Norwegian University
of Life Sciences

Master's Thesis 2018 60 ECTS

Faculty of Chemistry, Biotechnology and Food Science
Main supervisor: Solve Sæbø

Estimation of noise variance with dimension-reducing regression methods

Siri Nærland Skodvin

Applied Statistics
Faculty of Chemistry, Biotechnology and Food Science

Abstract

The focus of this thesis has been on investigating the performance of some estimators of the noise variance using the dimension-reducing methods PCR, PLSR and a recently developed Bayesian method, Bayes PLS, through a simulation study.

In all data modeling, there is a certain consumption of degrees of freedom due to the estimation of unknown parameters. It can be important to determine the degrees of freedom in order to assess the level of the noise variance (dependent of the choice of estimator). In this thesis, a definition of the degrees of freedom as the expected value of the trace of the first derivative of the fitted values (suggested by Krämer and Sugiyama [2011]) has been applied. For PCR this leads to the simplified or 'naive' definition that the degrees of freedom equals the number of components included in the fitted model (regression coefficients) + 1 (the intercept). In PLSR, the relationship between the response and the fitted values is non-linear, so finding an analytic expression of the derivative is quite complicated, maybe even impossible. Therefore, two alternative PLSR estimators of the noise variance has been investigated; one that uses the naive estimate of the degrees of freedom, and one that is based on a numerical approximation of the derivative of the fitted values.

Bayes PLS uses a numerical approach (MCMC) to estimate all the unknown parameters, so the noise variance estimate can be obtained without having to consider the degrees of freedom.

The results of the simulations show that the best estimators, in terms of smaller estimation error, fewer number of components included in the fitted model, and overall more stabile results, are the PLSR estimator with the naive estimate of the degrees of freedom and the Bayes PLS estimator. The simulations also show that the true value of the degrees of freedom of PLSR is probably larger than the naive estimate in some situations.

Sammendrag

I denne oppgaven har en simuleringsstudie blitt gjennomført, der de dimensjonsreduserende metodene PCR, PLSR og Bayes PLS har blitt brukt til å tilpasse lineære modeller, og til å estimere den vanligvis ukjente støyvariansen. Deretter har de forskjellige støyvarians-estimatorene blitt vurdert og sammenlignet med hverandre.

I all statistisk modellering må ukjente parametre estimeres, og til denne estimeringen brukes det et visst antall frihetsgrader. Det kan være viktig å anslå dette antallet frihetsgrader, for å kunne estimere nivået av tilfeldig støy i modellen (avhengig av valg av estimator). Frihetsgradene kan matematisk defineres som forventningen til trasen til den partiellderiverte av de tilpassede verdiene (foreslått av Krämer and Sugiyama [2011]). For PCR fører denne definisjonen til den relativt enkle eller "naive" definisjonen av frihetsgradene som antall komponenter som inkluderes i den tilpassede modellen (regresjonskoeffisienter) + 1 (konstantleddet). I en modell tilpasset ved PLSR er det et ikke-lineært forhold mellom responsen og de tilpassede verdiene, så å finne et analytisk uttrykk for den deriverte er komplisert, om ikke umulig. Derfor har to forskjellige forslag til frihetsgrader for modellen tilpasset ved PLSR blitt brukt; det naive estimatet, og en numerisk tilnærming til den deriverte av de tilpassede verdiene.

Bayes PLS bruker en numerisk metode (MCMC) til å estimere de ukjente parametrene, så støyvarians-estimatet gis uten at det er nødvendig å anslå frihetsgradene.

Resultatene av simuleringsstudien viser at de beste estimatorene, med hensyn på lavest estimeringsfeil, færrest komponenter inkludert i den tilpassede modellen, og gjennomgående mest stabile resultater, er PLSR-estimatoren med det naive estimatet av frihetsgrader, og Bayes PLS-estimatoren. Simuleringene viser også at den sanne verdien av frihetsgradene i PLSR i noen situasjoner trolig er høyere enn det naive estimatet.

Aknowledgement

This master thesis in Applied Statistics is written at the faculty of Chemistry, Biotechnology and Food Science at the Norwegian University of Life Sciences. I would like to thank my supervisor, Trygve Almøy, for his support and guidance, for always taking the time to answer my questions, and for showing a genuine interest in my work. I would also like to thank Solve Sæbø for his valuable contributions. I am grateful to Raju Rimal, Lars Snipen and Kristian Hovde Liland for their technical assistance, and to Peter Smith for proofreading the thesis. Finally, I want to thank my family, especially my beloved Fredrik Melsom, my mother Kari Skodvin, and my children, Georg and Jesper.

In memory of my father, Geir Nærland.

Contents

1 Introduction

Some dependencies between different variables in nature are quite obvious; for example the relationship between the length and weight of a person, or the relationship between the amount of accessible water and sunlight (predictor variables), and the growth of a plant (dependent variable/response). Discovering dependencies like these, that may be more or less obvious, is an underlying objective of most scientific research. Two main premises for this search may be formulated in the following way:

1. There exist some true relationships between variables that may be expressed matemathically
2. There is a natural variation of individual observations that is *not* captured by this defined relationship

This 'natural variation' is what will in this thesis be referred to as the noise variance, or random noise. It is 'noisy' because it can not be explained.

The discipline of statistics is based on trying to express these relationships as accurately as possible, by developing fitted models that serve both to explain the effect of the variables in question, and to predict the response for new values of the predictor variables. The quality of the prediction is closely linked to the size of the true noise variance. Clearly, if the noise variance is large, the prediction will also be less accurate. Of course, usually the true noise variance is unknown.

There exist several methods of estimating the unknown parameters of a statistical model, one of the more well-known is Ordinary Least Squares (OLS). The OLS estimates are quite straightforward and intuitive, but come with some major disadvantages; for high-dimensional data and/or highly correlated predictor variables, the least squares estimates may be very innacurate, or may not even be possible to calculate. Two statistical methods that deal better with such types of datasets are Principal Components Regression (PCR) and Partial Least Squares Regression (PLSR). Both methods are based on compressing the data to retrieve most of the relevant information from the predictor variables, and minimize any redundancy. One advantage of PLSR over PCR is that PLSR considers

the covariance between the predictor variables and the response, whereas PCR solely considers the covariance between the predictor variables.

Both PCR and PLSR provide estimates for most of the parameters of interest. However, both methods lack a good, uniformly accepted estimator for the noise variance. The unbiased OLS estimator of the noise variance is the sum of the squared residuals (SSE) divided by its degrees of freedom. In PCR and PLSR the degrees of freedom of SSE is a somewhat more complicated matter than it is in OLS.

A resently developed method, Bayes PLS, has its foundation in Bayesian statistics, and is also based on the concept of a relevant subspace such as PCR and PLSR. Bayes PLS uses in part a numerical approach (Markov chain Monte Carlo) to obtain estimates of the unknown parameters, including the noise variance.

Through simulation, this thesis will investigate the performance of some chosen PCR and PLSR estimators of the degrees of freedom and the noise variance. The datasets used will be simulated using the R package *simrel* (Sæbø [2014]), so the true (otherwise unknown) value of the noise variance will be known. A numeric approach (Krämer and Sugiyama [2011]) to the degrees of freedom of PLSR will be considered in the simulation study, and also the performance of PCR, PLSR and Bayes PLS respectively will be evaluated and compared.

2 Variables, models and concepts

2.1 Notation

In this thesis the following notation is used:

- Random, scalar variables are denoted by capital Latin letters, e.g. Y.
- Vectors of random variables are denoted by bold, lowercase Latin letters,

  e.g. $\boldsymbol{y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$.
- Matrices of random variables are denoted by bold, capital Latin letters, e.g. $\boldsymbol{X}$. Sometimes the dimensions of such matrices are also given, e.g. $\underset{n \times p}{\boldsymbol{X}}$, meaning that the matrix $\boldsymbol{X}$ has n rows and p columns.
- Scalars are denoted by lowercase Latin letters, e.g. k.
- Scalar parameters are denoted by lowercase Greek letters, e.g. β.
- Vectors of parameters are denoted by bold, lowercase Greek letters, e.g. $\boldsymbol{\beta}$.
- Matrices of parameters are denoted by bold, capital Greek letters, e.g. $\boldsymbol{\Sigma}$.
- Estimates of unknown parameters are denoted by a hat, e.g. $\hat{\boldsymbol{\beta}}$.
- The transpose of a vector is denoted by a t in superscript, e.g. the transpose of $\boldsymbol{y}$ is written as $\boldsymbol{y}^t$.

2.2 The linear model

Consider a response variable Y and a vector $\boldsymbol{x}$ of p predictor variables. The variables Y and $\boldsymbol{x}$ are assumed to be simultaneously normal distributed:

$$\begin{bmatrix} Y \\ \boldsymbol{x} \end{bmatrix} \sim N(\begin{bmatrix} \mu_Y \\ \boldsymbol{\mu_x} \end{bmatrix}, \begin{bmatrix} \sigma_Y^2 & \boldsymbol{\sigma}_{\boldsymbol{x}Y}^t \\ \boldsymbol{\sigma}_{\boldsymbol{x}Y} & \boldsymbol{\Sigma}_{\boldsymbol{xx}} \end{bmatrix})$$

where μ_Y is the expected value of Y, $\boldsymbol{\mu_x}$ is the $p \times 1$ vector of expected values of $\boldsymbol{x}$, σ_Y^2 is the variance of Y, $\boldsymbol{\sigma}_{\boldsymbol{x}Y}$ is the $p \times 1$ covariance vector of $\boldsymbol{x}$ and Y, and $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ is the symmetric $p \times p$ covariance matrix of $\boldsymbol{x}$.

The conditional distribution of $Y|\boldsymbol{x}$ is expressed by the equation

$$Y|\boldsymbol{x} = \beta_0 + \boldsymbol{\beta}^t\boldsymbol{x} + \epsilon \tag{1}$$

where β_0 is the intercept, $\boldsymbol{\beta}$ is the $p \times 1$ vector of coefficients and ϵ is the error term, which is assumed to be normal distributed with expected value 0 and constant variance σ^2

$$\epsilon \sim N(0, \sigma^2)$$

Note that the variance of the error terms is equivalent to what has previously been referred to as the 'noise variance'. Thus the common notation for the noise variance is σ^2.

The model in (1) is known as the general linear model.

Since Y and $\boldsymbol{x}$ are both normal distributed, $Y|\boldsymbol{x}$ is also normal distributed with expected value

$$E(Y|\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^t\boldsymbol{x}$$

The value of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} = \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{\sigma}_{\boldsymbol{x}Y} \tag{2}$$

and the intercept is

$$\beta_0 = \mu_Y - \boldsymbol{\beta}^t \boldsymbol{\mu}_{\boldsymbol{x}}$$

The variance of $Y|\boldsymbol{x}$ is

$$Var(Y|\boldsymbol{x}) = \sigma^2 = \sigma_Y^2 - \boldsymbol{\sigma}_{\boldsymbol{x}Y}^t \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1} \boldsymbol{\sigma}_{\boldsymbol{x}Y} \tag{3}$$

Now consider a number of n samples drawn at random from the population described above. The observations are stored in a $n \times 1$ response vector $\boldsymbol{y}$ and a $n \times p$ predictor matrix $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_p \end{bmatrix}$.

All variables are mean-centered:

$$\boldsymbol{y}^* = \boldsymbol{y} - \overline{Y}\mathbf{1}$$

and

$$\boldsymbol{x}_i^* = \boldsymbol{x}_i - \overline{X}_i\mathbf{1}$$

where $i = 1, 2, ..., p$, $\overline{Y}$ is the average of the n responses in vector $\boldsymbol{y}$, $\overline{X}_i$ is the average of the n observations of predictor variable $\boldsymbol{x}_i$, and $\mathbf{1}$ is a $n \times 1$ vector consisting of 1's.

For the remainder of this thesis $\boldsymbol{x}_i = \boldsymbol{x}_i^*$ and $\boldsymbol{y} = \boldsymbol{y}^*$.

The intercept β_0 is then equal to 0 due to the centering of the data.

The model in (1) can be expressed in matrix form

$$\boldsymbol{y}|\boldsymbol{X} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{4}$$

2.3 A relevant subspace

When p is large (i.e. there are many predictor variables) it is natural to question whether or not all the predictor variables are significant for

estimation and prediction, and also if there may be cases of multicollinearity. In this case it is common to assume that there is a subspace of the original p-dimensional X-space containing most of the relevant information about the response. This relevant subspace is spanned by a set of relevant components.

The true covariance matrix of $\boldsymbol{x}$ can be written as a linear combination of its eigenvalues λ_i and eigenvectors $\boldsymbol{e}_i$ (eigen-decomposition). Since $\boldsymbol{\Sigma_{xx}}$ is a square, symmetric matrix it follows from the Spectral Theorem (Lay [2006]) that $\boldsymbol{\Sigma_{xx}}$ has p real eigenvalues (λ_i), and that the eigenvectors $(\boldsymbol{e}_i)$ corresponding to different eigenvalues are orthogonal. Also, $\boldsymbol{\Sigma_{xx}}$ is a positive-definite matrix, so all its eigenvalues are positive. The eigenvalues and eigenvectors of $\boldsymbol{\Sigma_{xx}}$ satisfy the equation

$$\boldsymbol{\Sigma_{xx}}\boldsymbol{e}_i = \lambda_i \boldsymbol{e}_i$$

for $i = 1, 2, ..., p$.

$\boldsymbol{\Sigma_{xx}}$ can now be expressed in the following way

$$\boldsymbol{\Sigma_{xx}} = \sum_{i=1}^{p} \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^t = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^t$$

and correspondingly for the inverse

$$\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1} = \sum_{i=1}^{p} \frac{1}{\lambda_i} \boldsymbol{e}_i \boldsymbol{e}_i^t = \boldsymbol{E}\boldsymbol{\Lambda}^{-1}\boldsymbol{E}^t$$

where $\boldsymbol{E}$ is the matrix of the eigenvectors $\boldsymbol{e}_i$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with the corresponding λ_i's on the diagonal. The λ_i's are sorted in descending order so that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$. The eigenvectors are all unit vectors, so the length or norm of each eigenvector is 1.

The true value of the regression coefficients can now be expressed using the sum-representation of $\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}$

$$\boldsymbol{\beta} = \sum_{i=1}^{p} \frac{1}{\lambda_i} \boldsymbol{e}_i \boldsymbol{e}_i^t \boldsymbol{\sigma}_{\boldsymbol{x}Y} \tag{5}$$

The product of $\underset{1\times p}{\boldsymbol{e}_i^t}$ and $\underset{p\times 1}{\boldsymbol{\sigma}_{\boldsymbol{x}Y}}$ is a scalar, so equation (5) can be written as

$$\boldsymbol{\beta} = \sum_{i=1}^{p} \frac{\boldsymbol{e}_i^t \boldsymbol{\sigma}_{\boldsymbol{x}Y}}{\lambda_i} \boldsymbol{e}_i = \sum_{i=1}^{p} \alpha_i \boldsymbol{e}_i$$

where $\alpha_i = \frac{\boldsymbol{e}_i^t \boldsymbol{\sigma}_{\boldsymbol{x}Y}}{\lambda_i}$.

Consider a set P_m containing the positions of the true relevant components. Here m specifies the number of relevant components. (For example, if $P_m = \{1, 2, 5\}$, then the true relevant components are component 1, 2 and 5, and $m = 3$.)

Then $\alpha_i = 0$ for all $i \notin P_m$, and the true value of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} = \sum_{i \in P_m} \alpha_i \boldsymbol{e}_i$$

3 Estimation

3.1 The fitted model

Analyzing the relationship between variables is what is known as regression, and the purpose of regression is often to be able to predict future values of the response variable. The manner in which the response Y depends on the predictor variables $\boldsymbol{x}$ can best be described by trying to determine the values of the parameters in the model that is presumed to fit the variables. Since the true values of the parameters are rarely known, the analyst can only use the observed data to find an empirical approximation to the true value. This is what is known as estimation.

In this thesis the main perspective is on the linear model described in (4), but it should be mentioned that non-linear relationships between Y and $\boldsymbol{x}$ could just as well be considered.

The fitted model with estimated regression coefficients ($\hat{\boldsymbol{\beta}}$) is

$$\hat{\boldsymbol{y}}_\delta = \boldsymbol{X}\hat{\boldsymbol{\beta}}_\delta \tag{6}$$

where δ is a regularization parameter defined by the choice of regression method. Some different methods of regression and estimation will be explored further in chapter 5.

Now new values of Y can be predicted for new observed values of $\boldsymbol{x}$ using the fitted model in (6)

$$\hat{Y}_{pred} = \hat{\boldsymbol{\beta}}_\delta^t \boldsymbol{x}$$

A general expression for the estimated coefficient vector is given by (Helland and Almøy [1994])

$$\hat{\boldsymbol{\beta}}_\delta = \boldsymbol{A}_\delta(\boldsymbol{A}_\delta^t \boldsymbol{S} \boldsymbol{A}_\delta)^{-1}\boldsymbol{A}_\delta^t \boldsymbol{s} \tag{7}$$

where $\boldsymbol{S} = \frac{\boldsymbol{X}^t\boldsymbol{X}}{n-1}$ is the empirical covariance matrix of $\boldsymbol{X}$, $\boldsymbol{s} = \frac{\boldsymbol{X}^t\boldsymbol{y}}{n-1}$ is the empirical covariance matrix of $\boldsymbol{X}$ and $\boldsymbol{y}$, and $\boldsymbol{A}_\delta$ is a matrix defined by the choice of regression method.

The term $n-1$ cancels out, and the expression in (7) can be written as

$$\hat{\boldsymbol{\beta}}_\delta = \boldsymbol{A}_\delta(\boldsymbol{A}_\delta^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{A}_\delta)^{-1}\boldsymbol{A}_\delta^t\boldsymbol{X}^t\boldsymbol{y} \tag{8}$$

Now the fitted model in (6) can be expressed as a function of $\boldsymbol{y}$

$$\begin{aligned}\hat{\boldsymbol{y}}_\delta &= \boldsymbol{X}\boldsymbol{A}_\delta(\boldsymbol{A}_\delta^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{A}_\delta)^{-1}\boldsymbol{A}_\delta^t\boldsymbol{X}^t\boldsymbol{y} \\ &= \boldsymbol{H}_\delta\boldsymbol{y}\end{aligned}$$

where $\underset{n\times n}{\boldsymbol{H}_\delta}$ is known as the hat matrix defined as

$$\boldsymbol{H}_\delta = \boldsymbol{X}\boldsymbol{A}_\delta(\boldsymbol{A}_\delta^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{A}_\delta)^{-1}\boldsymbol{A}_\delta^t\boldsymbol{X}^t \tag{9}$$

3.1.1 Variance and degrees of freedom

The amount of random noise affects the predictive accuracy of a fitted model. In fact, the true value of the noise variance constitutes a theoretical lower bound for the prediction error given by

$$MSEP = E(Y - \hat{Y}_{pred})^2$$

MSEP is the abbreviation for Mean Square Error of Prediction.

