582 -10.830  < 2e-16 ***
PIRLow              0.546079   0.048220  11.325  < 2e-16 ***
PIRMedium           0.258494   0.046654   5.541 3.01e-08 ***
DistributionNormal -0.129605   0.047277  -2.741  0.00612 **
DistributionSkew   -0.210925   0.047632  -4.428 9.50e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -6.79335    0.09214  -73.73
-7|-6 -5.28341    0.08097  -65.25
-6|-5 -4.04557    0.07332  -55.18
-5|-4 -2.71237    0.06590  -41.16
-4|-3 -1.09720    0.06090  -18.02
-3|-2  0.39323    0.05976    6.58
-2|-1  1.94033    0.06279   30.90
-1|0   3.91756    0.07216   54.29
0|1    7.17236    0.09949   72.09
1|2    7.85422    0.11797   66.58
2|3    8.32593    0.13727   60.65
3|4    8.79925    0.16367   53.76
4|5    9.18680    0.19174   47.91
5|6    9.65925    0.23561   41.00
6|7    9.94802    0.26866   37.03
7|8   10.17195    0.29807   34.13
8|9   10.57864    0.36127   29.28
9|10  10.86714    0.41495   26.19
10|13 11.27375    0.50549   22.30

> ci1 <- confint(fm1)
> ci1
                          2.5 %       97.5 %
zNobs                0.10935706  0.186272752
zk                  -0.04391619  0.032638171
zAstar              -3.83322332 -3.662371453
zSNR_X               0.33826337  0.416057159
zSNR_Y               0.09375472  0.170649640
VDRMedium           -0.19794002 -0.009754233
VDRSlow             -0.60864694 -0.422123185
PIRLow               0.45163511  0.640658457
PIRMedium            0.16707907  0.349963514
DistributionNormal -0.22228419 -0.036953953
DistributionSkew   -0.30431071 -0.117588956

> 1-fm1$logLik/fmnull$logLik
[1] 0.3426775

> sum(is.na(SIM_DATA_ABC_PRED$PRED_RMSE_BestLVlessAstar)==FALSE)
[1] 9000
> klogn<-log(9000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)

> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC_PRED)
> summary(fm2)
formula:
```

```
ordered(PRED_RMSE_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution + zAstar:zSN
data:    SIM_DATA_ABC_PRED

 link  threshold nobs logLik    AIC      niter max.grad cond.H
 logit flexible  9000 -12889.83 25897.67 9(0)  5.03e-12 3.5e+03

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
zNobs                       0.16855    0.02001   8.423  < 2e-16 ***
zk                          0.05449    0.05177   1.052 0.292594
zAstar                     -3.91071    0.06889 -56.771  < 2e-16 ***
zSNR_X                      0.17326    0.03381   5.125 2.97e-07 ***
zSNR_Y                      0.02202    0.03552   0.620 0.535281
VDRMedium                  -0.26265    0.11093  -2.368 0.017893 *
VDRSlow                    -0.13986    0.10943  -1.278 0.201232
PIRLow                      0.92500    0.08800  10.512  < 2e-16 ***
PIRMedium                   0.37890    0.08267   4.584 4.57e-06 ***
DistributionNormal         -0.03602    0.08511  -0.423 0.672146
DistributionSkew           -0.50864    0.08842  -5.752 8.80e-09 ***
zAstar:zSNR_X               0.18061    0.01925   9.384  < 2e-16 ***
VDRMedium:PIRLow           -0.03124    0.12112  -0.258 0.796438
VDRSlow:PIRLow             -0.94939    0.12084  -7.856 3.95e-15 ***
VDRMedium:PIRMedium        -0.01291    0.11746  -0.110 0.912490
VDRSlow:PIRMedium          -0.37014    0.11624  -3.184 0.001451 **
zNobs:zAstar                0.13745    0.01900   7.236 4.63e-13 ***
zAstar:DistributionNormal  -0.17084    0.04967  -3.439 0.000583 ***
zAstar:DistributionSkew     0.22971    0.05084   4.519 6.23e-06 ***
zk:DistributionNormal       0.13281    0.04863   2.731 0.006309 **
zk:DistributionSkew        -0.22957    0.04892  -4.693 2.69e-06 ***
zAstar:VDRMedium           -0.08434    0.05112  -1.650 0.098953 .
zAstar:VDRSlow             -0.35534    0.04997  -7.112 1.15e-12 ***
zAstar:PIRLow               0.36588    0.05102   7.171 7.46e-13 ***
zAstar:PIRMedium            0.24769    0.04877   5.079 3.79e-07 ***
zNobs:zk                   -0.09998    0.01953  -5.119 3.08e-07 ***
zSNR_X:PIRLow               0.36605    0.05033   7.273 3.52e-13 ***
zSNR_X:PIRMedium            0.27518    0.04797   5.736 9.69e-09 ***
zk:zAstar                  -0.08235    0.01877  -4.387 1.15e-05 ***
zk:PIRLow                  -0.31825    0.04909  -6.483 8.97e-11 ***
zk:PIRMedium               -0.10151    0.04838  -2.098 0.035874 *
VDRMedium:DistributionNormal  0.12245    0.12127   1.010 0.312632
VDRSlow:DistributionNormal -0.26784    0.11861  -2.258 0.023930 *
VDRMedium:DistributionSkew  0.38022    0.12079   3.148 0.001645 **
VDRSlow:DistributionSkew    0.48497    0.12181   3.981 6.85e-05 ***
zSNR_Y:DistributionNormal   0.22166    0.04894   4.529 5.93e-06 ***
zSNR_Y:DistributionSkew     0.12321    0.05064   2.433 0.014971 *
zk:zSNR_X                  -0.06888    0.01929  -3.571 0.000356 ***
zk:VDRMedium                0.10604    0.05086   2.085 0.037073 *
zk:VDRSlow                  0.21955    0.04885   4.494 6.99e-06 ***
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -7.06809    0.11382 -62.101
-7|-6 -5.41316    0.10320 -52.452
-6|-5 -4.04516    0.09647 -41.930
-5|-4 -2.63963    0.09060 -29.135
-4|-3 -0.98559    0.08612 -11.445
-3|-2  0.53096    0.08486   6.257
-2|-1  2.10547    0.08761  24.033
-1|0   4.12570    0.09555  43.181
0|1    7.43972    0.11839  62.840
1|2    8.12540    0.13446  60.428
2|3    8.59952    0.15174  56.671
3|4    9.07450    0.17601  51.558
4|5    9.46356    0.20239  46.758
5|6    9.93770    0.24439  40.663
6|7   10.22739    0.27641  37.001
7|8   10.45175    0.30507  34.260
8|9   10.85873    0.36707  29.582
9|10  11.14739    0.42001  26.541
10|13 11.55397    0.50965  22.670