The estimate of the noise variance will therefore (if it is sufficiently accurate) help to assess the fitted model's ability for prediction.

An estimator of the noise variance is

$$\hat{\sigma}_\delta^2 = \frac{SSE_\delta}{n - DoF_\delta} \tag{10}$$

SSE_δ is a measure of the total variation of the data which is not explained by the fitted model

$$SSE_\delta = (\boldsymbol{y} - \hat{\boldsymbol{y}}_\delta)^t(\boldsymbol{y} - \hat{\boldsymbol{y}}_\delta) \tag{11}$$

The term DoF_δ indicates the degrees of freedom that are consumed by the specific regression method in use.

A general definition of the degrees of freedom has been referred to by Krämer and Sugiyama [2011]

$$DoF_\delta = E\Big[trace\Big(\frac{\partial \hat{\boldsymbol{y}}_\delta}{\partial \boldsymbol{y}}\Big)\Big] \tag{12}$$

Here $\boldsymbol{X}$ is assumed given and the expectation is taken with regard to $\boldsymbol{y}$.

If $\hat{\boldsymbol{y}}_\delta$ is linearly dependent on $\boldsymbol{y}$, meaning that $\boldsymbol{H}_\delta$ is not defined in terms of $\boldsymbol{y}$, the right side of equation (12) is simplified to the trace of $\boldsymbol{H}_\delta$ (Krämer and Sugiyama [2011]).

The estimation of the intercept consumes one degree of freedom, and so does the centering of the data (in which case the intercept is equal to 0), so in any case the correct expression of the degrees of freedom is

$$DoF_\delta = 1 + trace(\boldsymbol{H}_\delta) \tag{13}$$

3.2 Evaluating estimator performance

As mentioned previously, there are usually several different statistical methods that can be used to estimate the parameters of a model, and even for a given choice of method there may be several reasonable suggestions of estimators of a parameter. Determining which estimator is the 'best' can be a challenge. One attribute of an estimator that should be examined is its bias (Devore and Berk [2007]). An estimator $\hat{\theta}$ is an unbiased estimator of the parameter θ if the expected value of $\hat{\theta}$ is equal to the true value of θ for all possible values of θ. The bias of the estimator $\hat{\theta}$ can be calculated by

$$E(\hat{\theta}) - \theta$$

To further evaluate the estimator's performance the variance of the estimator should be considered. The variance is

$$\begin{aligned} Var(\hat{\theta}) &= E(\hat{\theta} - E(\hat{\theta}))^2 \\ &= E(\hat{\theta}^2) - (E(\hat{\theta}))^2 \end{aligned}$$

If $Var(\hat{\theta})$ is large, $\hat{\theta}$ is an unstable estimator; even if it is unbiased, and on average will hit the target, the individual estimates will sometimes deviate significantly from the true value of the parameter. So the desired estimator is unbiased (if possible), and with as little variance as possible. The unbiased estimator $\hat{\theta}$ of θ with minimum variance among all unbiased estimators of θ is called the uniformly minimum variance unbiased (UMVU) estimator of θ (Bickel and Doksum [2007]).

Note that the UMVU estimator of a parameter does not have the minimum variance among *all* estimators of that parameter. And in some cases another estimator which is biased but with a smaller variance may be preferable over the UMVU estimator, because even if on average the estimator will not hit the true value of the parameter, it will provide more stable estimates that do not vary as much as the estimates provided by the UMVU estimator. This concept of accepting a larger bias in favor of a lower variance is known as a bias-variance trade-off.

3.2.1 MSE and the bias-variance decomposition

The mean square error (MSE) of an estimator $\hat{\theta}$ is defined as the mean squared difference between the estimates and the true value of the parameter θ (Devore and Berk [2007])

$$MSE(\hat{\theta}) = E(\hat{\theta} - \theta)^2 \tag{14}$$

MSE can be decomposed into the sum of the variance and the squared bias of the estimator

$$
\begin{aligned}
MSE(\hat{\theta}) &= E(\hat{\theta}^2 - 2\theta\hat{\theta} + \theta^2) \\
&= E(\hat{\theta}^2) - 2\theta E(\hat{\theta}) + \theta^2 \\
&= E(\hat{\theta}^2) - (E(\hat{\theta}))^2 + (E(\hat{\theta}))^2 - 2\theta E(\hat{\theta}) + \theta^2 \\
&= E(\hat{\theta}^2) - (E(\hat{\theta}))^2 + (E(\hat{\theta}) - \theta)^2 \\
&= Var(\hat{\theta}) + (Bias(\hat{\theta}))^2
\end{aligned}
$$

MSE therefore provides a measure of an estimator's performance that take both bias and variance of the estimator into account. The estimator with the lowest MSE is preferable.

4 Bayesian inference

Most of the ideas presented in the previous section are examples of a frequentist way of thinking. The general idea is that all estimation of unknown parameters is done based on (and only on) the observed data. Bayesians, on the other hand, believe that the model parameters should themselves be regarded as stochastic variables with a prior probability distribution (independent of the observed data). Thus the principle of Bayesian inference is to fit models by combining prior assumptions and observed data.

The foundation of Bayesian statistical modeling is Bayes' theorem for conditional probability distributions (Gilks et al. [1996])

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

where $P(A)$ is the *a priori* probability of A, and $P(A|B)$ is the *a posteriori* probability of A given B.

Following is an example to elaborate on the idea of Bayesian inference.

Consider the observed data $\boldsymbol{D} = \{X_1, X_2, ..., X_n\}$, and assume that the X_i's are independent and identically distributed

$$X_i \sim N(\mu, \sigma^2)$$

so the set of unknown parameters is given by $\boldsymbol{\theta} = \{\mu, \sigma^2\}$.

The a posteriori probability of $\boldsymbol{\theta}$ given $\boldsymbol{D}$ is then

$$P(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{P(\boldsymbol{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})}{P(\boldsymbol{D})}$$

$P(\boldsymbol{\theta})$ is the probability of $\boldsymbol{\theta}$ with no knowlegde of the observed data, and $P(\boldsymbol{D}|\boldsymbol{\theta})$ is the probability distribution of the data given the parameter

$\boldsymbol{\theta}$, which is equivalent with the likelihood function $L(\boldsymbol{\theta}|\boldsymbol{D})$. $P(\boldsymbol{D})$ is the 'probability of the data', an entity which is hard to compute and interpret, but it is reasonable to view $P(\boldsymbol{D})$ simply as a normalizing constant ensuring that the integral of $P(\boldsymbol{\theta}|\boldsymbol{D})$ evaluates to 1. Thus

$$P(\boldsymbol{\theta}|\boldsymbol{D}) \propto P(\boldsymbol{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})$$

Now consider σ^2 as given, so $\boldsymbol{\theta} = \{\mu\}$. From the above it follows that $P(\boldsymbol{D}|\boldsymbol{\theta})$ is a joint normal distribution. Assume in addition that $\boldsymbol{\theta}$ is normally distributed

$$\begin{gathered}\boldsymbol{D}|\boldsymbol{\theta} = \{X_1, X_2, ..., X_n|\mu\} \sim N_n(\mu, \sigma^2) \\ \boldsymbol{\theta} = \{\mu\} \sim N(\nu, \tau^2)\end{gathered}$$

in which case it may be shown that $\boldsymbol{\theta}|\boldsymbol{D}$ is also normally distributed. In this special case it is possible to find an analytical expression of $f(\boldsymbol{\theta}|\boldsymbol{D})$, the true probability density function (pdf) of $\boldsymbol{\theta}|\boldsymbol{D}$

$$f(\boldsymbol{\theta}|\boldsymbol{D}) \sim N(\frac{\frac{n \cdot \overline{X}}{\sigma^2} + \frac{\nu}{\tau^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau^2}}, \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\tau^2}})$$

(Lehmann and Casella [2006]).

In general, an expression of $f(\boldsymbol{\theta}|\boldsymbol{D})$ is not so easily obtained, but a function $g(\boldsymbol{\theta}|\boldsymbol{D})$ proportional to $f(\boldsymbol{\theta}|\boldsymbol{D})$ can be found by multiplying the pdf of $\boldsymbol{D}|\boldsymbol{\theta}$ with the pdf of $\boldsymbol{\theta}$

$$g(\boldsymbol{\theta}|\boldsymbol{D}) = f(\boldsymbol{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})$$

Now as it turns out $g(\boldsymbol{\theta}|\boldsymbol{D})$ is actually a kind of 'un-normalized' version of $f(\boldsymbol{\theta}|\boldsymbol{D})$, and the two distributions share the same center of mass (see figure 1). So, in situations where $f(\boldsymbol{\theta}|\boldsymbol{D})$ is not known, but $g(\boldsymbol{\theta}|\boldsymbol{D})$ is, a number of samples can be drawn from $g(\boldsymbol{\theta}|\boldsymbol{D})$ to estimate a pdf $\hat{f}(\boldsymbol{\theta}|\boldsymbol{D})$ approximating the true a posteriori pdf.

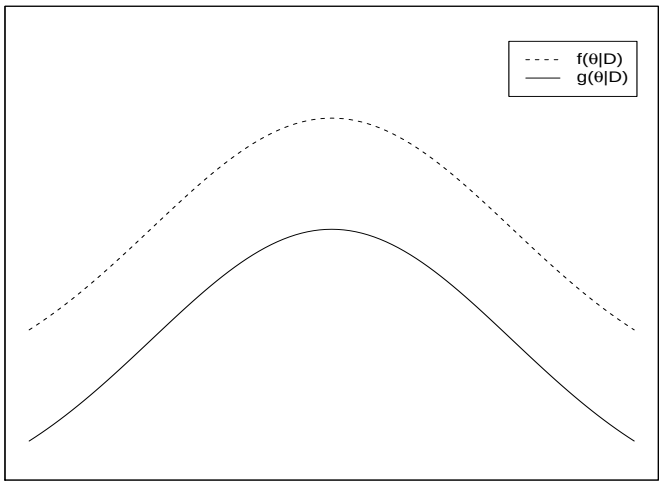


Figure 1: The distributions of f and g

4.1 Markov chain Monte Carlo (MCMC)

Monte Carlo integration is a numerical approach that uses random samples from a specific distribution to find an approximation of the true value of an integral. One way of drawing the random samples is to generate a Markov chain. This is what is known as Markov chain Monte Carlo (MCMC). MCMC can be applied in different contexts, but has proven to be especially suitable in use with Bayesian statistics. In this section a short overview of MCMC is presented. For further details see e.g. Gilks et al. [1996].

A Markov chain of n samples (n is now a number of choice for the analyst) for the Monte Carlo integration is generated. The main feature of a Markov chain is that the next element X_{t+1} in the chain depends solely on the current element X_t. The next element X_{t+1} is sampled from a distribution $P(X_{t+1}|X_t)$ which is called the *transition kernel* of the chain. Because only the current element is considered when sampling the next, the chain will gradually 'forget' its starting point X_0, and over time the distribution of the elements in the Markov chain will converge to a unique *stationary* distribution, if this exists.

To ensure that the stationary distribution, of which the distribution of the elements of the Markov chain converges to, actually *is* our desired pdf $f(\boldsymbol{\theta}|\boldsymbol{D})$, the Metropolis-Hastings algorithm is implemented when constructing the chain. The Metropolis-Hastings algorithm samples a candidate Y for the next element X_{t+1} of the chain from a proposal distribution based on the current element X_t. Then Y is accepted with some probability defined by the algorithm. If Y is accepted, X_{t+1} is set equal to Y, and if Y is not accepted X_{t+1} is set equal to X_t. The Metropolis-Hastings algorithm ensures that if X_t belongs to a certain distribution, then so does X_{t+1}. So, after a sufficient *burn-in phase*, once an element from f has been obtained in the chain, all the subsequent elements of the chain will also be from f. In a practical implementation of a Markov chain the analyst will have to decide the critical number m of initial elements of the chain that will be discarded as elements from the burn-in phase.

In order to generate the Markov chain a starting value X_0 needs to be determined by the analyst. This starting value can in principle be chosen completely at random, because the distribution of the elements of the chain will eventually converge to our distribution of interest anyway. However the more extreme the starting value, the longer the burn-in phase, and also the Metropolis-Hastings algorithm may have more difficulties leading the chain towards the desired distribution. So it may be wise to choose a well considered starting value for the chain.

5 Regression methods

5.1 Ordinary least squares

Ordinary least squares (OLS) is a well-known regression method that uses all observed information in $\boldsymbol{X}$ to estimate the model parameters. The estimated coefficient vector can be obtained by inserting an arbitrary $p \times p$ matrix in the place of $\boldsymbol{A}_\delta$ from (7). Inserting $\boldsymbol{A}_p = \boldsymbol{I}_p$ (the identity matrix) in (7) reduces the expression significantly

$$\hat{\boldsymbol{\beta}}_p = \boldsymbol{S}^{-1}\boldsymbol{s} = (\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{y}$$

Similarly as described in section 2.3, the estimator can also be expressed using a sum-representation of the estimated covariance matrix $\boldsymbol{S}$

$$\hat{\boldsymbol{\beta}}_p = \sum_{i=1}^{p} \frac{1}{\hat{\lambda}_i} \hat{\boldsymbol{e}}_i \hat{\boldsymbol{e}}_i^t \boldsymbol{s}$$

where $\hat{\lambda}_i$ and $\hat{\boldsymbol{e}}_i$ are estimates of the eigenvalues and eigenvectors of $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, and they are found by eigen-decomposition of $\boldsymbol{S}$.

The hat matrix in OLS is

$$\boldsymbol{H}_p = \boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t$$

Since $\boldsymbol{H}_p$ does not depend on $\boldsymbol{y}$, the relationship between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ is linear, and the degrees of freedom of SSE can be found by applying the simplified definition in (13)

$$DoF_p = 1 + trace(\boldsymbol{H}_p) = p + 1$$

Thus the OLS estimator of the noise variance σ^2 is

$$\hat{\sigma}_p^2 = \frac{SSE}{n-(p+1)}$$

5.1.1 Performance of the OLS estimators

The expected value and variance of the OLS estimators (when $\boldsymbol{X}$ is given) are

$$E(\hat{\boldsymbol{\beta}}_p) = \boldsymbol{\beta}$$

$$Var(\hat{\boldsymbol{\beta}}_p) = \sigma^2(\boldsymbol{X}^t\boldsymbol{X})^{-1}$$

$$E(\hat{\sigma}_p^2) = \sigma^2$$

$$Var(\hat{\sigma}_p^2) = \frac{2\sigma^4}{n-(p+1)}$$

It can be shown that the OLS estimators have minimum variance among all unbiased OLS estimators, so they are UMVU estimators (Bickel and Doksum [2007]). When $n >> p$, and there is little or no multicollinearity among the predictor variables $\boldsymbol{x}$, OLS is a common choice of method for regression. However, if $p \sim n$ or if some of the predictor variables are highly correlated the inverted matrix $(\boldsymbol{X}^t\boldsymbol{X})^{-1}$ can be very inaccurate, resulting in unstable estimates (with inflated variance). When $\boldsymbol{X}^t\boldsymbol{X}$ has less than full rank, for example when $n < p$, it is non-invertible, and the estimates can not be computed at all. In these cases it is natural to consider some choices of dimension reducing methods which are based on the concept of a relevant subspace explained in section 2.3.

5.2 Principal components regression

The main feature of principal components regression is to construct components Z_i as linear combinations of the p predictor variables using the eigenvectors of $\boldsymbol{S}$ as loading vectors (Jolliffe [2002])

$$Z_i = \hat{\boldsymbol{e}}_i^t \boldsymbol{x}$$

The principal components are pairwise orthogonal due to the orthogonality of the eigenvectors. The variances of the principal components Z_i are the corresponding eigenvalues $\hat{\lambda}_i$ of $\boldsymbol{S}$. If the decline of the eigenvalues is rapid, a smaller number k of components will account for a large amount of the variation in $\boldsymbol{X}$.

After reducing the number of variables from p original predictors to k principal components, the scores for the principal components for each observation are stored as columns in a $n \times k$ matrix $\boldsymbol{Z}$. The relationship between $\boldsymbol{Z}$ and $\boldsymbol{X}$ is

$$\boldsymbol{Z} = \boldsymbol{X}\hat{\boldsymbol{E}}_k$$

The columns of $\hat{\boldsymbol{E}}_k$ are the k first eigenvectors of $\boldsymbol{S}$.

The model fitted by PCR is

$$\hat{\boldsymbol{y}}_{PCR,k} = \boldsymbol{H}_{PCR,k}\boldsymbol{y}$$

The subscript $\{PCR, k\}$ is used to define the regularization parameter $\delta = k$, and to distinguish the PCR estimator from those of other regression methods.

The hat matrix $\boldsymbol{H}_{PCR,k}$ is defined as

$$\begin{aligned}\boldsymbol{H}_{PCR,k} &= \boldsymbol{Z}(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\boldsymbol{Z}^t \\ &= \boldsymbol{X}\hat{\boldsymbol{E}}_k(\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\boldsymbol{X}\hat{\boldsymbol{E}}_k)^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\end{aligned}$$

and by comparing to the general definition of the hat matrix in (9), it is clear that $\boldsymbol{A}_{PCR,k} = \hat{\boldsymbol{E}}_k$. The estimate of $\boldsymbol{\beta}$ is found by applying the definition in (8)

$$\hat{\boldsymbol{\beta}}_{PCR,k} = \hat{\boldsymbol{E}}_k(\hat{\boldsymbol{E}}_k^t \boldsymbol{X}^t \boldsymbol{X} \hat{\boldsymbol{E}}_k)^{-1} \hat{\boldsymbol{E}}_k^t \boldsymbol{X}^t \boldsymbol{y} \tag{15}$$

Note that if $k = p$ then $\boldsymbol{A}_{PCR,p} = \hat{\boldsymbol{E}}_p$ is a $p \times p$ matrix, and the estimator in (15) is the OLS estimator.

Also in this case, the estimator can be expressed using a sum-representation

$$\hat{\boldsymbol{\beta}}_{PCR,k} = \sum_{i=1}^{k} \frac{1}{\hat{\lambda}_i} \hat{\boldsymbol{e}}_i \hat{\boldsymbol{e}}_i^t \boldsymbol{s}$$

The sum of the squared errors (SSE) for the model fitted by PCR are defined as (in agreement with the general expression in (11))

$$SSE_{PCR,k} = (\boldsymbol{y} - \hat{\boldsymbol{y}}_{PCR,k})^t (\boldsymbol{y} - \hat{\boldsymbol{y}}_{PCR,k})$$

The hat matrix is not defined in terms of $\boldsymbol{y}$, so the relationship between $\hat{\boldsymbol{y}}_{PCR,k}$ and $\boldsymbol{y}$ is linear, and the definitions in (13) and (10) are valid. It can be shown that the trace of $\boldsymbol{H}_{PCR,k}$ is equal to k, which leads to the following PCR estimators of the degrees of freedom and the noise variance

$$DoF_{PCR,k} = 1 + tr(\boldsymbol{H}_{PCR,k}) = k + 1 \tag{16}$$

$$\hat{\sigma}^2_{PCR,k} = \frac{SSE_{PCR,k}}{n - DoF_{PCR,k}} = \frac{SSE_{PCR,k}}{n - (k+1)} \tag{17}$$

Some may argue that the DoF-estimator for the model fitted by PCR given in (16) is naive, as it only represents the number of independent parameters estimated, completely ignoring other features of the data that may also influence the consumption of DoF's. Hassani et al. [2012] argue that the search for maximal covariance also consumes DoF's, and that the eigenvector structure of the dataset affects the search process, and therefore also the DoF consumption. It is therefore interesting to study

how the estimator in (17) performs on datasets with different types of eigenvector structures.

5.2.1 Determining k, the number of components to include in the fitted model

The principal components are constructed in such a way that a low number of components can account for a large part of the variance in $\boldsymbol{X}$. However, even among the first PC's there may be components that are not significant for prediction of the response. Trying to assess whether or not a component is relevant may be a difficult task. Helland and Almøy [1994] has shown that with regard to prediction, components with large eigenvalues should be included in the fitted model, even if they are non-relevant. This finding may or may not be transferable to the estimation of unknown parameters, but in any case it seems that the safest approach (and what is also common practice in PCR) is to include all components up to a certain number. By doing this it must be accepted that k will with certainty be larger than m (the true number of relevant components) in all cases where $P_m \neq \{1, 2, ...m\}$ (and it may be larger than m in other situations too).

5.2.2 Performance of the PCR estimators

The bias and variance of the PCR estimator of the regression coefficient vector $\boldsymbol{\beta}$ (when $\boldsymbol{X}$ is given) are

$$E(\hat{\boldsymbol{\beta}}_{PCR,k}) - \boldsymbol{\beta} = -\sum_{i>k}^{p} \hat{\boldsymbol{e}}_i \hat{\boldsymbol{e}}_i^t \boldsymbol{\beta}$$

$$Var(\hat{\boldsymbol{\beta}}_{PCR,k}) = \frac{\sigma^2}{n-1}(\sum_{i=1}^{k} \frac{1}{\hat{\lambda}_i} \hat{\boldsymbol{e}}_i \hat{\boldsymbol{e}}_i^t)$$

(see Appendix A.1.1 for proofs).

When k increases, the absolute value of the bias decreases, and when $k = p$, the bias is 0, since $\hat{\boldsymbol{\beta}}_{PCR,p}$ is the OLS estimator. However, as k increases, the variance of the estimator will also increase, especially when

components with small estimated eigenvalues are included in the fitted model, so the estimates will be increasingly unstable for higher choices of k. So in this case choosing the optimal number of components k is an example of a bias-variance trade-off (as mentioned in section 3.2).

The bias and variance of the PCR estimator of the noise variance σ^2 are (for a given $\boldsymbol{X}$)

$$E(\hat{\sigma}^2_{PCR,k}) - \sigma^2 = \frac{(n-1)\sum_{i>k}^{p} \hat{\lambda}_i (\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2}{n-(k+1)}.$$

$$Var(\hat{\sigma}^2_{PCR,k}) = \frac{2\sigma^4}{n-(k+1)} + \frac{4\sigma^2(n-1)\sum_{i>k}^{p} \hat{\lambda}_i (\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2)}{(n-(k+1))^2}$$

(see Appendix A.1.2 and A.1.3 for proofs.)

Large estimated eigenvalues ($\hat{\lambda}_i$) of components that are not included in the estimate will have a significant contribution to the bias of the estimate. What this means in practice is that if the choice of number of components (k) leads to excluding components with high estimated eigenvalues, the estimate may be significantly biased, which is in concordance with the findings of Helland and Almøy [1994].

If n is large, and if all the relevant components are included in the estimate, the term $(\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2 = (\hat{\boldsymbol{e}}_i^t \sum_{j \in P_m} \frac{1}{\lambda_j} \boldsymbol{e}_j \boldsymbol{e}_j^t)^2$ will converge to 0 when $i > k$, because $\hat{\boldsymbol{e}}_i$ will converge in probability to $\boldsymbol{e}_i$ and $\boldsymbol{e}_i^t \boldsymbol{e}_j = 0$ when $i \neq j$, because the eigenvectors are pairwise orthogonal. If, however, all the relevant components are *not* included in the estimate, then there will be convergence in probability that some estimated eigenvector $\hat{\boldsymbol{e}}_i$ will be equal to one of the eigenvectors $\boldsymbol{e}_j$ included in $\boldsymbol{\beta}$. In this case the term $(\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2$ will not converge to 0, since $\boldsymbol{e}_i^t \boldsymbol{e}_j \neq 0$ if $i = j$.

Because the numerator and the denominator in the expression of the bias both depend on n and k, it is difficult to evaluate analytically how the bias will be affected by the size of k relative to n. But also in this case, when $k = p$ the estimator is equivalent to the OLS estimator, and the bias will then be equal to 0. So as k increases, it seems reasonable to assume that the bias will decrease. However, as k increases, the variance of the estimator will also increase (this is evident by examining the first summand of the expression of the variance).

So to summarize, the size of the bias and the variance of the PCR estimator of the noise variance seems inextricably connected to the crucial choice of k. The number of components included in the fitted model should not be to few or too many, and neither true relevant components or non-relevant components with large estimated eigenvalues should be excluded from the fitted model.

5.3 Partial least squares regression

Partial least squares regression (PLS or PLSR) is a method that has some common features with PCR. The objective of PLSR is also to reduce the dimension of the data by compressing the relevant information in the p predictor variables into a lower number k of components (Wold et al. [1983]). But in contrast to PCR, PLSR also considers the covariance between the response variable and the predictor variables, and projects both Y and $\boldsymbol{x}$ to a latent subspace. As a result the first PLSR components will usually be of some significance for predicting Y (whereas in PCR, even among the first components there may be no significant information for prediction).

There are several different algorithms developed to compute the loadings for the PLS components and the scores for regression. Many of these algorithms are iterative, i.e. the computations will be done stepwise in a loop until an initial condition fails. The orthogonalized PLSR algorithm (Martens and Næs [1989]) is presented in Appendix B.1 as an example.

The model fitted by PLSR can also, like the models fitted by OLS and PCR, be written as a function of $\boldsymbol{y}$ with an appropriate hat matrix

$$\hat{\boldsymbol{y}}_{PLSR,k} = \boldsymbol{H}_{PLSR,k}\boldsymbol{y}$$

where the hat matrix $\boldsymbol{H}_{PLSR,k}$ is defined in terms of $\boldsymbol{A}_{PLSR,k}$ (introduced in (7)). Helland [1990] showed that $\boldsymbol{A}_{PLSR,k}$ can be defined as

$$\boldsymbol{A}_{PLSR,k} = [\boldsymbol{S}^0\boldsymbol{s} \quad \boldsymbol{S}^1 s \quad \dots \quad \boldsymbol{S}^{k-1}\boldsymbol{s}] \tag{18}$$

The components included in the model fitted by PLSR are component number $1, 2, ..., k$. If $k = p$ then $\boldsymbol{A}_{PLSR,p}$ is a $p \times p$ matrix, giving the OLS estimator.

Consider a model fitted by PLSR with only 1 component ($k = 1$). Then, following the definition in (18)

$$\boldsymbol{A}_{PLSR,1} = [\boldsymbol{S}^0\boldsymbol{s}] = [\boldsymbol{s}] = \frac{1}{n-1}[\boldsymbol{X}^t\boldsymbol{y}]$$

and inserting into (9) gives

$$\boldsymbol{H}_{PLSR,1} = \boldsymbol{X}\boldsymbol{X}^t\boldsymbol{y}(\boldsymbol{y}^t\boldsymbol{X}\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{X}^t\boldsymbol{y})^{-1}\boldsymbol{y}^t\boldsymbol{X}\boldsymbol{X}^t$$

(the term $n - 1$ cancels out.)

The model fitted by PLSR with 1 component can then be written as

$$\hat{\boldsymbol{y}}_{PLSR,1} = \boldsymbol{X}\boldsymbol{X}^t\boldsymbol{y}(\boldsymbol{y}^t\boldsymbol{X}\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{X}^t\boldsymbol{y})^{-1}\boldsymbol{y}^t\boldsymbol{X}\boldsymbol{X}^t\boldsymbol{y}$$

Clearly, the relationship between $\hat{\boldsymbol{y}}_{PLSR,1}$ and $\boldsymbol{y}$ is not linear, so the straight-forward definition in (13) cannot be applied. The general definition in (12) may still be valid, but computing the derivative of $\hat{\boldsymbol{y}}_{PLSR,1}$ is difficult (maybe even impossible), and it gets even more complex if more components are included in the fitted model.

For the simulation experiments of this thesis, two alternative approaches to the degrees of freedom of the model fitted by PLSR has been used. The first approach is to let $DoF_{PLSR,k} = k + 1$, thus leading to a naive noise variance estimator similar to the PCR estimator in (17)

$$\hat{\sigma}^2_{PLSR,k} = \frac{SSE_{PLSR,k}}{n - (k+1)}$$

$SSE_{PLSR,k}$ is defined as

$$SSE_{PLSR,k} = (\boldsymbol{y} - \hat{\boldsymbol{y}}_{PLSR,k})^t(\boldsymbol{y} - \hat{\boldsymbol{y}}_{PLSR,k})$$

The second approach is a numerical approximation of the degrees of freedom of PLSR suggested by Krämer and Sugiyama [2011]. They propose two equivalent numerical methods to compute the derivative: one using

a Lanczos matrix decomposition, and one which is based on Krylov subspace techniques. The latter method has a more favorable runtime as it computes the DoF directly, whereas the Lanczos representation requires several iterations of matrix-matrix-multiplications. However, from the Lanczos decomposition algorithm derivatives of the regression coefficients are also obtained, so both algorithms are implemented in the R package *plsdof* (Krämer and Braun [2014]). This package also provides estimates of the noise variance using the naive approach (described previously and denoted by $\hat{\sigma}^2_{PLSR,k}$), and both Lanczos and Krylov representation. For PLSR, the simulations done in this thesis are limited to only consider the naive estimator and the Krylov estimator of the noise variance.

It should be noted that Krämer and Sugiyama [2011] mention some numerical problems with the R package that sometimes lead to implausible results, i.e. negative degrees of freedom.

5.4 Bayes PLS

Bayes PLS (Helland et al. [2012]) is a recently developed regression method that is also (like PCR and PLSR) based on the concept of a relevant subspace. In Bayes PLS the unknown parameters are estimated using a prior probability distribution (Bayesian inference), and a numerical approach, Markov chain Monte Carlo.

As shown in section 2.3 the true value of $\boldsymbol{\beta}$ can be written as

$$\boldsymbol{\beta} = \sum_{i \in P_m} \alpha_i \boldsymbol{e}_i$$

where P_m is the set of positions of the relevant components, $\boldsymbol{e}_i$ is the i$^{\text{th}}$ eigenvector of $\boldsymbol{\Sigma_{xx}}$, and α_i is a scalar given by

$$\alpha_i = \frac{\boldsymbol{e}_i^t \boldsymbol{\sigma}_{\boldsymbol{x}Y}}{\lambda_i}$$

Here λ_i is the eigenvalue corresponding to $\boldsymbol{e}_i$.

The parameter vector $\boldsymbol{\theta}$ (as described in section 4) now consists of the following unknown parameters (Helland et al. [2012])

$$\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\lambda}, \sigma^2, \boldsymbol{E}\}$$

where $\boldsymbol{E}$ is a matrix in which the columns are the eigenvectors $\boldsymbol{e}_i$ of $\boldsymbol{\Sigma_{xx}}$.

The parameter $\boldsymbol{\alpha}$ is a priori assumed normally distributed, $\boldsymbol{\lambda}$ and σ^2 are assumed inverse gamma distributed, and $\boldsymbol{e}_i$ has a flat (uniform) distribution on the unit sphere.

The joint a priori pdf of $\boldsymbol{\theta}$ is assumed to be

$$f(\boldsymbol{\theta}) = f(\boldsymbol{\alpha}) \cdot f(\boldsymbol{\lambda}) \cdot f(\sigma^2) \cdot f(\boldsymbol{E})$$

and the a posteriori distribution to be estimated can then be written as

$$f(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) \propto f(\boldsymbol{y}, \boldsymbol{X}|\boldsymbol{\alpha}, \boldsymbol{\lambda}, \sigma^2, \boldsymbol{E}) \cdot f(\boldsymbol{\alpha}) \cdot f(\boldsymbol{\lambda}) \cdot f(\sigma^2) \cdot f(\boldsymbol{E})$$

Following is the algorithm used to generate a Markov chain of sampled values of $\boldsymbol{\theta}$:

1. An initial run of either PCR or PLSR is performed to obtain starting values for the chain.

2. Each of the four parameters of $\boldsymbol{\theta}$ are sampled one at a time, while the others are held fixed. The new values of the parameters are sampled from a suitable proposal distribution.

3. The four new samples together form a candidate element $\hat{\boldsymbol{\theta}}_{t+1}$, which may or may not be accepted as a sample from the true distribution $f(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X})$. The probability of acceptance is defined in the Metropolis-Hastings algorithm, and depends on the previous element of the chain.

Step 2 and 3 are repeated a given number of times decided by the analyst. To obtain the desired random pattern the chain can be thinned by only saving for example every 10^{th} element. The number of burn-in phase

elements must also be decided. After the burn-in phase there is reasonable certainty that most of the elements in the chain actually belong to the distribution $f(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X})$. The estimated expected value of the a posteriori distribution will then serve as the posterior estimator of $\boldsymbol{\theta}$. It can be shown (Helland et al. [2012]) that this estimator will minimize the uniform loss if the loss function is the expected squared error, $E(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^2$.

The analyst can either decide on a fixed number of components to include in the fitted model, or choose to test for significance of the components continously during the sampling, discarding any components that are found to be not significant.

The algorithm above is implemented in the R package *BayesPLS* (Sæbø [2016,]). The R-code for Bayes PLS is available at `http://www.github.com/solvsa/BayesPLS`. This package provides estimates of all the unknown parameters of the model, including estimates of σ^2, which are used in this thesis.

6 Analysis of variance

Analysis of variance (ANOVA) is a widely used method of comparing two or more groups of observations (Devore and Berk [2007]). The general idea is to try to determine whether the group means are equal by comparing the variance between groups with the variance within groups.

Note that some of the notation used in this chapter may coincide with notation used in previous chapters. This is to abide by the nomenclature more commonly used in connection with ANOVA. Some symbols and letters may not have the same interpretation as earlier, and should not get mixed up. The interpretation of the specific notation should either be clear from the context, or explained explicitly.

6.1 One-way ANOVA

In a one-way ANOVA, the observations are denoted by Y_{ij} where i is the group number ($i = 1, 2, ..., a$) and j is the observation number ($j = 1, 2, ..., n$). Assume that the number of observations are the same for all groups, so the total number of observations is $N = a \cdot n$. The model is

$$Y_{ij} = \mu + \alpha_i + \epsilon_{ij} \tag{19}$$

where μ is the overall expected value for all observations, α_i is the effect of group i, and ϵ_{ij} are the error terms. The error terms are assumed normally distributed with expected value 0 and constant variance τ^2, i.e. $\epsilon_{ij} \sim N(0, \tau^2)$.

The total variation of the data (also known as the total sum of squares, abbreviated by SST) is the sum of the squared differences between the observations and the overall mean (denoted by $\overline{Y}..$)

$$SST = \sum_{i=1}^{a} \sum_{j=1}^{n} (Y_{ij} - \overline{Y}..)^2$$

SST can be partitioned into SSG (group sum of squares) and SSE (error sum of squares)

$$SST = SSG + SSE$$

where

$$SSG = \sum_{i=1}^{a} \sum_{j=1}^{n} (\overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot})^2$$
$$SSE = \sum_{i=1}^{a} \sum_{j=1}^{n} (Y_{ij} - \overline{Y}_{i\cdot})^2$$

$\overline{Y}_{i\cdot}$ is the mean of group i.

The mean sums of squares are found by dividing the sums of squares with their respective degrees of freedom

$$MSG = \frac{SSG}{a-1}$$

$$MSE = \frac{SSE}{N-1}$$

MSG is an estimate of the variance between groups, and MSE is an estimate of the variance within groups.

To test for difference of group means, the following hypotheses are formulated:

$H_0 : \mu_1 = \mu_2 = ... = \mu_a$

vs.

H_1 : at least two μ_i's are different

The test statistic F is defined as

$$F = \frac{MSG}{MSE}$$

When H_0 is true, the ratio MSG/MSE should be close to 1, and F is then Fisher distributed with $(a-1)$ and $(N-1)$ degrees of freedom. Using a table, the critical value $F_{\alpha,a-1,N-1}$ can be obtained, and H_0 is rejected if $F > F_{\alpha,a-1,N-1}$. Here, α is the significance level decided by the analyst, which defines the acceptable probability of wrongly rejecting H_0.

6.2 Factorial design

The ANOVA model described above can be expanded to consider several factors, each with two or more levels (Devore and Berk [2007]). If all possible combinations of levels across all factors are tested, it is called a full factorial design (also known as a completely crossed design). It is also possible to do a fractional factorial design, where only a fraction of the possible combinations are tested, but if it is not too resource intensive, it is often favorable to do a full factorial design.