> ci2 <- confint(fm2)
> ci2
                                    2.5 %       97.5 %
zNobs                          0.129348911  0.207794831
zk                            -0.046898373  0.156072482
zAstar                        -4.046199611 -3.776159852
zSNR_X                         0.107017918  0.239542492
zSNR_Y                        -0.047580812  0.091651974
VDRMedium                     -0.480109674 -0.045260895
VDRSlow                       -0.354363784  0.074618838
PIRLow                         0.752678903  1.097647865
PIRMedium                      0.216924828  0.540989776
DistributionNormal            -0.202857260  0.130806671
DistributionSkew              -0.682014266 -0.335377590
zAstar:zSNR_X                  0.142913873  0.218366119
VDRMedium:PIRLow              -0.268684930  0.206121849
VDRSlow:PIRLow                -1.186384180 -0.712669773
VDRMedium:PIRMedium           -0.243141256  0.217310959
VDRSlow:PIRMedium             -0.598022543 -0.142365183
zNobs:zAstar                   0.100242294  0.174710841
zAstar:DistributionNormal     -0.268222299 -0.073495733
zAstar:DistributionSkew        0.130108745  0.329397097
zk:DistributionNormal          0.037506811  0.228123979
zk:DistributionSkew           -0.325470683 -0.133714885
zAstar:VDRMedium              -0.184557529  0.015833265
```

```
zAstar:VDRSlow               -0.453320686 -0.257449748
zAstar:PIRLow                 0.265895499  0.465917709
zAstar:PIRMedium              0.152141147  0.343310345
zNobs:zk                     -0.138282183 -0.061710761
zSNR_X:PIRLow                 0.267444676  0.464746450
zSNR_X:PIRMedium              0.181177559  0.369235658
zk:zAstar                    -0.119191376 -0.045601801
zk:PIRLow                    -0.414498010 -0.222069093
zk:PIRMedium                 -0.196355095 -0.006708676
VDRMedium:DistributionNormal -0.115237075  0.360158750
VDRSlow:DistributionNormal   -0.500348691 -0.035405933
VDRMedium:DistributionSkew    0.143491970  0.616999239
VDRSlow:DistributionSkew      0.246263235  0.723781311
zSNR_Y:DistributionNormal     0.125759215  0.317621838
zSNR_Y:DistributionSkew       0.023964731  0.222474029
zk:zSNR_X                    -0.106695236 -0.031075125
zk:VDRMedium                  0.006365779  0.205744454
zk:VDRSlow                    0.123784122  0.315301709

> 1-fm2$logLik/fmnull$logLik
[1] 0.3572079
```

D.2 PLS1 Prediction Coverage Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(PRED_InCI_BestLVlessAstar) ~ 1,link="logit",data=SIM_DATA_ABC_PRED)
> fmnull$logLik
[1] -17638.6

> fm1 <- clm(ordered(PRED_InCI_BestLVlessAstar) ~ zNobs+zk+zAstar+zSNR_X+zSNR_Y+VDR+PIR+Distribution,link="logit",
> summary(fm1)
formula:
ordered(PRED_InCI_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC_PRED

 link  threshold nobs logLik   AIC     niter max.grad cond.H
 logit flexible  9000 -1712.41 3468.82 11(0) 8.98e-07 6.0e+02

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               -0.13569    0.05912  -2.295   0.0217 *
zk                  -0.05175    0.05881  -0.880   0.3789
zAstar             -18.60456    0.28420 -65.463   <2e-16 ***
zSNR_X               0.07464    0.05865   1.273   0.2032
zSNR_Y               0.01655    0.05845   0.283   0.7771
VDRMedium            0.01272    0.13290   0.096   0.9237
```

```
VDRSlow              0.04187    0.13246   0.316   0.7519
PIRLow               0.02768    0.13331   0.208   0.8355
PIRMedium           -0.06264    0.13335  -0.470   0.6386
DistributionNormal  -0.08484    0.13697  -0.619   0.5356
DistributionSkew     0.29990    0.13546   2.214   0.0268 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -21.9632     0.4040 -54.359
-7|-6 -16.2025     0.3191 -50.780
-6|-5  -8.1212     0.2337 -34.755
-5|-4  -1.5910     0.2065  -7.706
-4|-3   6.0761     0.2039  29.806
-3|-2  13.5154     0.2848  47.459
-2|-1  19.7296     0.3622  54.471
-1|0   28.7493     0.4557  63.088
0|6    29.6656     0.5317  55.798
6|7    30.3587     0.6385  47.548
7|8    31.7450     1.0759  29.504

> ci1 <- confint(fm1)
> ci1
                          2.5 %       97.5 %
zNobs               -0.25204396  -0.02014037
zk                  -0.16742870   0.06311389
zAstar             -19.17216624 -18.05771335
zSNR_X              -0.04016491   0.18991524
zSNR_Y              -0.09800991   0.13126065
VDRMedium           -0.24782793   0.27342281
VDRSlow             -0.21776166   0.30176787
PIRLow              -0.23352464   0.28935771
PIRMedium           -0.32417679   0.19884263
DistributionNormal  -0.35349050   0.18372588
DistributionSkew     0.03498069   0.56631000

> 1-fm1$logLik/fmnull$logLik
[1] 0.902917

> sum(is.na(SIM_DATA_ABC_PRED$PRED_InCI_BestLVlessAstar)==FALSE)
[1] 9000
> klogn<-log(9000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM_DATA_ABC_PRED)
> summary(fm2)
formula:
ordered(PRED_InCI_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + VDR + PIR + Distribution
data:    SIM_DATA_ABC_PRED
```

```
 link  threshold nobs logLik   AIC     niter max.grad cond.H
 logit flexible  9000 -1712.41 3468.82 11(0) 8.98e-07 6.0e+02

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               -0.13569    0.05912  -2.295   0.0217 *
zk                  -0.05175    0.05881  -0.880   0.3789
zAstar             -18.60456    0.28420 -65.463   <2e-16 ***
zSNR_X               0.07464    0.05865   1.273   0.2032
zSNR_Y               0.01655    0.05845   0.283   0.7771
VDRMedium            0.01272    0.13290   0.096   0.9237
VDRSlow              0.04187    0.13246   0.316   0.7519
PIRLow               0.02768    0.13331   0.208   0.8355
PIRMedium           -0.06264    0.13335  -0.470   0.6386
DistributionNormal  -0.08484    0.13697  -0.619   0.5356
DistributionSkew     0.29990    0.13546   2.214   0.0268 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -21.9632     0.4040 -54.359
-7|-6 -16.2025     0.3191 -50.780
-6|-5  -8.1212     0.2337 -34.755
-5|-4  -1.5910     0.2065  -7.706
-4|-3   6.0761     0.2039  29.806
-3|-2  13.5154     0.2848  47.459
-2|-1  19.7296     0.3622  54.471
-1|0   28.7493     0.4557  63.088
0|6    29.6656     0.5317  55.798
6|7    30.3587     0.6385  47.548
7|8    31.7450     1.0759  29.504

# Same as linear screening model!
```