The model is similar to the model in (19), expanded with several factor effects, e.g. α_i, β_j (and so on), interpreted as the effect of a given factor A at level i, the effect of factor B at level j, etc. If the effect of one factor varies dependent on the level of another factor, there is an interaction effect of the two factors. A second order interaction effect will be symbolized in the model by, for example, the term $(\alpha\beta)_{ij}$, interpreted as the effect of two given factors A and B at the respective levels i and j. Here, $(\alpha\beta)$ is merely a way of notation, and does not symbolize a product. The highest possible order of interaction effects included in a model is the same as the total number of main factors included in the model.

Several types of parametrization of the factor effect estimates are possible. One example is the sum-to-zero-parametrization. By choosing this parametrization, a restriction is imposed that the sum of the factor effect estimates over all possible levels of the factor is 0

$$\sum_i \alpha_{i=1}^{a} = 0$$

where a is the number of levels of the factor.

Another possibility is the reference level parametrization, where the effect of each factor (and interaction) at the first level is 0.

The total sum of squares (SST) is calculated in a similar way as in the one-way ANOVA, i.e. as the sum of the squared differences between the observations and the overall mean. SST can be partitioned into the sums of squares corresponding to each of the main effects and interactions effects, and the error sum of squares (SSE). When testing for significant effects, it is common to start with the highest order interaction effect. The test statistic F is defined as the mean sum of squares of the current interaction effect (or main effect) being tested, divided by the mean sum of squares of the error. If the test statistic exceeds the corresponding table value of $F_{\alpha,df1,df2}$, the particular interaction effect (or main effect) should be kept in the fitted model.

If an interaction effect is considered to be significant, then all lower order interaction effects and main effects that are a part of that specific interaction effect should also be included in the fitted model, even if they are themselves *not* significant. All other lower order interaction effects should be tested as described above. Then, a reduced model with only significant effects (and included lower order interactions) can be fitted.

6.2.1 Fixed, random and nested factors

In the situations described above, the factors of the ANOVA models are *fixed*, meaning that the factor has specific choices of levels that are not selected at random, and/or it is not reasonable to claim that these levels are representative for all possible levels of the factor (Giesbrecht and Gumpertz [2004]). Hypothesis tests on fixed effects are performed as described previously. The results of these hypothesis tests are only valid for the actual levels being investigated.

If, on the other hand, the levels of a factor can be considered as randomly selected from a larger population of possible levels, the factor is said to be *random*. By conducting hypothesis tests on random effects, the objective is to make inferences that are valid for all possible levels of the factor, not only the specific levels included in the fitted model.

A random effect ϕ at a given level g is assumed to be normal distributed with expected value 0 and constant variance

$$\phi_g \sim N(0, \tau_\phi^2)$$

The variance of the error terms, τ^2, and the variance of the random effect, τ_ϕ^2, are now referred to as *variance components*. The sum of the variance components is a measure of the total variation across all replicates and all levels of the random factor(s). Testing for significance of the random effect ϕ is equivalent to testing if $\tau_\phi^2 > 0$.

A model that consists of both fixed and random effects is known as a *mixed model*. Sometimes the levels of one factor may depend of the levels of another factor, a concept known as *nesting*. If the levels of a factor B depends on the levels of a factor A, the factor B is said to be nested within factor A.

7 Simulation

The response and predictor data used in this thesis are simulated by the R package *simrel* (Sæbø [2014]). The data simulation is based on a multivariate normal distribution and on the concept of a relevant subspace. Some of the key properties of the data are entered as input arguments in *simrel* (some of these properties are the usually unknown parameters of the true model). The data are then drawn at random under constraints set by the true parameters. The output provided by *simrel* includes the simulated response vector $\boldsymbol{y}$ and predictor matrix $\boldsymbol{X}$ (the training data), as well as some of the true parameters such as the regression coefficients. The value of the regression coefficients will disclose which of the predictor variables are truly significant, since all the non-significant predictor variables will have a coefficient equal to 0. An optional set of test data is also available as a part of the output.

In the following section, the basic concepts of *simrel* are described. For further reading, see Sæbø et al. [2015].

7.1 A description of *simrel*

Below is a description of some of the input arguments of *simrel*:

- n: The number of samples.
- p: The number of predictor variables.
- m: The number of relevant components.
- *relpos*: A vector containing the positions of the relevant components.
- γ: A parameter of the exponential decline in the eigenvalues.
- *R2*: The R^2 of the true model (the proportion of variance explained by the model).

The predictor variables span a p-dimensional space, and as discussed in section 2.3, most of the relevant information of the predictor variables can

be compressed into a m-dimensional subspace spanned by the eigenvectors of the covariance matrix of the predictor variables. The principle of *simrel* is similar to that of PCR, only in reverse; first the true eigenvalues are generated by a formula defined in such a way that the first eigenvalue is equal to 1, and then the succeding eigenvalues decline in a rate defined by the *simrel*-argument γ (when γ increases, the eigenvalues decline more rapidly). Then the covariance matrix of the response Y and the components $\boldsymbol{z}$ is constructed, the Y's and $\boldsymbol{z}$'s are sampled, and then an orthonormal rotation matrix $\boldsymbol{R}$ is constructed. $\boldsymbol{R}$ rotates the data from the Z-space to the X-space by $\boldsymbol{x} = \boldsymbol{R}^t\boldsymbol{z}$. Since $\boldsymbol{R}$ is orthonormal it follows that

$$\boldsymbol{R}^t\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^t = \boldsymbol{I}_p$$

so the same matrix $\boldsymbol{R}$ can also be used to rotate the data from the X-space and back to the Z-space. In other words $\boldsymbol{R}$ is in fact equivalent to the true matrix of eigenvectors $\boldsymbol{E}$ of $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ presented in section 2.3.

The covariance matrix of $\boldsymbol{x}$ and the covariance vector of $\boldsymbol{x}$ and Y can then be found by

$$\boldsymbol{\Sigma}_{\boldsymbol{xx}} = \boldsymbol{R}^t\boldsymbol{\Sigma}_{\boldsymbol{zz}}\boldsymbol{R}$$

$$\boldsymbol{\sigma}_{\boldsymbol{x}Y} = \boldsymbol{R}^t\boldsymbol{\sigma}_{\boldsymbol{z}Y}$$

Thus all parameters needed to calculate the true coefficient vector $\boldsymbol{\beta}$ (as defined in (2)) are given.

The true value of the noise variance σ^2 (defined in (3)) can be expressed as

$$\begin{aligned}\sigma^2 &= \sigma_Y^2(1 - \frac{\boldsymbol{\sigma}_{\boldsymbol{x}Y}^t\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{\sigma}_{\boldsymbol{x}Y}}{\sigma_Y^2}) \\ &= \sigma_Y^2(1 - R^2)\end{aligned}$$

In *simrel* σ_Y^2 is set to be equal to 1, so the true value of σ^2 is $1 - R^2$. Since R^2 is one of the input arguments of *simrel*, the true value of σ^2 can easily be defined by the analyst.

7.2 The design of the experiment

Because all the true parameters of the simulated data are known when using *simrel*, a number of interesting features of estimation and prediction can be studied as functions of these parameters. Since the philosophy of *simrel* is so similar to that of PCR, it is for example possible to use PCR to try to determine the number and positions of the relevant components, and compare with the number and positions of the true, relevant components as specified in the *simrel*-arguments m and $relpos$. Also, since *simrel* provides the true values of $\boldsymbol{\beta}$ and σ^2, estimates of these parameters can be compared with the true values, serving both to assess the estimation ability of one specific method, and to compare different methods to each other.

The parameters of the simulated data can be varied, and then the effect of the variation can be assessed by studying the quality of the estimation and/or prediction for one or several specific methods. For example it is well known that the performance of OLS is affected by the size of n relative to p. It may be of interest to study how big an impact a change in n may have on other methods as well. There is also reason to believe that there may be interaction effects of some of the parameters of the data, such as the position of the relevant components and how rapid the eigenvalues decline (specified in the *simrel*-argument γ).

When planning the experiment, it was decided to use two different levels of some of the main *simrel*-arguments, in order to generate some combinations of true parameters of the datasets believed to be interesting to study. The levels of the *simrel*-arguments used (both the ones that are varied and the ones that are held fixed) are presented in table 1.

Table 1: The values of the *simrel*-arguments.

n	**p**	**m**	**relpos**	$\boldsymbol{\gamma}$	$\boldsymbol{R^2}$
50	25	3	{1, 2, 3}	0.9	0.7
15			{3, 5, 7}	0.2	0.2

As table 1 shows, some of the *simrel*-arguments are chosen to be held fixed, because otherwise the amount of data would simply be too large to handle for this thesis. Four of the *simrel*-arguments are varied between two different levels, so there is a total number of 16 different combinations of *simrel*-arguments. For the remainder of this thesis these combinations

will be referred to as *design points* (dp). For example, the first design point, dp1, has the following *simrel*-arguments:

$$n = 50, p = 25, m = 3, relpos = \{1, 2, 3\}, \gamma = 0.9, R^2 = 0.7$$

(see Appendix C.1 for an overview of the *simrel*-arguments of all the design points.)

By keeping the number of predictor variables p fixed, and then choosing the two levels of n to be one that is smaller than p and one that is larger, the two important scenarios of $n > p$ and $n < p$ are included in the study. The argument m is inextricably connected to $relpos$, as m defines the number of relevant components, which obviously must equal the length of $relpos$. In this study only situations with 3 true, relevant components are considered, so even though $relpos$ has two different levels, m stays fixed at 3.

The levels of $relpos$ are chosen so that there is one type of dataset where the relevant components are all in sequence, and starting with component 1 ($relpos = \{1, 2, 3\}$, the low level), and another type of dataset where the opposite is the case; the relevant components are not in sequence, and the first relevant component is not component 1 ($relpos = \{3, 5, 7\}$, the high level). Both γ and R^2 are set at one high level (0.9 and 0.7, respectively) and one low level (both 0.2). Thus the true value of σ^2 also has two levels (given by $1 - R^2$): 0.3 and 0.8.

The focus of this simulation study is on the estimation of the noise variance, σ^2, by using the three dimension-reducing methods PCR, PLSR and Bayes PLS. Both the naive PLSR-estimator and the Krylov-estimator (of PLSR) will be used, so the study will consider a total of four estimators. The number of components (k) included in the fitted model varies from 1 to 8.

For each choice of k for all 16 design points, 7 different seeds are used when simulating (so that all the data in this thesis are reproducable). Also, there is a number of 3 replicated datasets for each of the 7 seeds. This is done by setting *simrel* to draw $n \cdot 3$ observations instead of just n, and then subsetting the simulated data matrices into 3 datasets with n observations each. This results in a total number of $N = 21$ comparable datasets for each choice of k for each design point.

7.2.1 Estimation error

The estimation error of a single estimate is measured by

$$\omega = (\hat{\sigma}^2 - \sigma^2)^2 \tag{20}$$

The average of the estimation errors then serve as an estimate of the mean square error defined in (14)

$$\widehat{MSE} = \frac{1}{N} \sum_{i=1}^{N} \omega_i$$

In some cases it may be preferable with a measure of estimation error that is on the same scale as the estimates themselves, such as the root mean square error

$$\widehat{RMSE} = \sqrt{\widehat{MSE}}$$

For the remainder of this thesis, the hat operator of the estimates described above will be skipped for the sake of readability.

8 Results

8.1 Results of the main simulation: estimation of σ^2

The results of the main simulation are presented in this section. The RMSE of the 21 σ^2-estimates are plotted against k (the number of components), and there is one plot for each design point. The plots are displayed in groups of four in one figure, and all four plots in the same figure belong to design points with the same number of observations (n) and the same positions of the true relevant components *relpos*. The two top plots of each figure both display a high level of R^2, and the two bottom plots both display a low level of R^2. Similarly, the two left plots of each figure both display a high level of γ, and the two right plots both display a low level of γ.

In addition to the RMSE-plots, the averages of the σ^2-estimates have also been plotted against k. These plots can be found in Appendix D.1.

Several numerical issues were encountered with the PLSRkrylov estimates obtained from the R package *plsdof* (Krämer and Braun, 2014). One issue was that out of a total of 2688 Krylov estimates of degrees of freedom there were 144 occurences of negative DoF's. All the PLSRkrylov σ^2-estimates with negative DoF-estimates have been removed from the experiment.

Another problem encountered was that some of the DoF-estimates reached the upper bound for the DoF defined by the analyst when using *plsdof*. This upper bound was set to $min(n-1, p-1)$. The reason for choosing this particular upper bound stems from the intuitive belief that the DoF of PLSR should not exceed the DoF of OLS (N. Krämer, personal communication, February 28, 2018). A total of 103 incidents of such upper bound DoF's were found, and several of them significantly deviated from the comparable DoF-estimates, resulting in some implausible values for the corresponding σ^2-estimates. Since these upper bound DoF's are so easily identifiable, their corresponding σ^2-estimates have also been removed from the experiment. In total, approximately 9 % of the estimates had to be removed due to either negative or upper bound DoF's.

Because of this, the actual number of PLSRkrylov estimates that go into the RMSE estimates illustrated in the plots vary from 14 to 21. For an overview of the number of occurences of negative and upper bound DoF's, see Appendix C.2.

The Bayes PLS function of the R package *BayesPLS* is at its present state both time consuming and quite difficult to use. The function requires that the user defines several parameters that are not easily interpretable for someone not familiar with the theory of Bayesian inference and MCMC. The user may also find it difficult to obtain the desired convergence of the Markov chain, which can be monitored by plots supplied by the function. Sometimes small adjustments in the input arguments have to be made for the chain to converge, and so it may be necessary to run the function several times to obtain reliable estimates of the parameters of the model. There are also certain types of datasets that will cause the function to break before completing the estimation. In this simulation, experience has been that the combination of a large value of γ with a large number of components to be included in the fitted model may sometimes cause a computational error of the function. However, by adjusting the input arguments of the function, all the desired Bayes PLS estimates of σ^2 were successfully obtained.

8.1.1 RMSE of the estimates

For almost all combinations of n, *relpos*, γ and R^2, the PCR estimator needs more components than all of the other three estimators to obtain the smallest possible RMSE. Comparing the RMSE's of the PLSRnaive, PLSRkrylov and Bayes PLS estimates, it looks like they behave quite similarly overall, especially when γ and/or R^2 are large. When R^2 is small the RMSE of the PCR estimator also behaves somewhat similarly to that of the other three estimators, needing fewer components included in the fitted model than when R^2 is large (but still more components than the other three estimators need). As shown previously, σ^2 and R^2 are closely connected. When σ^2 is large R^2 is small, and conversely. Therefore, it is no surprise that all four estimators have a smaller minimum RMSE when R^2 is large than when it is small.

The staircase shape of the PCR estimates in the two left plots of figure 3 indicates that the non-relevant components 2, 4 and 6 on average do not provide much useful information to improve the estimates. It seems that one can actually distinguish the relevant components from the non-

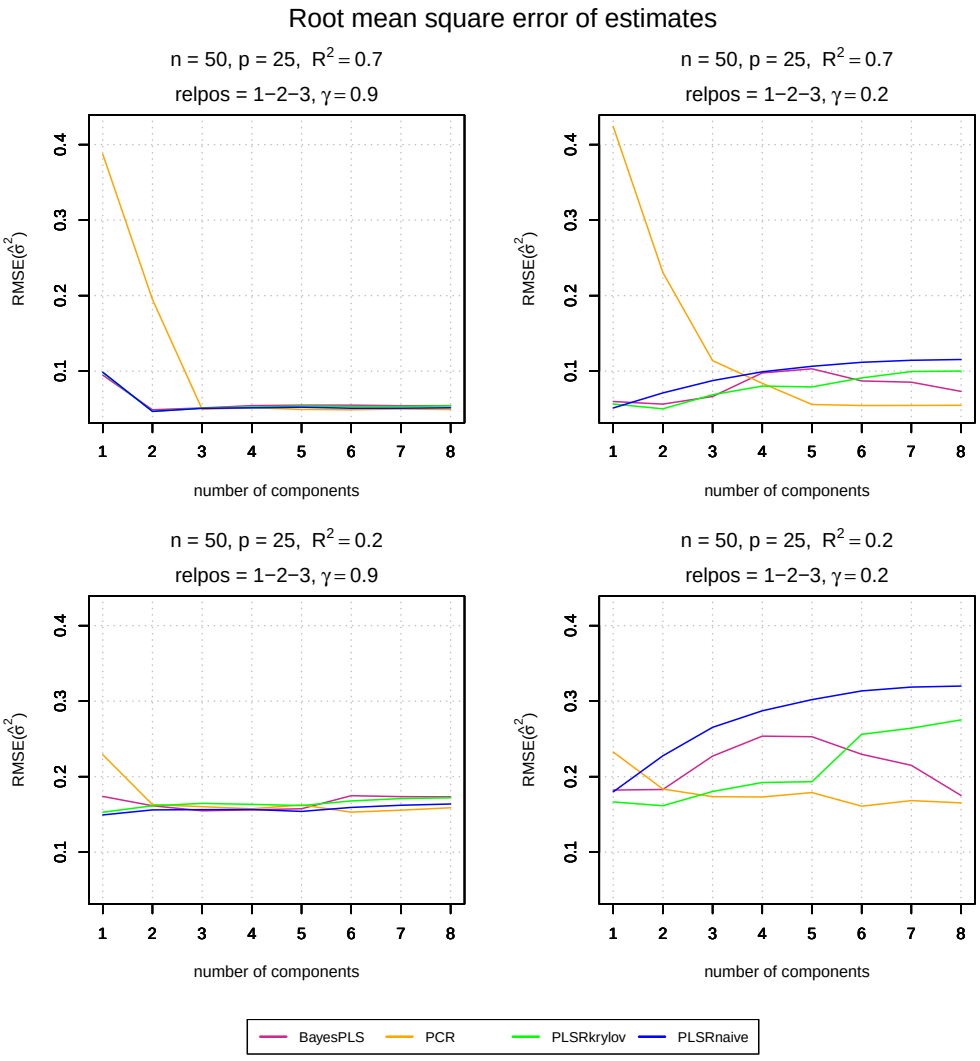


Figure 2: Root mean square error of the estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The plots belong to dp1, dp2, dp9 and dp10, all having $n = 50$, $p = 25$ and $relpos = \{1, 2, 3\}$.

relevant components directly from this plot. In the plots to the right of figure 3 the characteristic staircase shape of the PCR estimates is no longer present. Here γ is small, meaning that the non-relevant components 2, 4 and 6 have higher true eigenvalues than when γ is large.