D.3 PLS2 Prediction RMSE Logistic Model

```
> require(ordinal)

> fmnull <- clm(ordered(PRED_RMSE_BestLVlessAstar) ~ 1,link="logit",data=SIM2_PRED_DATA_AtoT)
> fmnull$logLik
[1] -22948.88

> fm1 <- clm(ordered(PRED_RMSE_BestLVlessAstar) ~ zNobs+zk+zAstar+zSNR_X+zSNR_Y+zm+zs+VDR_X+VDR_Y+PIR+RCF+Distribu
> summary(fm1)
formula:
ordered(PRED_RMSE_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs + VDR_X + VDR_Y + PIR + RCF +
data:    SIM2_PRED_DATA_AtoT

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -18074.65 36223.30 10(1) 5.46e-09 3.1e+03

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
zNobs               0.50968    0.01821  27.989  < 2e-16 ***
zk                  0.32810    0.01833  17.900  < 2e-16 ***
zAstar             -2.24578    0.02932 -76.597  < 2e-16 ***
zSNR_X              0.33258    0.01780  18.682  < 2e-16 ***
zSNR_Y              0.06985    0.01758   3.974 7.08e-05 ***
zm                  0.07184    0.01792   4.009 6.10e-05 ***
zs                  0.91597    0.02132  42.959  < 2e-16 ***
VDR_XMedium         1.66360    0.04661  35.695  < 2e-16 ***
VDR_XSlow           1.40978    0.04173  33.787  < 2e-16 ***
VDR_YMedium         0.08685    0.04328   2.007   0.0448 *
VDR_YSlow           0.40611    0.04050  10.026  < 2e-16 ***
PIRLow              0.47211    0.04064  11.616  < 2e-16 ***
PIRMedium           0.09792    0.04300   2.277   0.0228 *
RCFLow             -0.20081    0.04050  -4.958 7.13e-07 ***
RCFMedium          -0.21634    0.04344  -4.981 6.33e-07 ***
DistributionNormal -0.03165    0.04321  -0.732   0.4639
DistributionSinh    0.07990    0.04406   1.813   0.0698 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -5.61670    0.11827 -47.492
-7|-6 -3.95218    0.07785 -50.767
-6|-5 -2.82788    0.06714 -42.121
-5|-4 -1.86971    0.06210 -30.109
-4|-3 -1.06409    0.05966 -17.836
-3|-2 -0.30050    0.05878  -5.113
-2|-1  0.68009    0.05927  11.474
-1|0   2.11866    0.06186  34.247
```

```
0|1    6.98939    0.10259  68.132
1|2    7.71389    0.11758  65.608
2|3    8.20368    0.13358  61.416
3|4    8.59914    0.15113  56.898
4|5    8.89348    0.16750  53.094
5|6    9.30553    0.19616  47.440
6|7    9.73187    0.23418  41.556
7|8   10.00743    0.26405  37.900
8|10  10.21807    0.29008  35.225
10|11 10.38735    0.31326  33.159
11|12 10.84392    0.38741  27.991
12|13 11.18215    0.45523  24.564

> 1-fm1$logLik/fmnull$logLik
[1] 0.212395

> ci1 <- confint(fm1)
> ci1
                          2.5 %      97.5 %
zNobs                0.474042916  0.54542603
zk                   0.292229894  0.36408381
zAstar              -2.303534775 -2.18860006
zSNR_X               0.297724463  0.36750868
zSNR_Y               0.035403583  0.10430697
zm                   0.036730365  0.10698573
zs                   0.874289213  0.95787374
VDR_XMedium          1.572421549  1.75511618
VDR_XSlow            1.328124446  1.49168813
VDR_YMedium          0.002045352  0.17169567
VDR_YSlow            0.326760038  0.48553690
PIRLow               0.392488434  0.55181029
PIRMedium            0.013664104  0.18221865
RCFLow              -0.280213638 -0.12143701
RCFMedium           -0.301479922 -0.13121124
DistributionNormal -0.116368848  0.05302668
DistributionSinh   -0.006468624  0.16624661

> sum(is.na(SIM2_PRED_DATA_AtoT$PRED_RMSE_BestLVlessAstar)==FALSE)
[1] 12000
> klogn<-log(12000)
> fmstep <- step(fm1,scope=~.^2, direction="forward", k=klogn)
> fm2 <- clm(formula=fmstep$formula,link="logit",data=SIM2_PRED_DATA_AtoT)
> summary(fm2)
formula:
ordered(PRED_RMSE_BestLVlessAstar) ~ zNobs + zk + zAstar + zSNR_X + zSNR_Y + zm + zs + VDR_X + VDR_Y + PIR + RCF +
data:    SIM2_PRED_DATA_AtoT

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12000 -17185.35 34500.70 10(1) 1.81e-09 5.0e+03
```

```
Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
zNobs                            0.611283   0.030783  19.858  < 2e-16 ***
zk                               0.332352   0.033802   9.832  < 2e-16 ***
zAstar                          -3.354674   0.061970 -54.134  < 2e-16 ***
zSNR_X                           0.344370   0.030981  11.115  < 2e-16 ***
zSNR_Y                           0.076290   0.017854   4.273 1.93e-05 ***
zm                               0.148141   0.020562   7.205 5.82e-13 ***
zs                               1.127889   0.052333  21.552  < 2e-16 ***
VDR_XMedium                      1.361435   0.104490  13.029  < 2e-16 ***
VDR_XSlow                        1.483647   0.095249  15.577  < 2e-16 ***
VDR_YMedium                      0.168668   0.044815   3.764 0.000167 ***
VDR_YSlow                        0.400193   0.041635   9.612  < 2e-16 ***
PIRLow                           0.935639   0.065985  14.179  < 2e-16 ***
PIRMedium                        0.147010   0.071378   2.060 0.039438 *
RCFLow                          -0.174020   0.041297  -4.214 2.51e-05 ***
RCFMedium                       -0.159707   0.044545  -3.585 0.000337 ***
DistributionNormal              -0.598398   0.072686  -8.233  < 2e-16 ***
DistributionSinh                -0.291011   0.074859  -3.887 0.000101 ***
zs:VDR_XMedium                   0.884491   0.047320  18.692  < 2e-16 ***
zs:VDR_XSlow                     1.134884   0.048717  23.295  < 2e-16 ***
zAstar:zs                       -0.787182   0.042363 -18.582  < 2e-16 ***