When γ is large and R^2 is large, the PLSRnaive, PLSRkrylov and Bayes PLS estimator all need more than 1 component included in the fitted model to obtain their respective minimum RMSE's. In all other situations, the RMSE's of all three estimators seem reasonably small for $k = 1$. In

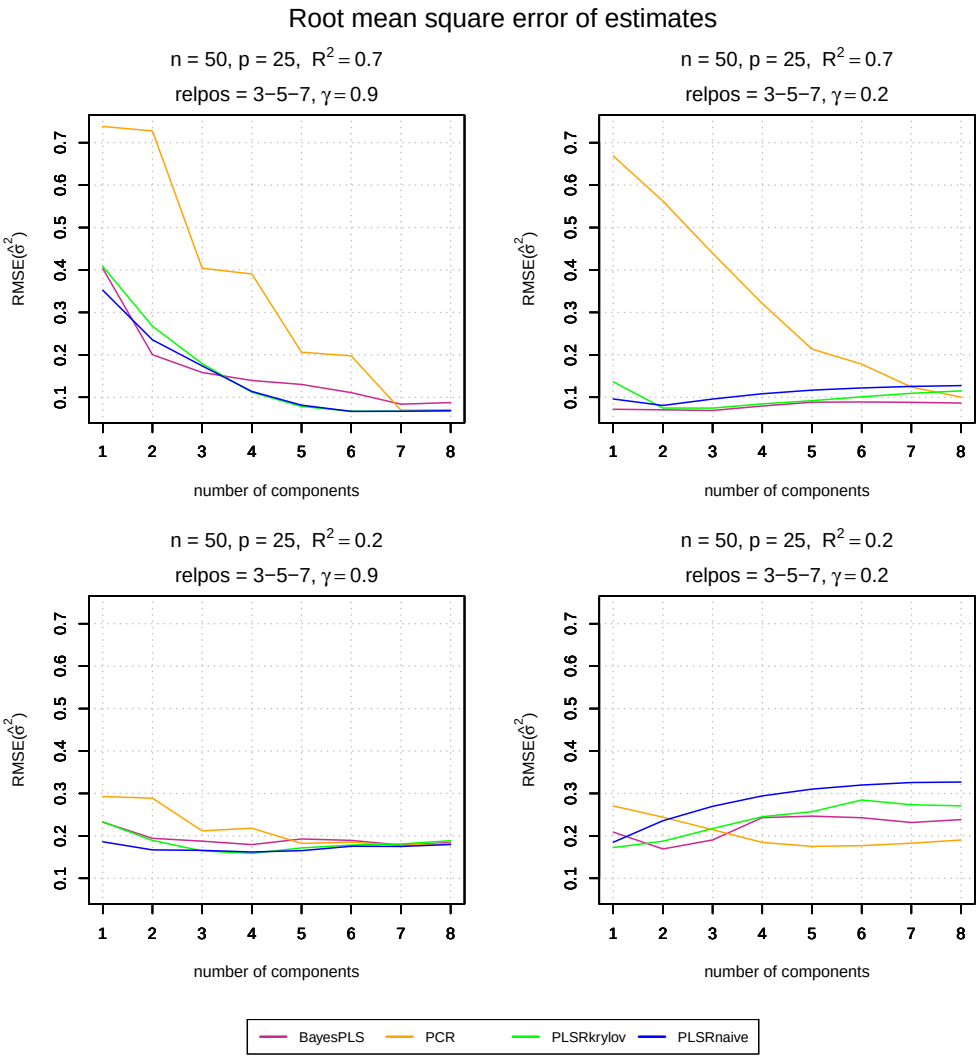


Figure 3: Root mean square error of the estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The plots belong to dp3, dp4, dp11 and dp12, all having $n = 50$, $p = 25$ and $relpos = \{3, 5, 7\}$.

some situations, especially when γ is small, the RMSE of the PLSRnaive estimates increases quite rapidly for $k > 1$.

Comparing only the RMSE's of the PLSRnaive and PLSRkrylov estimates in figure 2, 3 and 5, they are almost identical when γ is large, and when γ is small their behaviour is also quite similar, only with the RMSE of the PLSRnaive estimates being a bit larger than that of the PLSRkrylov estimates. Remembering that the PLSRnaive and PLSRkrylov σ^2-estimator is found by dividing the SSE with the degrees of freedom, and keeping in mind that their SSE's are equal, since it is calculated from the same model

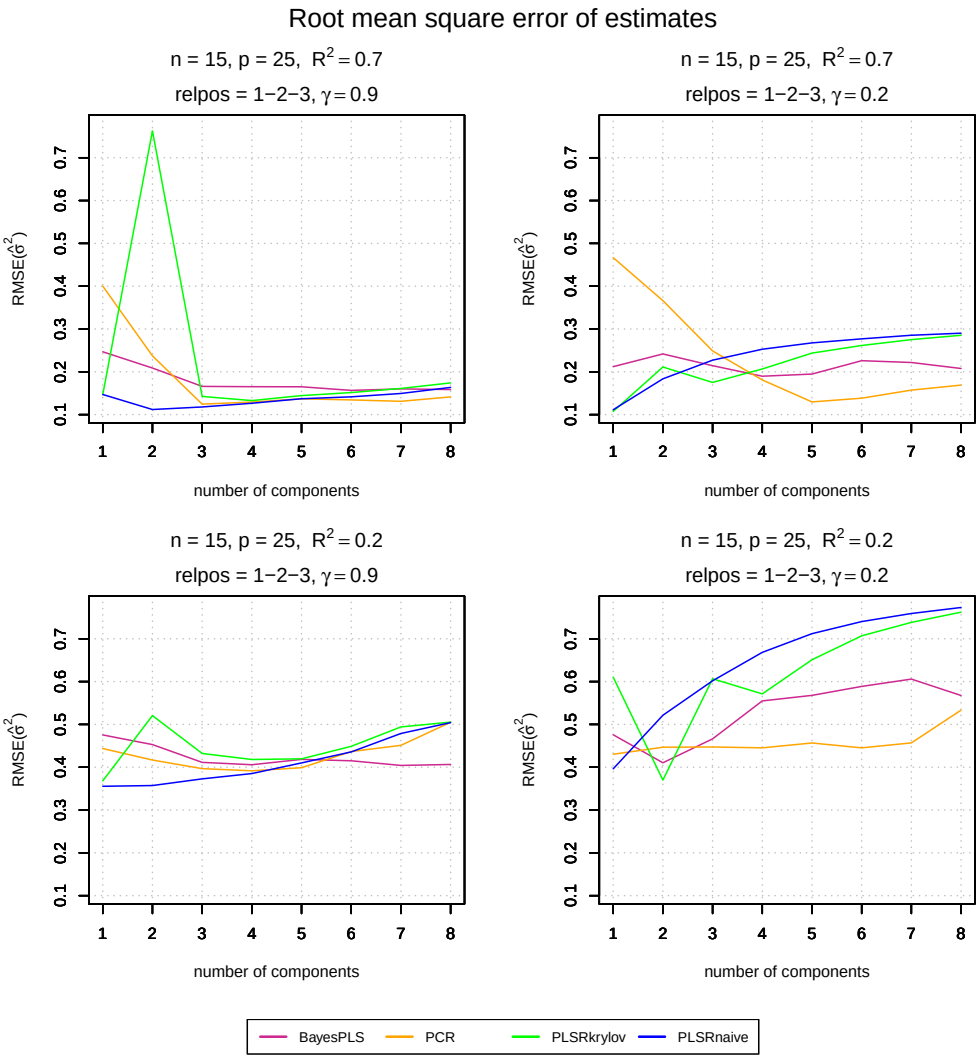


Figure 4: Root mean square error of the estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The plots belong to dp5, dp6, dp13 and dp14, all having $n = 15$, $p = 25$ and $relpos = \{1, 2, 3\}$.

fitted by PLSR, the only element that differentiates between the PLSR-naive estimator and the PLSRkrylov estimator is the choice of degrees of freedom used for the estimator. Following are the Krylov estimates of the DoF of dp10 (lower right plot of figure 2) with 2 components, as an example:

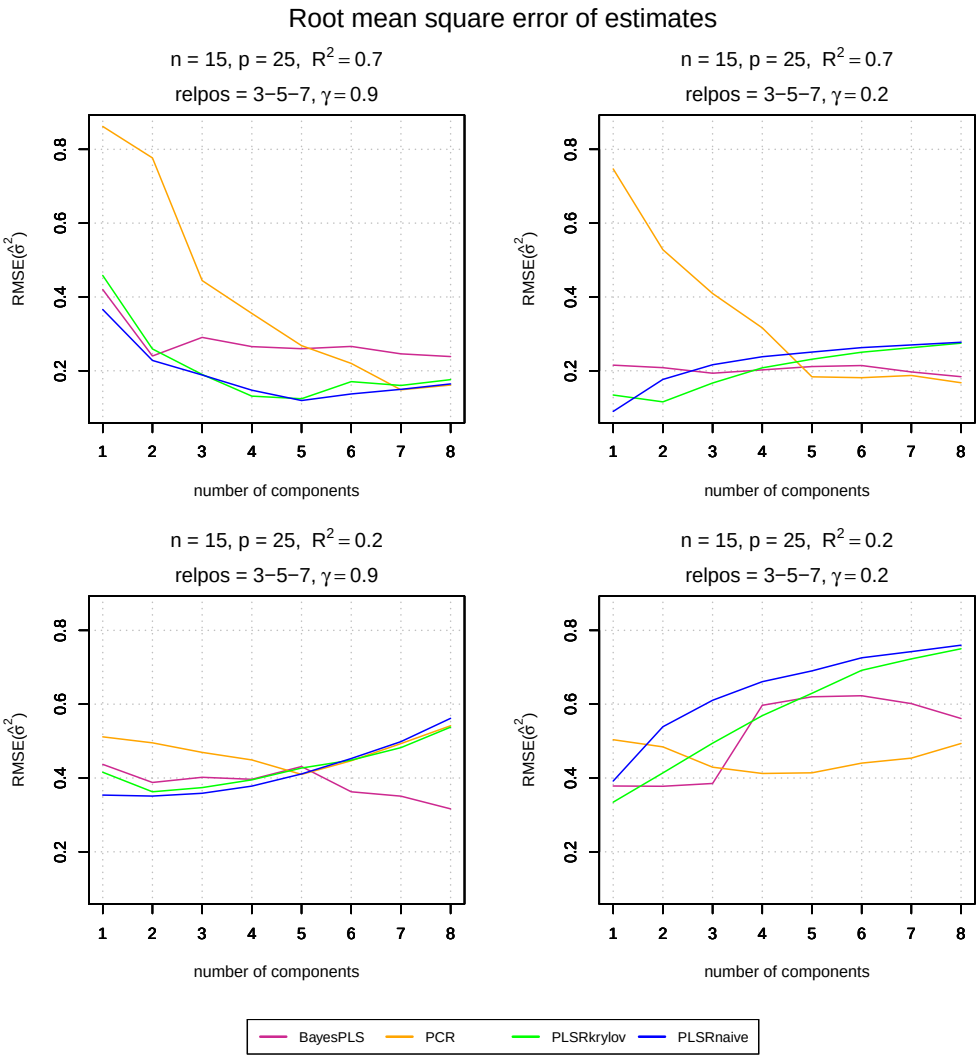


Figure 5: Root mean square error of the estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The plots belong to dp7, dp8, dp15 and dp16, all having $n = 15$, $p = 25$ and $relpos = \{3, 5, 7\}$.

	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Seed 6	Seed 7
r1	9.7908	9.7983	12.4978	11.9634	10.6185	9.1964	11.3779
r2	12.7983	12.1846	17.3590	9.6715	10.3793	10.9729	26.0000
r3	11.6814	10.4959	11.6823	12.0115	9.5507	8.9191	11.5473

Disregarding the seed7-r2 estimate, which is an upper bound DoF, all the DoF-estimates have a value between 8.9 and 17.4, with the majority being between 9 and 13. In comparison, the naive estimate of degrees of freedom is $k + 1$, which in this case equals 3. The fact that the RMSE of

the PLSRkrylov estimates is clearly smaller than that of the PLSRnaive estimates may suggest that the true value of the degrees of freedom is in fact larger than $k+1$ in the particular situation of dp10 with 2 components.

Figure 4 show some erratic behaviour of the RMSR of the PLSRkrylov estimates, especially in the upper left plot (dp5), where the RMSE reaches an unexpected peak for $k = 2$. The PLSRkrylov σ^2-estimates for dp5 with 2 components are

	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Seed 6	Seed 7
r1	0.22362	0.35830	0.13796	0.28804	0.07312	0.26781	0.14404
r2	0.13687	3.67224	0.15571	0.19756	0.28483	0.16028	NA
r3	0.13377	0.29861	0.43007	0.24704	0.35614	0.37615	0.41033

(The NA is due to the removal of negative and upper bound DoF's.) Here the true value of σ^2 is 0.3. The seed2-r2 estimate with a value of 3.67 is almost 10 times larger than the true value, and it is also much larger than all the other σ^2-estimates for dp5 with 2 components. The corresponding DoF-estimates are

	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Seed 6	Seed 7
r1	3.64164	3.17395	2.97611	3.13670	3.96819	2.53925	3.08775
r2	2.32001	13.70660	2.92633	3.07782	2.77503	2.99436	-10.76062
r3	2.84871	2.80426	4.02443	2.92254	3.09747	3.92587	2.71768

The seed2-r2 DoF-estimate is also notably larger than the other DoF-estimates. Since it is not equal to the upper bound (which in this case is 14), its corresponding σ^2-estimate has not been removed from the data, and it is obviously what is causing the inflated RMSE.

Unlike the RMSE of the PLSRnaive estimates, the RMSE of the Bayes PLS estimates seems to sometimes behave quite randomly with regard to the size of k. However, a small number of components returns an acceptably small RMSE in all situations except when γ is large and R^2 is large (as also mentioned previously).

8.1.2 Analysis of the effects of the simulation factors

To analyze the main effects and interaction effects of the factors of the simulation, a mixed model was fitted, including also the random effects

of seed and r (replicate nested within seed). The full model, with all interactions, is inconveniently long to be expressed here with symbols, so a simplified version of the model, including only the main effects, is given instead

$$\omega_{abcdefgh} = \mu + \alpha_a + \beta_b + \zeta_c + \eta_d + \nu_e + \rho_f + \phi_g + \psi_{h(g)} + \epsilon_{abcdefgh} \quad (21)$$

where

- ω is the estimation error as defined in (20)
- μ is the overall mean
- α_a is the effect of method at level a
- β_b is the effect of component at level b
- ζ_c is the effect of n at level c
- η_d is the effect of $relpos$ at level d
- ν_e is the effect of γ at level e
- ρ_f is the effect of R^2 at level f
- ϕ_g is the random effect of seed at level g
- $\psi_{h(g)}$ is the random effect of r at level h within seed at level g

The interaction effect terms, omitted in (21), would for example be symbolized by (for a second order interaction)

$$(\alpha\beta)_{ab}$$

in this case meaning the interaction effect of method at level a and component at level b.

Experience so far has showed that the PLSRkrylov estimator can be highly unreliable in individual cases. Therefore it has been decided to disregard this estimator in the analysis of variance.

The levels of the different factors are

- $a = 1, 2, 3$, corresponding to the estimators Bayes PLS, PCR, and PLSRnaive
- $b = 1, 2, ..., 8$, indicating the number of components
- $c = 1, 2$: $n = 15$ and $n = 50$
- $d = 1, 2$: $relpos = 1, 2, 3$ and $relpos = 3, 5, 7$
- $e = 1, 2$: $\gamma = 0.2$ and $\gamma = 0.9$
- $f = 1, 2$: $R^2 = 0.2$ and $R^2 = 0.7$
- $g = 1, 2, ..., 7$
- $h = 1, 2, 3$

Also, $\phi \sim N(0, \tau_\phi^2)$, $\psi \sim N(0, \tau_\psi^2)$ and $\epsilon \sim N(0, \tau^2)$.

The full model with all interactions up to the sixth order interaction was fitted in R, and the ANOVA table of the fitted model can be found in Appendix C.3.1. Using a test level of 0.05, the table shows that the sixth order interaction effect and all of the fifth order interactions effects are non-significant. Among the fourth order interactions there were four significant interactions:

- method - component - $relpos$ - R^2
- method - component - γ - R^2
- method - n - γ - R^2
- component - $relpos$ - γ - R^2

All of the 12 third order interactions that make up the four significant fourth order interactions mentioned above were retained in the fitted model. In addition, four other third order interactions were significant on a 0.05 level:

- method - component - n
- method - $relpos$ - γ
- component - n - γ

- component - n - R^2

Out of the total number of 15 second order interactions, all were significant except for the n-*relpos*-interaction.

The reduced model was fitted in R, and the ANOVA table of the model can be found in Appendix C.3.2. Note that the third order interaction of component - γ - R^2 obtained a p value of 0.424 (> 0.05), but it was nevertheless kept in the fitted model because it is a part of a higher order interaction included in the fitted model.

Plots of some of the interaction effects of the reduced fitted mixed model are presented and interpreted in this section. None of the lower order interactions that are included in a higher order interaction will be considered, as they only offer a simplification (and maybe even an oversimplification) of the information and trends that can be seen in the higher order interaction effect plots. Plots that are not directly commented on are placed in Appendix D.2.

Any interaction that does not include the factor component must be regarded with caution, as those effects are averages over all choices of number of components to include in the fitted model. Because one estimator can have quite a large estimation error for one choice of k and still obtain a reasonably small estimation error for another choice of k, these averages may not be directly comparable between estimators. However, the change of estimation error for one estimator due to the other factors in the interaction may be compared with the others.

For all three estimators, the minimum estimation errors are (as seen before) smaller when R^2 is large than when R^2 is small. Also, all three estimators seem to obtain an approximately equal minimum estimation error, but for a different choice of number of components included in the fitted model. The largest estimation errors occur for the PCR estimator when R^2 is large and only a few components are included in the fitted model.

Figure 6 shows than when R^2 is large and *relpos* is at a high level, all three estimators need more than 1 component included in the fitted model to obtain the minimum estimation error. This is also the case when R^2 is large and γ is large (figure 7). When R^2 is large and *relpos* is at a low level, the estimation errors of Bayes PLS and PLSRnaive are approximately equal regardless of k. They are also very similar when R^2 is large and γ is small.

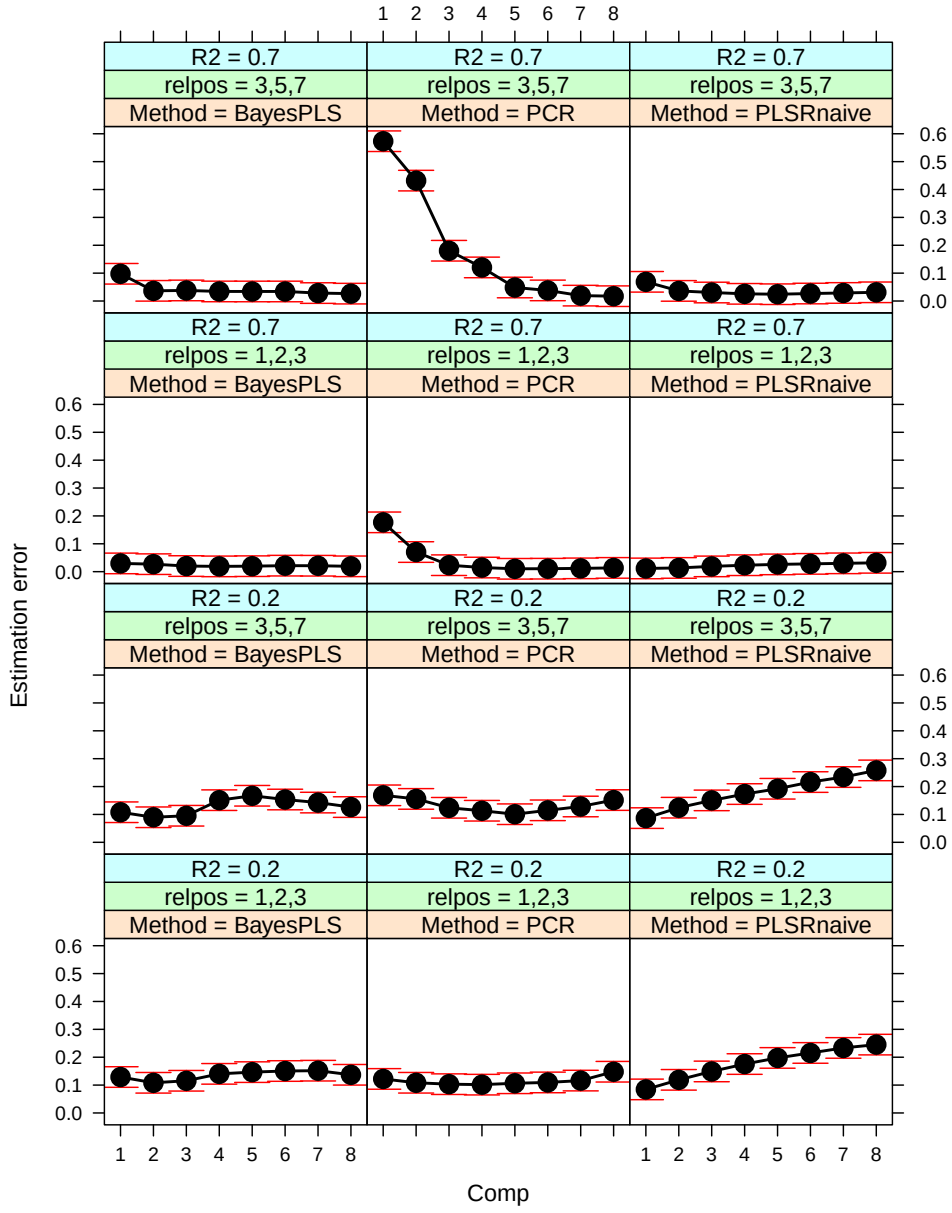


Figure 6: Interaction effect of methods/estimators, components, *relpos* and R^2. The red lines are confidence intervals.

When R^2 is small, a change in *relpos* does not seem to have much of an impact on any of the three estimators. For Bayes PLS and PCR it seems that all choices of k results in approximately equal estimation errors. For PLSRnaive, however, the choice of k seems more crucial when R^2 is small, as its estimation error appears to evenly increase as k increases. Also when

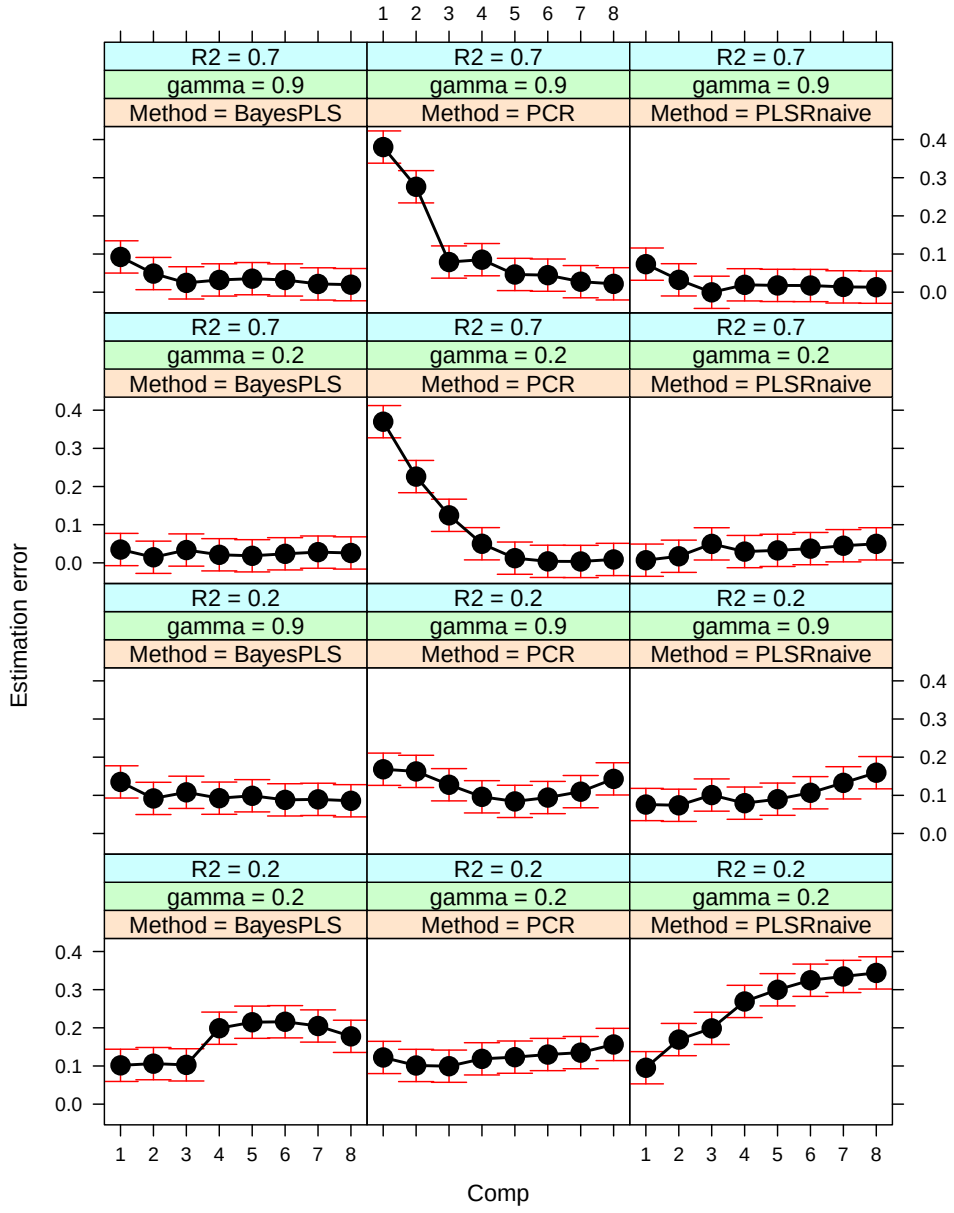


Figure 7: Interaction effect of methods/estimators, components, γ and R^2. The red lines are confidence intervals.

R^2 is small and γ is small, both the Bayes PLS and PLSRnaive estimators have a smaller estimation error when k is small.

Figure 8 shows that when R^2 is large, the effect of both n and γ is approximately equal for all three estimators. When R^2 is small, the PCR

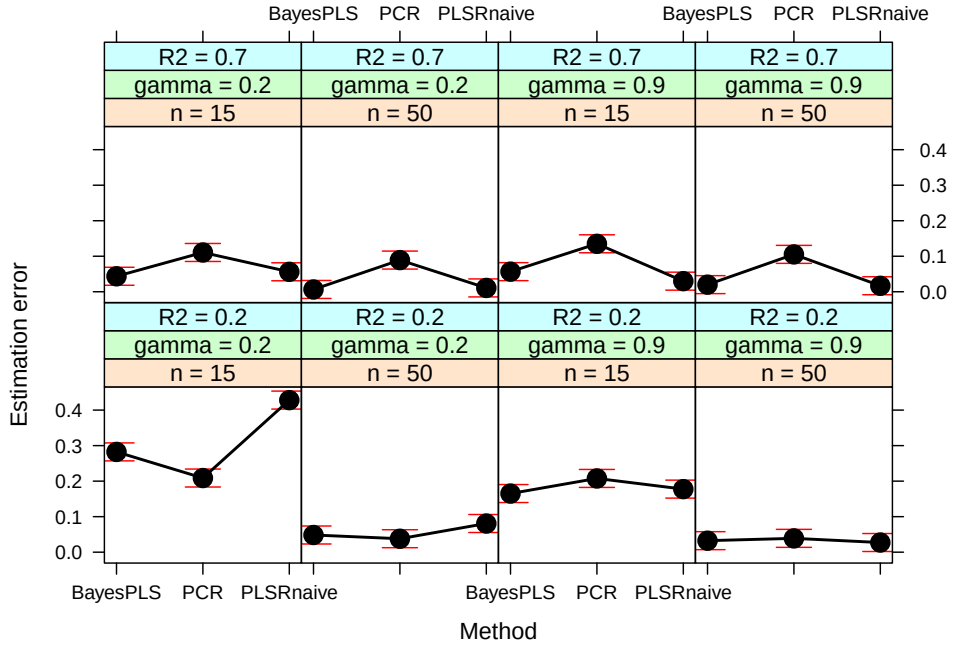


Figure 8: Interaction effect of methods/estimators, n, γ and R^2. The red lines are confidence intervals.

estimator seem less affected by a change in γ than the other two estimators.

The minimum estimation errors are smaller when n is large than when n is small for all three estimators (figure 9). Again, all three estimators obtain approximately equal minimum estimation errors for each respective level of n, but for a varying number of components. As seen before, the PCR estimator needs more components than the other two estimators to reach its minimum estimation error.

For the PCR estimator, the optimal number of components appears to be almost unaffected by the level of n. For the Bayes PLS and PLSRnaive estimator, all choices of k seem to result in approximately the same estimation error when n is large. When n is small, the estimation error of the Bayes PLS estimator behaves a bit randomly in relation to k, whereas the estimation error of the PLSRnaive estimator increases when k increases.

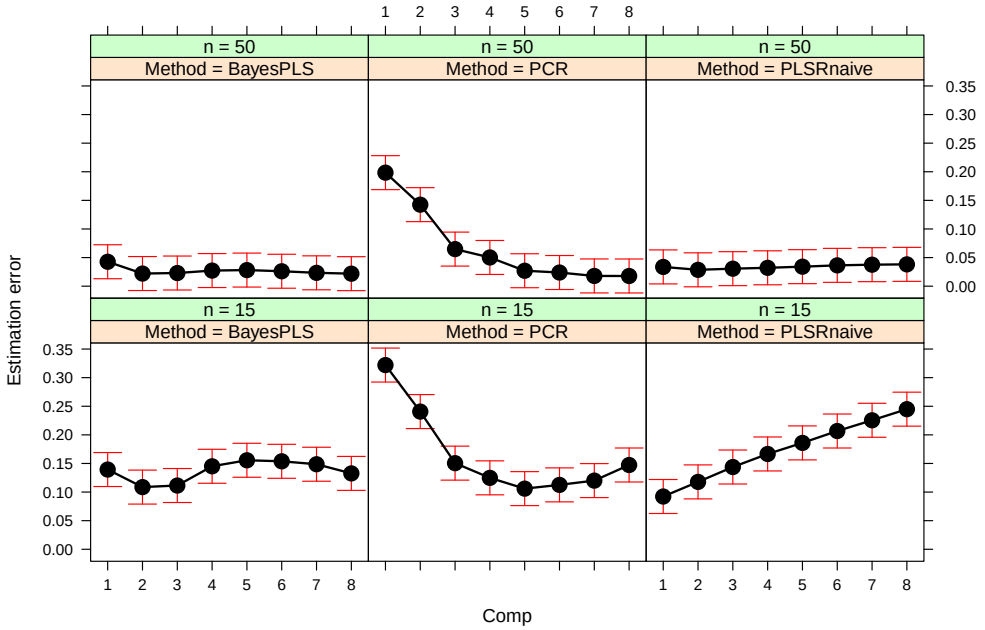


Figure 9: Interaction effect of method, components and n. The red lines are confidence intervals.

8.2 Bias of the PCR and PLSRnaive estimates

When $k = p$ and $p < n$, the PCR and PLSRnaive estimators of the noise variance both equal the OLS estimator of the noise variance, and so they are unbiased. To study how the bias of the estimates evolves as k increases towards p, a simulation study similar to the one presented above has been done. Now only the situations where $n > p$ are considered, namely design point 1, 2, 3, 4, 9, 10, 11 and 12, and only the two above-mentioned estimators. The number of components (k) ranges from 1 to 25 (equal to p). The *simrel*-arguments *relpos*, γ and R^2 still vary between two levels as before. The results are presented in figure 10 (plots of the average estimates) and figure 11 (plots of the RMSE's of the estimates). All the plotted averages and RMSE's are calculated from a number of 21 estimates (7 seeds and 3 replicates).

The plots in figure 10 show that the value of the PLSRnaive estimates are on average always less than the value of the PCR estimates, regardless of *relpos*, γ and R^2. Both estimators seem to be approximately unbiased for $k = 25$, as anticipated. The overall bias of the PCR estimates is not notably influenced by a change in γ. In general, the bias of the PCR estimates appears to be decreasing as k increases.

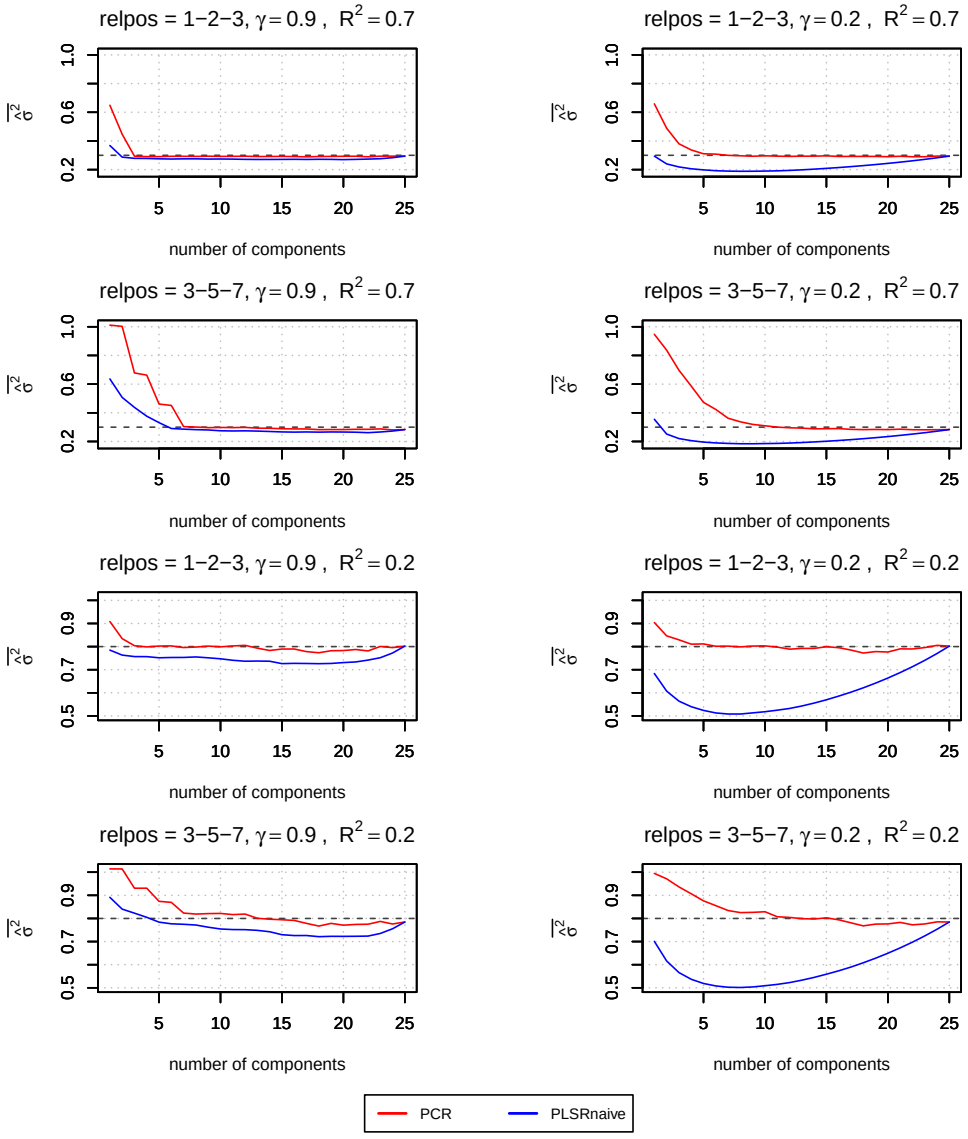


Figure 10: Average estimates vs. number of components. The dotted line is the true noise variance. The plots belong to dp1, dp2, dp3, dp4, dp9, dp10, dp11 and dp12, all having $n = 50$ and $p = 25$.

When γ is small and/or R^2 is small the overall bias of the PLSRnaive estimates is negative. The most extreme cases of negative bias are found when both γ and R^2 are small, then the PLSRnaive estimates do not reach their smallest possible average bias (in absolute value) until $k = 25$. When *relpos* is at a high level and γ is large the PLSRnaive estimates needs approximately 4-6 components to reach a near-minimum bias. In all other cases the PLSRnaive estimates obtains a near-minimum bias with only a few components included in the fitted model.