zNobs:zk                         0.179877   0.015963  11.268  < 2e-16 ***
VDR_XMedium:DistributionNormal   1.131414   0.113291   9.987  < 2e-16 ***
VDR_XSlow:DistributionNormal     0.514416   0.103002   4.994 5.91e-07 ***
VDR_XMedium:DistributionSinh     0.941346   0.114339   8.233  < 2e-16 ***
VDR_XSlow:DistributionSinh       0.205281   0.105589   1.944 0.051877 .
zAstar:zSNR_X                    0.148215   0.015868   9.341  < 2e-16 ***
zNobs:zAstar                     0.123416   0.016041   7.694 1.43e-14 ***
zAstar:VDR_XMedium               0.457758   0.052492   8.721  < 2e-16 ***
zAstar:VDR_XSlow                 0.181977   0.051887   3.507 0.000453 ***
VDR_XMedium:PIRLow              -0.670940   0.105824  -6.340 2.30e-10 ***
VDR_XSlow:PIRLow                -0.575016   0.095488  -6.022 1.72e-09 ***
VDR_XMedium:PIRMedium           -0.003717   0.111291  -0.033 0.973359
VDR_XSlow:PIRMedium              0.107969   0.101657   1.062 0.288195
zm:zs                            0.127804   0.020986   6.090 1.13e-09 ***
zAstar:VDR_YMedium               0.035841   0.046471   0.771 0.440561
zAstar:VDR_YSlow                 0.242356   0.046851   5.173 2.30e-07 ***
zk:zSNR_X                       -0.070231   0.015831  -4.436 9.15e-06 ***
zk:DistributionNormal           -0.060244   0.045214  -1.332 0.182719
zk:DistributionSinh              0.173009   0.045789   3.778 0.000158 ***
zNobs:zSNR_X                    -0.059224   0.014870  -3.983 6.81e-05 ***
zSNR_X:VDR_YMedium              -0.146699   0.043575  -3.367 0.000761 ***
zSNR_X:VDR_YSlow                 0.088417   0.043924   2.013 0.044120 *
zNobs:VDR_XMedium               -0.075466   0.043967  -1.716 0.086088 .
zNobs:VDR_XSlow                 -0.206907   0.043304  -4.778 1.77e-06 ***
zAstar:zm                       -0.057413   0.015892  -3.613 0.000303 ***
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Threshold coefficients:
      Estimate Std. Error z value
-8|-7 -6.67580    0.13455 -49.616
-7|-6 -4.93991    0.09934 -49.726
-6|-5 -3.72281    0.08966 -41.522
-5|-4 -2.63438    0.08436 -31.229
-4|-3 -1.66539    0.08115 -20.522
-3|-2 -0.72671    0.07956  -9.134
-2|-1  0.43640    0.07941   5.496
-1|0   2.01597    0.08124  24.814
0|1    6.46199    0.10799  59.841
1|2    7.19251    0.12241  58.756
2|3    7.68490    0.13786  55.744
3|4    8.08096    0.15492  52.164
4|5    8.37551    0.17092  49.003
5|6    8.78856    0.19909  44.143
6|7    9.21687    0.23668  38.942
7|8    9.49340    0.26629  35.651
8|10   9.70422    0.29213  33.219
10|11  9.87321    0.31516  31.327
11|12 10.32869    0.38895  26.555
12|13 10.66654    0.45654  23.364

> ci2 <- confint(fm2)
> ci2
                                       2.5 %      97.5 %
zNobs                            0.551021256  0.67169259
zk                               0.266189870  0.39869887
zAstar                          -3.476509864 -3.23358551
zSNR_X                           0.283684884  0.40513324
zSNR_Y                           0.041303956  0.11129291
zm                               0.107857104  0.18846159
zs                               1.025441221  1.23058819
VDR_XMedium                      1.156821539  1.56643488
VDR_XSlow                        1.297098574  1.67048554
VDR_YMedium                      0.080865658  0.25654219
VDR_YSlow                        0.318623575  0.48183377
PIRLow                           0.806395141  1.06506451
PIRMedium                        0.007153148  0.28696578
RCFLow                          -0.254978306 -0.09309317
RCFMedium                       -0.247019680 -0.07239959
DistributionNormal              -0.740967909 -0.45602851
DistributionSinh                -0.437800494 -0.14434275
zs:VDR_XMedium                   0.791842326  0.97734354
zs:VDR_XSlow                     1.039520954  1.23049967
zAstar:zs                       -0.870260353 -0.70419624
zNobs:zk                         0.148599512  0.21117730
```

```
VDR_XMedium:DistributionNormal  0.909463462  1.35357393
VDR_XSlow:DistributionNormal    0.312570758  0.71634554
VDR_XMedium:DistributionSinh    0.717320887  1.16554035
VDR_XSlow:DistributionSinh     -0.001667591  0.41224703
zAstar:zSNR_X                   0.117113385  0.17931826
zNobs:zAstar                    0.091943685  0.15482878
zAstar:VDR_XMedium              0.354876018  0.56065087
zAstar:VDR_XSlow                0.080284183  0.28368563
VDR_XMedium:PIRLow             -0.878425559 -0.46358897
VDR_XSlow:PIRLow               -0.762216586 -0.38790156
VDR_XMedium:PIRMedium          -0.221834317  0.21443421
VDR_XSlow:PIRMedium            -0.091272714  0.30722788
zm:zs                           0.086695114  0.16896108
zAstar:VDR_YMedium             -0.055252298  0.12691989
zAstar:VDR_YSlow                0.150526823  0.33418820
zk:zSNR_X                      -0.101265301 -0.03920808
zk:DistributionNormal          -0.148879949  0.02836215
zk:DistributionSinh             0.083265299  0.26276146
zNobs:zSNR_X                   -0.088374568 -0.03008282
zSNR_X:VDR_YMedium             -0.232113470 -0.06129764
zSNR_X:VDR_YSlow                0.002332019  0.17451681
zNobs:VDR_XMedium              -0.161646137  0.01070928
zNobs:VDR_XSlow                -0.291813010 -0.12205975
zAstar:zm                      -0.088565587 -0.02626807

> 1-fm2$logLik/fmnull$logLik
[1] 0.2511463
```

Bibliography

[1] Akaike, H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control* , (6),19, 716-723.

[2] Aly, E-E. AA. (1990) Simple tests for dispersive ordering. *Statistics & Probability Letters*, 9, (4), 323-325.

[3] Arneberg, R., Rajalahti, T., Flikka, K, Berven, K.S, Kroksveen, A.C., Berle, M., Myhr, K.-M., Vedeler, C.A.,Ulvik, R.J., and Kvalheim, O.M. (2007), Pretreatment of Mass Spectral Profiles : Application to Proteomic Data *Analytical Chemistry* 79, (18), 70147026.

[4] Balakrishnan, N. and Lai, C.-D. (2009) *Continuous Bivariate Distributions*. 2nd Edition. New York, Springer-Verlag (2009)

[5] Barker M, Rayens W. (2003) Partial least squares for discrimination. *Journal of Chemometrics* ,17, (3), 166173.

[6] Barnet, S. (1990) *Matrices. Methods and Applications*. Oxford Applied Mathematics and Computing Science Series. Oxford, Clarendon Press.

[7] Boulet, J.-C. and Roger, J.-M. (2012) Pretreatments by means of orthogonal projections. *Chemometrics and Intelligent Laboratory Systems*, (1), 117, 61-69.

[8] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C. (1984) *Classification and Regression Trees*. Belmont, California, Wadsworth.

[9] Bro, R. and Eldén, R. (2009) PLS works. *Journal of Chemometrics*, 23, (2), 69-71.

[10] Brown, P., Fearn, T., Vannucci, M., (2001). Bayesian Wavelet Regression on Curves with Application to a Spectroscopic Calibration Problem. *Journal of the American Statistical Association*, 96, (454), 398408.

[11] Burnham, A.J., Viveros, R. and MacGregor, J.F. (1996). Frameworks for multivariate latent variable regression. *Journal of Chemometrics*, 10, (1), 31-45.

[12] Christensen, R.H.B. (2015). Ordinal : Regression Models for Ordinal Data. R package version 2015.1-21, URL http://CRAN.R-project.org/package=ordinal.

[13] Chun, H. and Keles, S. (2010) Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical. Society*, B, 72, (1), 3-25.

[14] Clark, M. and Cramer, R.D. III, (1993). The Probability of Chance Correlation using Partial Least Squares (PLS)., *Quantitative Structural-Activity Relationships*, 12, (2), 137-145.

[15] Cloarec, O. (2014) Can we beat over-fitting? *Journal of Chemometrics* 28, (8), 610 - 614.

[16] de Jong, S. (1993) SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18, (3), 251-263.

[17] Denham. M.C. (1997) Prediction intervals in partial least squares, *Journal of Chemometrics*, 11, (1), 39-52.

[18] Derrico, J. for function derivest.m in MATLAB® File Exchange.

[19] Dias, C.T.d.S., Samaranayaka, A. and Manly, B. (2008). On the use of correlated beta random variables with animal population modelling. *Ecological Modelling* 215, 293-300.

[20] Di Ruscio, D. (1997) *On the Partial Least Squares Algorithm*, BIBSYS, r97005598

[21] Dormann, C.F. et al. (2013) Collinearity: a review of methods to deal with it and a simulation study evaluating their performance *Ecography*, 36, (1), 2746.

[22] Eastment, H.T. and Krzanowski, W.J *Cross-validatory choice of the number of components from a principal component analysis*, Technometrics, 24, 73-77.

[23] Efron, B. (2004) The Estimation of Prediction Error: Covariance Penalties and Cross-Validation *Journal of the American Statistical Association*, 99, (467), 619-633.

[24] Eldén, L. (2004) Partial least-squares vs. Lanczos bidiagonalization - I: analysis of a projection method for multiple regression. *Computational Statistics and Data Analysis*, 46, (1), 11-31.

[25] Ergon, R. (2009), Re-interpretation of NIPALS results solves PLSR inconsistency problem, *Chemometrics and Intelligent Laboratory Systems*, 23, (1), 72-75.