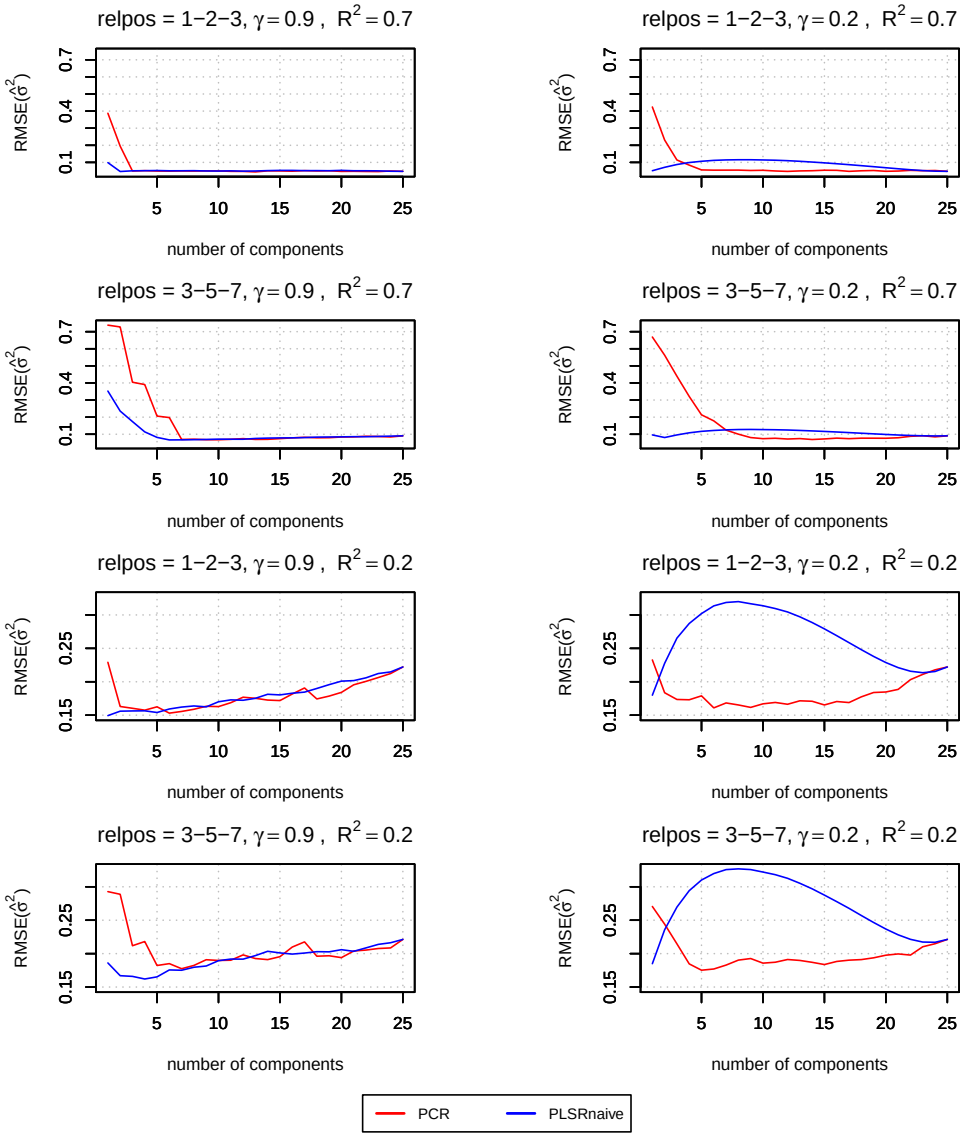


Figure 11: Root mean square error of the estimates vs. number of components. The plots belong to dp1, dp2, dp3, dp4, dp9, dp10, dp11 and dp12, all having $n = 50$ and $p = 25$.

From the RMSE plots in figure 11 it is clear that the PCR estimates on average performs poorly for $k = 1$(or a small number of components) regardless of *relpos*, γ and R^2. The choice of k giving the first near-minimum RMSE for the PCR estimates, however, seems inextricably connected to the position of the relevant components. When R^2 is large, after reaching its first near-minimum, the RMSE of the PCR estimates does not seem to change much as k increases. However when R^2 is small, the RMSE will (after reaching its minimum) start to increase as k increases. The

overall behaviour of the RMSE of the PCR estimates does not seem to be influenced much by the change in γ.

For the PLSRnaive estimates, regarding first the two lower right plots where γ and R^2 are both small, the minimum RMSE is clearly obtained for $k = 1$. As seen in figure 10 the estimates with this particular choice of k do not have the minimum bias (in absolute value) over all k, so the variance of the estimator for this particular choice of k must be small (due to the bias-variance decomposition of MSE explained in section 3.2.1). In fact, in all cases where γ is small, the PLSRnaive estimates seem to obtain a near-minimum RMSE for $k = 1$. When γ is large the RMSE of the PLSRnaive estimates behave more similarly to the RMSE of the PCR estimates, but PLSRnaive generally needs fewer components than PCR to obtain a near-minimum RMSE.

8.3 Estimating the number and positions of the relevant components

As briefly mentioned in section 5.2.1, estimating the number and positions of the relevant components can be a challenge. The true, relevant components are the only components that have a non-zero covariance with the response, so one approach is to consider the empirical covariance between the estimated component and the response. If the estimated covariances are similar enough to the true covariances, it could be possible to point out the positions of the true, relevant components.

In this section the true and estimated eigenvalues and covariances of the components has been compared for one example dataset from each of the 16 design points. The plots of the 8 first example datasets are shown in figure 12 and 13. The plots of the remaining 8 example datasets are in Appendix D.3.

Only a subset of the *simrel*-arguments decides the positions of the true relevant components and their covariances. Because the same seed is used to draw the sample data, all datasets that are simulated with the same level of *relpos* and the same value of γ, will also have the same positions of the true relevant components, with the exact same true covariances. Therefore, when regarding the plots of the true and estimated eigenvalues and covariances, some of the design point example datasets are directly comparable (dp1-dp5-dp9-dp13, dp2-dp6-dp10-dp14, and so on). What is different between them is the quality of the estimation, which depends

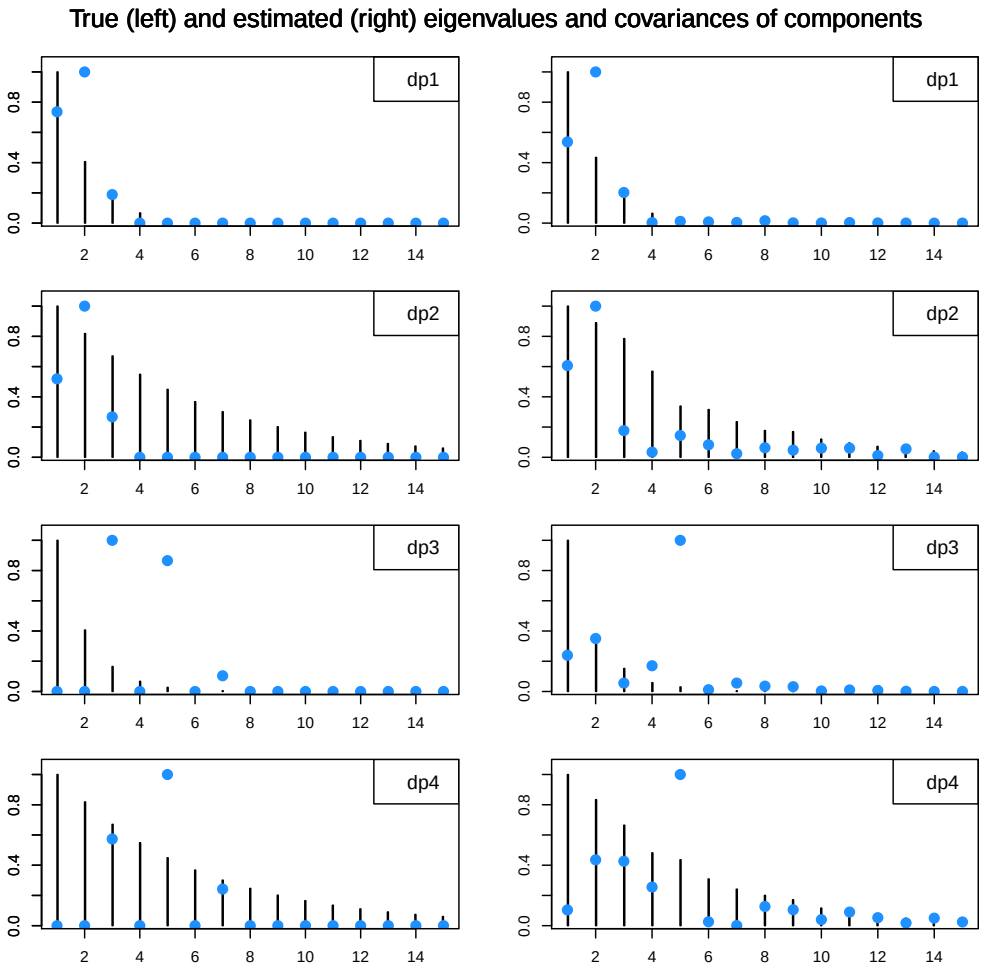


Figure 12: The true eigenvalues (bars) and covariances (blue dots) are illustrated in the plot to the left, and the corresponding estimates are illustrated in the plot to the right. All values are scaled by the largest occurring value. The design points featured here all have $n = 50$ and $R^2 = 0.7$.

on (in addition to *relpos* and γ) the remaining varied *simrel*-arguments n and R^2.

It must be emphasized that the true and estimated covariances and eigenvalues are all scaled, for visual purposes, and therefore their numerical values can not be compared directly. What should instead be investigated are the relative sizes of the covariances for each component. The largest covariance should be regarded as belonging to the most relevant component, and so on. For the eigenvalues, it is the speed of decline of the eigenvalues which is interesting, not the values of the eigenvalues themselves.

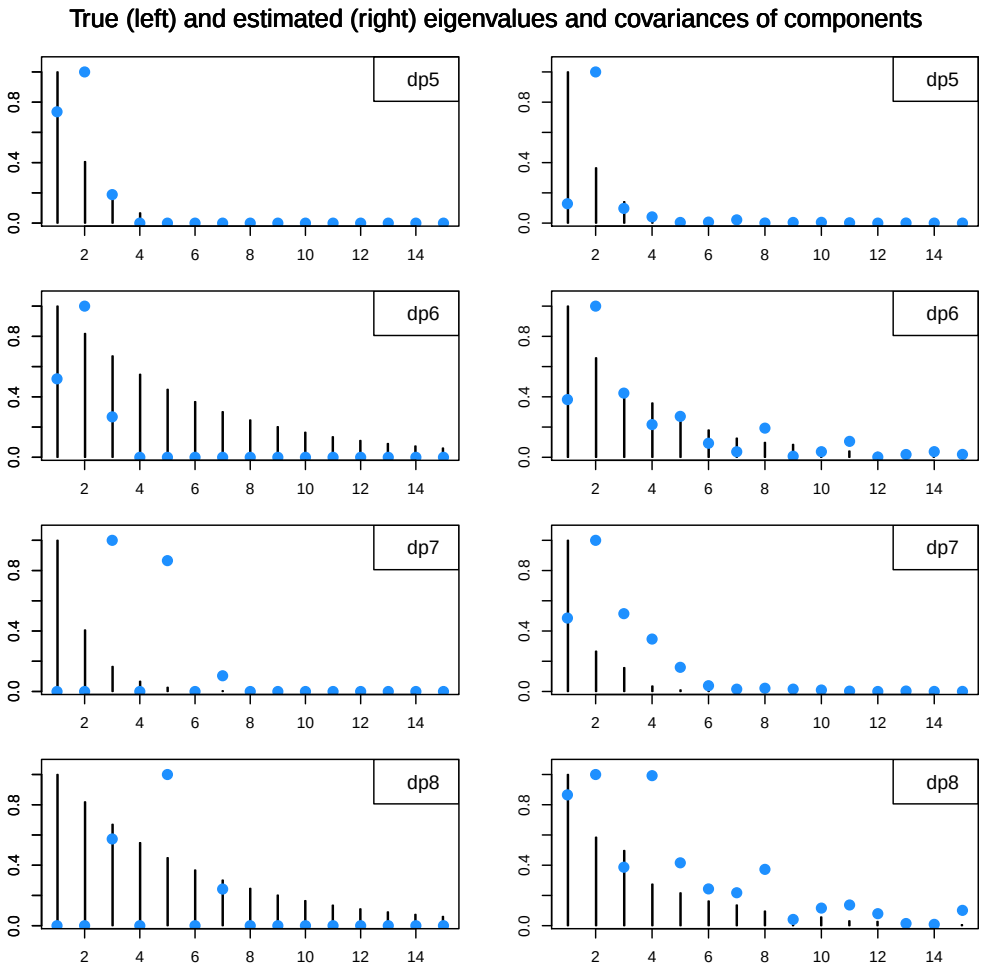


Figure 13: The true eigenvalues (bars) and covariances (blue dots) are illustrated in the plot to the left, and the corresponding estimates are illustrated in the plot to the right. All values are scaled by the largest occurring value. The design points featured here all have $n = 15$ and $R^2 = 0.7$.

In some cases the value of the estimated covariances actually seem to give a decent indication as to which are the true, relevant components, for example in the situations where *relpos* is at a low level, γ is large and R^2 is large (dp1 and dp5). However, in most other cases, the estimated covariances appear to be quite unreliable. Especially when γ is small, a large number of the estimated covariances are notably different from 0 (when their corresponding true covariances are, in fact, equal to 0). The differences between the relative size of the true covariance versus the relative size of the estimated covariance also seem random at times; a component that has one of the largest true covariances to the response (of all components) may have one of the smallest estimated covariances, and conversely.

In earlier plots of the RMSE of dp3 and dp11 (figure 3), a characteristic staircase shape of the RMSE of the PCR estimates suggested that the true positions of the relevant components could be identifiable. Comparing with the single example of estimated covariances of dp3 (figure 12), the true, relevant components are clearly not so easily identified. In this case, component number 3 (for example) has an estimated covariance with the response approximately equal to 0, which is highly inaccurate, as the true covariance in this case is relatively large. Although not illustrated here, experience has shown that changing the seed when simulating can result in greatly varying estimates of the covariance of the components. What this means is that different sets of sample data drawn from the same, true distribution may give very differing estimates of the number and positions of the relevant components.

While the estimated covariances seem to be quite inaccurate, the speed of decline of the estimated eigenvalues does appear to be quite close to the speed of decline of the true eigenvalues (a little less accurate when n is small). Although this observation may not bring any direct information regarding whether or not a component is relevant, it may still be of some interest with regard to estimating σ^2 (and possibly also for other reasons).

9 Discussion

9.1 The known and unknown factors/parameters

In this thesis the quality of the σ^2-estimate has been studied for different choices of estimation methods and of number of components (k) included in the fitted model, with regard to the following factors/parameters of the data:

1. The size of n (the number of samples) relative to the size of p (the number of predictor variables)
2. The position of the relevant components (*relpos*)
3. The speed of decline of the eigenvalues (γ)
4. The amount of variation explained by the model (R^2)

In any practical situation, the number of observations and the number of predictor variables are defined by the design of the experiment at hand, so the n - p relationship is known to the analyst. The last three factors, however, are usually unknown. So even though the results of the simulations of this thesis are presented on the basis of knowing the true values of *all* these factors, the reader should always keep in mind that in a practical situation the analyst will have to make a choice of which method to use, and of how many components to include in the fitted model, without knowing the true values of *relpos*, γ and $R2$.

As seen in section 8.3, trying to estimate *relpos* is quite difficult. As for the factor R^2, it is closely related to σ^2, so trying to estimate R^2 in order to find the best estimate of σ^2 is a sort of a circular reference problem.

Remember that γ is in fact a parameter created by *simrel*, in order to be able to simulate the eigenvalues of a dataset with the help of a straightforward formula. In a real-life, practical situation, the behaviour of the eigenvalues may not be modeled in this manner, as they would be expected to behave more randomly. Still, for argument's sake, one may assume that there exists some parameter γ^* similar to γ, defining the *overall* speed of

decline of the eigenvalues. As also seen in section 8.3, the speed of the decline of the eigenvalues appears to be somewhat easier to estimate than many of the other unknown parameters of a dataset. Therefore, it seems plausible that there could exist a reasonable estimator of γ^*, that may provide some valuable information on the nature of the dataset.

9.2 The performance of the estimators

The PCR estimator stands out from the other three estimators by almost always needing a higher number of components included in the fitted model to obtain its near-minimum estimation error. The factor *relpos* especially affects the optimal k for the PCR estimator. As mentioned above, estimates of *relpos* can be highly inaccurate, so relying on such estimates when choosing the size of k is not well-advised. Since all three other estimators always seem to obtain an equally small minimum estimation error as PCR, but for a smaller choice of k, it seems hard to justify the use of the PCR estimator in any of the situations discussed in this thesis.

When γ and R^2 are both large, the Bayes PLS, PLSRkrylov and PLSRnaive estimators all obtain smaller estimation errors when more than 1 component is included in the fitted model. For the PLSRnaive estimator the situation is quite different when γ is small, especially when n is also small: the optimal choice of k is 1, and the estimation error then increases quite rapidly as k increases. In other words, when using the PLSRnaive estimator, the factor γ appears to be especially important to consider when deciding which number of components to include in the fitted model. When γ is large, k should be larger than 1. When γ is small, k should *not* be larger than 1.

The PLSRkrylov estimator performs just as well or better than the PLSRnaive estimator in many situations. However, several weaknesses of the PLSRkrylov estimator have been identified through this simulation. Firstly, the practice of removing estimates from a simulation experiment is in itself questionable. It may be justified in this situation because the faulty estimates (with negative or upper bound DoF's) are so easily identified, also for a practitioner. Of course, for the practitioner this means that sometimes the method will simply fail in bringing an estimate altogether (as often as almost 1 out of 10 situations, jugding from the rate of failure seen in this simulation). What is worse is that sometimes, in some single incidents, the method returns a highly inaccurate estimate, that may *not*

be so detectable in a practical situation. The unpredictable behaviour of the PLSRkrylov estimator suggests that the method can not be relied on in a practical situation.

The estimation error of the Bayes PLS estimator sometimes behaves quite random, but with relatively little variation, for different choices of k. A small k seems to be a good choice for the Bayes PLS estimator in most situations, except for when γ and R^2 are both large (as mentioned above). If the choice of estimator is between Bayes PLS and PLSRnaive, the fact that the Bayes PLS estimator is somewhat less sensitive to the choice of k works to its advantage. The fact that the Bayes PLS estimator may be both more time-consuming and difficult to use makes the PLSRnaive estimator appear more attractive.