[26] Ergon, R., Halstensen,M. and Erbensen, K.H. (2011) Model Choice and Squared Prediction Errors in PLS Regression, *Journal of Chemometrics*, 25, (6), 301-312.

[27] Eriksson, L., Johansson, E., Kettaneh-Wold, N., Trygg, J., Wikström, C. and Wold, S. (2006) *Multi- and Megavariate Data Analysis. Part I. Basic Principles and Applications (2nd revised and enlarged edition).* Umeå, Umetrics Academy. ISBN-10: 91-973730-2-8

[28] Eriksson, L., Johansson, E., Kettaneh-Wold, N., Trygg, J., Wikström, C. and Wold, S. (2006) *Multi- and Megavariate Data Analysis. Part II. Advanced Applications and Method Extensions. (2nd revised and enlarged edition).* Umeå, Umetrics Academy. ISBN-10: 91-973730-3-6

[29] Eriksson, L. Trygg, J. and Wold, S. (2008). CV-ANOVA for significance testing of PLS and OPLS models. *Journal of Chemometrics*, 22, (11-12), 594-600.

[30] Faber, N.M. and Rajko, R. (2007) How to avoid over-fitting in multivariate calibration - The conventional validation approach and an alternative. *Analytica Chimica Acta*, 595, (1-2), 98-106.

[31] Faber, N.M. (1999). Estimating the uncertainty in estimates of root mean square error of prediction: application to determining the size of an adequate test set in multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 49, (1), 79-89.

[32] Faber, N.M. (2009) The X-residuals calculated by partial least squares are problematic for uncertainty estimation. textitChemometrics and Intelligent Laboratory Systems, 96, (2), 264-265.

[33] Fonville, J.M., Richards, S.E., Barton, R.H., Boulange,C.L., Ebbels, T.M.D, Nicholson, J.K., Holmes, E. and Dumas, M.-E. (2010) The evolution of partial least squares models and related chemometric approaches in metabonomics and metabolic phenotyping. *Journal of Chemometrics*, 24, (11-12), 636-649.

[34] Fleishman, A. I. (1978). A method for simulating non-normal distributions. *Psychometrika* 43(4), 521-532.

[35] Frank, I.E. and Freidman, J.H. (1993) A Statistical View of Some Chemometrics Regression Tools, *Technometrics*, 35, (2), 109-135.

[36] Garthwaite, P. (1994) An Interpretation of PLS. *Journal of the American Statistical Association* 89, (425), 122-127.

[37] Geladi, P. (1988) Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics*, 2, (4), 231-246.

[38] Gentle, J.E. (2007) *Matrix Algebra : Theory, Computations, and Applications in Statistics* New York, Springer-Verlag.

[39] Gilbert, P. (2012). numDeriv: Accurate Numerical Derivatives. R package version 2012.9-1, URL http://CRAN.R-project.org/package=numDeriv.

[40] Golub, G.H. and Kahan, W. (1965) Calculating the singular values and pseudo-inverse of a matrix, *SIAM Journal of Numerical Analysis*, Series B, 2, 205-224.

[41] Good, P. (2005) *Permutation, Parametric and Bootstrap Tests of Hypotheses* Third Edition, New York, Springer-Verlag.

[42] Goodhue, D. Lewis, W. and Thompson, R. (2006) PLS, Small Sample Size and Statistical Power in MIS Research, in *Proceedings of the 39th Hawaii International Conference on System Sciences*, R. Sprague Jr. (ed.), Los Alamitos, California, IEEE Computer Society Press.

[43] Goutis, C. and Fearn, T., (1996) Partial least squares regression on smooth factors. *Journal of the American Statistical Association*, 91, (434), 627-632.

[44] Gouvénec, S., J.A. Fernández Pierna, J.A., Massart, D.L. and Rutledge, D.N. (2003) An evaluation of the PoLiSh smoothed regression and the Monte Carlo Cross-Validation for the determination of the complexity of a PLS model. *Chemometrics and Intelligent Laboratory Systems*, 68, (1), 41-51.

[45] Harrell, F.E. (2001) Regression Modeling Strategies : *With Applications to Linear Models, Logistic Regression, and Survival Analysis.* New York, Springer-Verlag

[46] Hastie, T., Tibshirani, R.J. and Freidman, J. (2001). *The Elements of Statistical Learning.* New York, Springer-Verlag

[47] Helland, I.S. (1988) On the structure of PLS regression, *Communications in Statistics - Simulation and Computation*, 17, (2), 581-607.

[48] Henseler, J. (2010) On the convergence of the partial least squares path modeling algorithm. *Computational Statistics*, 25, (1), 107-120.

[49] Horst, P. (1961) Relations among m sets of measures. *Psychometrika*, 26, (2), 129-149.

[50] Höskuldsson, A. (1988) PLS Regression Methods, *Journal of Chemometrics*, 2, (3), 211-228.

[51] Hotelling, H. (1936) Relations between two sets of variates. *Biometrika*, 28, (3-4), 321-377.

[52] Hotelling, H. and M.R. Pabst (1936). Rank Correlation and Tests of Significance Involving No Assumption of Normality. *Annals of Mathematical Statistics* 7, 29-43.

[53] Hotelling, H. (1957) The relations of the newer multivariate statistical methods to factor analysis. *British Journal of Statistical Psychology*, 10, 69-79.

[54] HullandRyanRayner2010 *Handbook of Partial Least Squares : Concepts, Methods and Applications*

[55] Hurvich, C.M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples, *Biometrika*, 76, (2), 297-307.

[56] (2012) Hybrid Space-Filling Designs for Computer Experiments. *Frontiers in Statistical Quality Control*, 10, 287-301. Berlin, Springer-Verlag.

[57] Johnson, M.E. Ramberg, J.S. and Wang C. (1982). The Johnson translation system in Monte Carlo studies *Communications in Statistics - Simulation and Computation*, 11(5), 521-525.

[58] Jolliffe, I.T. (2002) *Principal Components Analysis*, Second Edition, Berlin, Springer-Verlag.

[59] Kalivas, J.H. (1997) Two datasets of near infrared spectra, *Chemometrics and Intelligent Laboratory Systems*, 37, (2), 255-259.

[60] Kendall, M.G. (1957), *A Course in Multivariate Analysis*, London, Griffin.

[61] Henk A. L. Kiers, H.A.L. and Smilde, A.K. (2007). A comparison of various methods for multivariate regression with highly collinear variables. *Statistical Methods and Applications*, 16, (2), 193-228.

[62] Kondylis, A. and Whittaker, J. (2012) Feature Selection for functional PLS. *Chemometrics and Intelligent laboratory Systems*. 122, 1, 82-89.

[63] Kowalski, F. *Prediction of wine quality and geographic origin from chemical measurements by partial least-squares regression modeling.*, Analytica Chimica Acta , 162, 241-251, (1984).

[64] Krämer, N., and Braun, M. L. (2013), plsdof: Degrees of Freedom and Confidence Intervals for Partial Least Squares Regression, R package version 0.2-6. URL http://CRAN.R-project.org/package=plsdof.

[65] Krämer, N. and Sugiyama, M. (2011) The Degrees of Freedom of Partial Least Squares Regression, *Journal of the American Statistical Association*, 106, (494), 697-705.

[66] Kvalheim, O.M. (2010) Interpretation of partial least squares regression models by means of target projection and selectivity ratio plots. *Journal of Chemometrics* , 24, (7-8), 496-504.

[67] Li, S. T. and Hammond,J. L. (1975) Generation of pseudo-random numbers with specified univariate distributions and correlation coefficients. *IEEE Transactions Systems Management and Cybernetics*, SMC-5, 557-561.

[68] Li, B., Morris, J. and Martin, E.B. (2002) Model selection for partial least squares regression, *Chemometrics and Intelligent laboratory Systems*. 64, (1), 28, 7989.

[69] Lindgren, F. Geladi, P. and Wold, S. (1993) The Kernel Algorithm for PLS. *Journal of Chemometrics*, 7, (1), 45-59.

[70] Lindgren, F., Hansen, B., Karcher, W. Sjöström, M. and Eriksson, L. (1996) Model Validation by Permutation tests: Applications for Variable Selection. *Journal of Chemometrics*, 10, (4), 521-532.

[71] Lorber, A., Wangen, L.E. and Kowalski, B.R. (1987) A theoretical foundation for the PLS algorithm, *Journal of Chemometrics*, 1, (1), 19-31.

[72] Lyttkens, E. (1996) in *Multivariate Analysis*, ed. by P. R. Krishnaiah, pp. 335-350, New York, Academic Press

[73] Magnus, J.R. and Neudecker, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Chichester, John Wiley.

[74] Manne, R. (1987) Analysis of two partial-least-squares algorithms for multivariate calibration, *Chemometrics and Intelligent Laboratory Systems*, 2, (1-3),187-197.

[75] Marcoulides, G.A. and Saunders, C. (2006) *MIS Quarterly*, 30, (2), iii-ix.

[76] Marcoulides, G.A., Chin, W. and Saunders, C. (2006) *MIS Quarterly*, 31, (1), 171-175.

[77] Martens, H. and Dardenne, P. (1998) Validation and verification of regression in small datasets, *Chemometrics and Intelligent Laboratory Systems*, 44, (1-2), 99-121.

[78] Martens, H., Hoy, M., Westad, F., Folkenberg, D. and Martens, M. (2001) Analysis of designed experiments by stabilising PLS regression and jack-knifing. *Chemometrics and Intelligent Laboratory Systems* ,58, (2), 151-170.

[79] Martens, H. and Martens, M. (2001) *Multivariate Analysis of Quality*, Chichester, John Wiley.

[80] Martens,H. and Næs, T. *Multivariate Calibration*, Chichester, John Wiley.

[81] Massart, D.L., Vandeginste, B.G.M., Buydens, L.M.C., de Jong, S. Lewi, P.J. and Smeyers-Verbek, J. *Handbook of Chemometrics and Qualimetrics: Part B.* Elsevier, 1998.

[82] Mevik, B-H., Wehrens, R. and Liland, K.H. (2012). pls: Partial Least Squares and Principal Component Regression. R package version 2012.2.3-0, URL http://CRAN.R-project.org/package=pls.

[83] McFadden, D. (1973), Conditional Logit Analysis of Qualitative Choice Behavior, *Frontiers in Econometrics*, pp.105-142.

[84] Næs, T. and Martens,H. (1985), Comparison of Prediction Methods for Multicollinear Data, *Communications in Statistics - Simulation and Computation*,14, 545-576.

[85] Osborne, B.G., Fearn, T., Miller, A.R. and Douglas, S. (1984). Application of near infrared reflectance spectroscopy to the compositional analysis of biscuits and biscuit doughs. *Journal of the Science of Food and Agriculture*, 35, 91), 99-105.

[86] Paige, C.C. and Saunders, M.A. (1982). A bidiagonalization algorithm for sparse linear equations and least squares problems. *ACM Transactions in Mathematical Software*. 8, 4371.

[87] Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space *Philosophical Magazine*, 2, (11), 559-572.

[88] Pell, R.J., Ramos, L.S. and Manne, R. (2007) The model space in partial least squares regression, *Journal of Chemometrics*, 21, (3-4), 165-172.

[89] Pesarin, F. and Salmaso, L. (2010) *Permutation Tests for Complex Data: Theory, Applications and Software*, Chichester, John Wiley and Sons.

[90] Phatak, A. and de Jong, S. (1997) The geometry of partial least squares, *Journal of Chemometrics*, 11, (4), 311-338.

[91] Phatak, A. Reilly, P.M. and Penlidis, A. (1993) An approach to interval estimation in partial least squares regression, *Analytica Chimica Acta*, 277, (2), 495-501.

[92] Phatak, A. Reilly, P.M. and Penlidis, A. (2002) The asymptotic variance of the univariate PLS estimator. *Linear Algebra and Its Applications* 354, (1-3), 245-253.

[93] Piepel, G.F., Hicks, R.D., Szchowski, J.M. and Loeppky, J.L., (2002) Methods for Assessing Curvature and Interaction in Mixture Experiments, *Technometrics*, 44, (2), 161-172.

[94] Romera, R. (2010) Prediction intervals in Partial Least Squares regression via a new local linearization approach. *Chemometrics and Intelligent Laboratory Systems*, 103, (2), 122-128.

[95] Ryan, T.P. (2007) *Modern Experimental Design*. John Wiley and Sons.

[96] Schwarz, G.E. (1978) Estimating the dimension of a model. *Annals of Statistics*, 6, (2), 461-464.

[97] Serneels, S. Lamberge, P. and Van Espen, P.J. (2004) Calculation of PLS prediction intervals using efficient recursive relations for the Jacobian matrix, *Journal of Chemometrics*, 18, (1), 76-80

[98] Sjöstrom, M., Wold, S. and Söderström, B. (1986). PLS Discriminant Plot. *Proceedings of PARC in Practice*, Amsterdam, June 19-21, 1985. Elsevier Science Publishers B.V. North-Holland.

[99] Snee, R. D. (1977) Validation of regression models: Methods and examples. *Technometrics*, 19(4), 415-428.

[100] Stone, M. and Brooks, R.T. (1990) Continuum Regression: Cross-Validated Sequentially Constructed Prediction Embracing Ordinary Least Squares, Partial Least Squares and Principal Components Regression, *Journal of the Royal Statistical Society. Series B (Methodological)*, 52, (2), 237-269.

[101] Sundberg, R. (1993) Continuum Regression and Ridge Regression, *Journal of the Royal Statistical Society. Series B (Methodological)*, 55, (3), 653-659.

[102] Tso, M. (1981) Reduced-Rank Regression and Canonical Analysis,*Journal of the Royal Statistical Society. Series B (Methodological)*, 43, (2), 183-189.

[103] Trygg, J, and Wold, S. (2002) Orthogonal projections to latent structures. *Journal of Chemometrics*, 16, (3), 119-128.

[104] van der Voet,H. (1994) Comparing the predictive accuracy of models using a simple randomization test, *Chemometrics and Intelligent Laboratory Systems*, 25, (2), 313-323.

[105] van der Voet, H. (1999) Pseudo-degrees of freedom for complex predictive models: The example of partial least squares, *Journal of Chemometrics*, 13, (3-4), 195-208.

[106] Venables, W.N. and Ripley, B.D. (2002) *Modern Applied Statistics with S*. Berlin, Springer-Verlag.

[107] Esposito Vinzi, V., Chin, W.W., Henseler, J. and Wang, H. (Eds.) (2010) *Handbook of Partial Least Squares: Concepts, Methods and Applications*, Berlin, Springer-Verlag.

[108] Wehrens, R. *Chemometrics with R, Multivariate Data Analysis in the Natural Sciences and Life Sciences*. Berlin, Springer-Verlag.

[109] Wiklund,S., Nilsson,D., Eriksson,L. Sjöström,M., Wold.S. and Faber,K. (2007) A randomization test for PLS component selection, *Journal of Chemometrics*, 21, (10-11), 427-439.

[110] Wise, B.M. (Dated 2004). Properties of Partial Least Squares (PLS) Regression, and Differences between Algorithms.
Available at `http://www.eigenvector.com/Docs/Wise-pls-properties.pdf`.

[111] Wold, H. (1975) Path models with latent variables: the NIPALS approach, in: H.M. Blalock (Ed.), *Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building*. New York, Academic Press, 307-335.

[112] Wold, S. (1978) Cross-validatory estimation of the number of components. *Technometrics*, 20, (4), 397-405

[113] Wold H. (1982) Soft modeling: the basic design and some extensions. *Systems under Indirect Observation*, Part 2, Jöreskog K.G. and Wold, H. (eds). Amsterdam, North-Holland, 1-54.

[114] Wold, S. Martens, H. and Wold, H.(1983) The multivariate calibration method in chemistry solved by the PLS method. in Ruhe,A. and Kagstrom, B. (Eds.) *Proceedings on the Conference on Matrix Pencils*, Lecture Notes in Mathematics, Heidelberg, Springer-Verlag, 286-293.

[115] Wold, S. Albano, C. Dunn III, W.J., Esbensen, K., Hellberg, S., Johansson, E. and Sjöström, M. (1983) Pattern recognition: finding and using patterns in multivariate data, in: H. Martens, H. Russwurm Jr. (Eds), *Food Research and Data Analysis* London, Applied Science Publications. 147-188.

[116] Wold, S., Johansson, E. and Cocchi, M. (1993). PLS-partial least squares projections to latent structures. In *3D QSAR in Drug Design*, Kubinyi, H. (ed.). Leiden, ESCOM Science Publishers, 523-548.

[117] Wold, S. Antii, H., Lindgren, F. and Öhman,J. (1998) Orthogonal signal correction of near-infrared spectra *Chemometrics and Intelligent Laboratory Systems* 44, (1-2), 175-185.

[118] Wold, S. (2001) Personal memories of PLS. *Chemometrics and Intelligent Laboratory Systems*, 58, (2), 8384.

[119] Wold, S. Sjöström, M. and Eriksson,L. (2001) PLS-regression: a basic tool of chemometrics *Chemometrics and Intelligent Laboratory Systems*, 58, (2), 109-130.

[120] Wold, S., Høy, M., Martens, H., Trygg, J., Westad, F., MacGregor, J.,and Wise, B.M. (2009). The PLS model space revisited. *Journal of Chemometrics*, 23(12), 67?68.

[121] Wong, K.K-K. (2011) Book Review: Handbook of Partial Least Squares: Concepts, Methods and Applications, *International Journal of Business Science and Applied Management*, 6, (2), 53-54.

[122] Wu, W. and Manne, R. Fast regression methods in a Lanczos (or PLS1) basis. Theory and applications. (2000) *Chemometrics and Intelligent Laboratory Systems*, 51, (2), 145-161.

[123] Xu, Q-S and Liang, Y-Z. (2001) Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56, (1), 1-11.