9.3 The degrees of freedom of PCR and PLSR

The naive estimate of degrees of freedom $(k + 1)$ has been investigated for both methods PCR and PLSR. The PCR estimator of σ^2 (with $k + 1$ degrees of freedom) steadily approaches the true value of σ^2 as k increases. This is a trend seen for all combinations of $relpos$, γ and R^2. The optimal choice of k is obviously affected by the level of $relpos$. However, a change in γ seems to have less of an impact on the average bias of the PCR estimates, suggesting that the degrees of freedom are not affected by the speed of decline of the eigenvalues (contradicting some of the claims of Hassani et al. [2012]). The naive estimate of the degrees of freedom for the PCR estimator therefore appears to be a reasonable estimate. When the PCR estimate of σ^2 is biased, it is likely a result of not including enough components in the fitted model, rather than of the estimate of the degrees of freedom being erroneous. It has also been shown that including non-relevant components with large eigenvalues in the fitted model may improve the precision of the estimation (which is in line with what Helland and Almøy [1994] argued for prediction).

For the PLSRnaive estimator, the bias of the estimate is affected by the size of γ and R^2. Especially when γ and R^2 is small, the true degrees of freedom is probably larger than the naive estimate of $k + 1$ (as also suggested by Krämer and Sugiyama [2011]). This claim is also backed up by the fact that the PLSRkrylov estimator often performs better than the PLSRnaive estimator in this specific situation, and the corresponding Krylov estimates of degrees of freedom have then been shown to be no-

tably larger than $k+1$. When γ is large, the Krylov estimates of degrees of freedom do not differ so much from the naive estimate of $k+1$.

9.4 Further studies

The RMSE's of the estimates are themselves estimates of the true RMSE's. Therefore, the inferences made in this thesis come with a margin of uncertainty related to the uncertainty of the RMSE-estimator. The most straightforward way of evaluating an estimator is by trying to estimate its expected value and variance, but since there is only one RMSE for each combination of method, design point and number of components, this is not possible. The only way to do this would be to repeat the entire experiment (with different seeds) several times.

The simulations of this thesis are limited to only consider certain selected levels of the parameters of a dataset, and therefore, the validity of the results are also limited. An expansion of the study, including more levels of the parameters, and possibly also other parameters, would contribute to ratify or invalidate the results discussed above. For example, it may seem like an unrealistically simple situation in which only 3 components are truly relevant. A more realistic (practically oriented) scenario would perhaps be to consider a *relpos* with more of a spread, with positions ranging from for example 1 up to p, and with m being notably larger than 3.

The parameter γ has stood out as a factor that could possibly have an effect on the quality of the σ^2-estimate. In addition to considering more levels of γ in a follow-up study, it would also be of interest to investigate the ability of estimating γ^* (as described previously). In such a study, the simulated datasets should not all have a 'smoothened' decline of the eigenvalues, as the datasets provided by *simrel* have.

It seems reasonable that the parameter p (the number of predictor variables, which has not been focused on in this thesis) should affect the degrees of freedom of both PCR and PLSR. The naive estimate of the degrees of freedom does not take the size of p into account. Is this reasonable? One can maybe argue that when the number of predictor variables increase, the search for relevant components becomes more extensive, and therefore the consumption of degrees of freedom should increase.

The findings of this thesis should be attemptively applied to real datasets. Since the noise variance is inextricably connected to the precision of pre-

diction, one suggestion is to compare the σ^2-estimate to the MSEP (mean square error of prediction) of real datasets, found by using (for example) cross-validation.

The estimation error/RMSE of the estimates has in this thesis been studied for all choices of k (up to a selected maximum number of components), which sometimes makes it difficult to compare between methods. A different approach could have been to only regard the number of components that return the smallest error, either for each individual estimation, or for all comparable estimates (i.e. the smallest RMSE). An even more elegant tactic would be to find some tradeoff measure that accomodates both criterias of smallest possible estimation error and fewest possible number of components included in the fitted model, and then compare between methods.

It is quite unsatisfactory that the PLSRkrylov estimator seems to perform so well all in all, but then in some odd, individual cases, it fails with no apparent explanations as to why. The indications seen in this thesis that the Krylov estimates of the degrees of freedom are, in some situations, more correct than the naive estimate of $k+1$, makes it tempting to investigate the Krylov estimator further, and try to find better implementations of the theory behind it, in order to reduce the amount of random, bad estimations.

9.5 Conclusion

There are notable differences in performance of the four σ^2-estimators investigated in this thesis. The PCR estimator requires a higher number of components included in the fitted model to obtain reasonably good estimates. The PLSRkrylov estimator sometimes performs well, but it is very unstabile and therefore unreliable. Out of the four estimators, the Bayes PLS and the PLSRnaive estimator provide the overall best and/or most stabile estimates of σ^2. The PLSRnaive estimator is more sensitive to the choice of number of components, especially relative to the collinearity of the predictor variables (quantified by γ, the speed of decline of the eigenvalues). There are indications that the true value of the degrees of freedom of PLSR is probably larger when the predictor variables are less correlated. The Bayes PLS estimator usually performs well with a few (2-3) components regardless of the structure of the data, but it is somewhat more difficult and time-consuming to use.

References

N. Krämer and M. Sugiyama. The degrees of freedom of partial least squares regression. *Journal of the American Statistical Association*, 2011.

S. Sæbø. simrel: Linear model data simulation and design of computer experiments, 2014. URL `https://CRAN.R-project.org/package=simrel`.

D. C. Lay. *Linear algebra and its applications*. Pearson, 3. edition, 2006.

I. S. Helland and T. Almøy. Comparison of prediction methods when only a few components are relevant. *Journal of the American Statistical Association*, 1994.

J. L. Devore and K. N. Berk. *Modern mathematical statistics with applications*. Thomson Brooks/Cole, 2007.

P. J. Bickel and K. A. Doksum. *Mathematical statistics: Basic ideas and selected topics vol. I.* Pearson Prentice Hall, New Jersey, 2. edition, 2007.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, London, 1. edition, 1996.

E. L. Lehmann and G. Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

I. T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2. edition, 2002.

S. Hassani, H. Martens, E. M. Qannari, and A. Kohler. Degrees of freedom estimation in principal component analysis and consensus principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2012.

S. Wold, H. Martens, and H. Wold. The multivariate calibration problem in chemistry solved by the pls method. In A. Ruhe and B. Kågström, editors, *Lecture Notes in Mathematics*, pages 286–293. Springer Verlag, Heidelberg, Germany, 1983.

H. Martens and T. Næs. *Multivariate calibration*. Wiley & sons, 1989.

I. S. Helland. Partial least squares regression and statistical models. *Scandinavian Journal of Statistics*, 1990.

N. Krämer and M. L. Braun. plsdof: Degrees of freedom and statistical inference for partial least squares regression, 2014. URL `https://CRAN.R-project.org/package=plsdof`.

I. S. Helland, S. Sæbø, and H. Tjelmeland. Near optimal prediction from relevant components. *Scandinavian Journal of Statistics*, 2012.

S. Sæbø. Bayespls: Bayesian estimation in pls regression, 2016,. URL `http://www.github.com/solvsa/BayesPLS`.

F. G. Giesbrecht and M. L. Gumpertz. *Planning, construction and statistical analysis of comparative experiments*. Wiley & sons, 2004.

S. Sæbø, T. Almøy, and I. S. Helland. simrel - a versatile method for linear model data simulation based on the concept of a relevant subspace and relevant predictors. *Chemometrics and Intelligent Laboratory Systems*, 2015.

A Proofs

A.1 Expected value and variance of the PCR estimators

For the sake of readability, all PCR estimates mentioned in this section will be denoted with a subscript $_k$, rather than $_{PCR,k}$.

Theory of linear regression gives the following

$$E(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\beta}$$
$$\boldsymbol{\beta} = \sum_{i=1}^{p} \frac{1}{\lambda_i} \boldsymbol{e}_i \boldsymbol{e}_i^t \boldsymbol{X}^t \boldsymbol{y}$$

where λ_i and $\boldsymbol{e}_i$ are the i$^{\text{th}}$ eigenvalue and eigenvector of $\Sigma_{\boldsymbol{xx}}$, respectively. The eigenvectors are all of unit length and orthogonal, so $\boldsymbol{e}_i^t \boldsymbol{e}_j$ is equal to 1 if $i = j$ and 0 if $i \neq j$.

As shown in section 5.2

$$\hat{\boldsymbol{y}}_k = \boldsymbol{H}_k \boldsymbol{y}$$

where $\boldsymbol{H}_k = \boldsymbol{Z}(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\boldsymbol{Z}^t$ (the hat matrix).

Assume that $\boldsymbol{X}$ is given, and that all variables are centered. Then $\boldsymbol{Z} = \boldsymbol{X}\hat{\boldsymbol{E}}_k$, and the PCR estimator of the regression coefficients for the original predictor variables is given by $\hat{\boldsymbol{\beta}}_k = \hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\boldsymbol{y}$.

A.1.1 Bias and variance of $\hat{\boldsymbol{\beta}}_k$

The bias of $\hat{\boldsymbol{\beta}}_k$ is

$$\begin{aligned}
E(\hat{\boldsymbol{\beta}}_k) - \boldsymbol{\beta} &= E(\hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\boldsymbol{y}) - \boldsymbol{\beta} \\
&= \hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t \cdot E(\boldsymbol{y}) - \boldsymbol{\beta} \\
&= \hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\beta} \\
&= \sum_{i=1}^{k}\frac{1}{\hat{\lambda}_i}\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t\sum_{j=1}^{p}\hat{\lambda}_j\hat{\boldsymbol{e}}_j\hat{\boldsymbol{e}}_j^t\boldsymbol{\beta} - \boldsymbol{\beta} \\
&= \sum_{i=1}^{k}\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t\boldsymbol{\beta} - \boldsymbol{\beta} \\
&= (\sum_{i=1}^{k}\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t - \boldsymbol{I})\boldsymbol{\beta} \\
&= -\sum_{i>k}^{p}\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t\boldsymbol{\beta}
\end{aligned}$$

The variance of $\hat{\boldsymbol{\beta}}_k$ is

$$\begin{aligned}
Var(\hat{\boldsymbol{\beta}}_k) &= Var(\hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\boldsymbol{y}) \\
&= \hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t \;\; Var(\boldsymbol{y}) \;\; (\hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t)^t \\
&= \sigma^2(\hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\boldsymbol{X}\hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t) \\
&= \frac{\sigma^2}{n-1}(\sum_{i=1}^{k}\frac{1}{\hat{\lambda}_i}\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t\sum_{j=1}^{p}\hat{\lambda}_j\hat{\boldsymbol{e}}_j\hat{\boldsymbol{e}}_j^t\sum_{q=1}^{k}\frac{1}{\hat{\lambda}_q}\hat{\boldsymbol{e}}_q\hat{\boldsymbol{e}}_q^t) \\
&= \frac{\sigma^2}{n-1}(\sum_{i=1}^{k}\frac{1}{\hat{\lambda}_i}\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t)
\end{aligned}$$

A.1.2 Expected value and variance of SSE_k

The sum of the squared errors of PCR is given by

$$\begin{aligned} SSE_k &= (\boldsymbol{y} - \hat{\boldsymbol{y}}_k)^t(\boldsymbol{y} - \hat{\boldsymbol{y}}_k) \\ &= ((\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{y})^t((\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{y}) \\ &= \boldsymbol{y}^t(\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{y} \end{aligned}$$

The expected value of SSE_k is

$$\begin{aligned} E(SSE_k) &= E(\boldsymbol{y}^t(\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{y}) \\ &= tr((\boldsymbol{I} - \boldsymbol{H}_k)\sigma^2) + \boldsymbol{\beta}^t\boldsymbol{X}^t(\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{X}\boldsymbol{\beta} \\ &= \sigma^2(n - (k+1)) + \boldsymbol{\beta}^t(\boldsymbol{X}^t\boldsymbol{X} - \boldsymbol{X}^t\boldsymbol{H}_k\boldsymbol{X})\boldsymbol{\beta} \\ &= \sigma^2(n - (k+1)) + \boldsymbol{\beta}^t(\boldsymbol{X}^t\boldsymbol{X} - \boldsymbol{X}^t\boldsymbol{Z}(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\boldsymbol{Z}^t\boldsymbol{X})\boldsymbol{\beta} \\ &= \sigma^2(n - (k+1)) + \boldsymbol{\beta}^t(\boldsymbol{X}^t\boldsymbol{X} - \boldsymbol{X}^t\boldsymbol{X}\hat{\boldsymbol{E}}_k(\boldsymbol{Z}^t\boldsymbol{Z})^{-1}\hat{\boldsymbol{E}}_k^t\boldsymbol{X}^t\boldsymbol{X})\boldsymbol{\beta} \\ &= \sigma^2(n - (k+1)) + (n-1)\boldsymbol{\beta}^t(\sum_{i=1}^{p}\hat{\lambda}_i\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t \\ &\quad - \sum_{j=1}^{p}\hat{\lambda}_j\hat{\boldsymbol{e}}_j\hat{\boldsymbol{e}}_j^t\sum_{q=1}^{k}\frac{1}{\hat{\lambda}_q}\hat{\boldsymbol{e}}_q\hat{\boldsymbol{e}}_q^t\sum_{r=1}^{p}\hat{\lambda}_r\hat{\boldsymbol{e}}_r\hat{\boldsymbol{e}}_r^t)\boldsymbol{\beta} \\ &= \sigma^2(n - (k+1)) + (n-1)\boldsymbol{\beta}^t(\sum_{i=1}^{p}\hat{\lambda}_i\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t - \sum_{j=1}^{k}\hat{\lambda}_j\hat{\boldsymbol{e}}_j\hat{\boldsymbol{e}}_j^t)\boldsymbol{\beta} \\ &= \sigma^2(n - (k+1)) + (n-1)\boldsymbol{\beta}^t\sum_{i>k}^{p}\hat{\lambda}_i\hat{\boldsymbol{e}}_i\hat{\boldsymbol{e}}_i^t\boldsymbol{\beta} \\ &= \sigma^2(n - (k+1)) + (n-1)\sum_{i>k}^{p}\hat{\lambda}_i(\hat{\boldsymbol{e}}_i^t\boldsymbol{\beta})^2 \end{aligned}$$

The variance of SSE_k is

$$\begin{aligned} Var(SSE_k) &= Var(\boldsymbol{y}^t(\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{y}) \\ &= 2tr((\boldsymbol{I} - \boldsymbol{H}_k)\sigma^2(\boldsymbol{I} - \boldsymbol{H}_k)\sigma^2) + 4\boldsymbol{\beta}^t\boldsymbol{X}^t(\boldsymbol{I} - \boldsymbol{H}_k)\sigma^2(\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{X}\boldsymbol{\beta} \\ &= 2\sigma^4(n - (k+1)) + 4\sigma^2\boldsymbol{\beta}^t\boldsymbol{X}^t(\boldsymbol{I} - \boldsymbol{H}_k)\boldsymbol{X}\boldsymbol{\beta} \\ &= 2\sigma^4(n - (k+1)) + 4\sigma^2(n-1)\sum_{i>k}^{p}\hat{\lambda}_i(\hat{\boldsymbol{e}}_i^t\boldsymbol{\beta})^2 \end{aligned}$$

A.1.3 Bias and variance of $\hat{\sigma}_k^2$

Regarding the naive estimator

$$\hat{\sigma}_k^2 = \frac{SSE_k}{n-(k+1)}$$

The bias of $\hat{\sigma}_k^2$ is

$$\begin{aligned}
E(\hat{\sigma}_k^2) - \sigma^2 &= E\Big(\frac{SSE_k}{n-(k+1)}\Big) - \sigma^2 \\
&= \frac{1}{n-(k+1)} E(SSE_k) - \sigma^2 \\
&= \frac{1}{n-(k+1)} ((n-(k+1))\sigma^2 + (n-1)\sum_{i>k}^{p} \hat{\lambda}_i (\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2) - \sigma^2 \\
&= \frac{(n-1)\sum_{i>k}^{p} \hat{\lambda}_i (\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2}{n-(k+1)}
\end{aligned}$$

The variance of $\hat{\sigma}_k^2$ is

$$\begin{aligned}
Var(\hat{\sigma}_k^2) &= Var\Big(\frac{SSE_k}{n-(k+1)}\Big) \\
&= \frac{1}{(n-(k+1))^2}\ Var(SSE_k) \\
&= \frac{1}{(n-(k+1))^2} (2\sigma^4(n-(k+1)) + 4\sigma^2(n-1)\sum_{i>k}^{p} \hat{\lambda}_i (\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2) \\
&= \frac{2\sigma^4}{n-(k+1)} + \frac{4\sigma^2(n-1)\sum_{i>k}^{p} \hat{\lambda}_i (\hat{\boldsymbol{e}}_i^t \boldsymbol{\beta})^2)}{(n-(k+1))^2}
\end{aligned}$$

B Algorithms

B.1 The orthogonalized PLSR algorithm

The orthogonalized PLSR algorithm for one response variable (Martens & Næs, 1989):

1. Scale the data to obtain the standardized variables $\boldsymbol{X}_0$ and $\boldsymbol{y}_0$.
2. Choose k_{max} to be a number higher than the expected number of latent PLS components (k).
3. For each level of $a = 1, ..., k_{max}$ run through steps 3a-3f:
   (a) Compute the loading weigths $\boldsymbol{w}_a$:

   $$\boldsymbol{w}_a = c\boldsymbol{X}_{a-1}^t \boldsymbol{y}_{a-1}$$

   where c is a number scaling $\boldsymbol{w}_a$ to a unit vector

   $$c = (\boldsymbol{y}_{a-1}^t \boldsymbol{X}_{a-1} \boldsymbol{X}_{a-1}^t \boldsymbol{y}_{a-1})^{-0.5}$$

   (b) Compute the scores $\boldsymbol{t}_a$:

   $$\boldsymbol{t}_a = \boldsymbol{X}_{a-1} \boldsymbol{w}_a$$

   (c) Compute the $\boldsymbol{X}$-loadings $\boldsymbol{p}_a$:

   $$\boldsymbol{p}_a = \frac{\boldsymbol{X}_{a-1}^t \boldsymbol{t}_a}{\boldsymbol{t}_a^t \boldsymbol{t}_a}$$

   (d) Compute the $\boldsymbol{y}$-loading q_a:

   $$q_a = \frac{\boldsymbol{y}_{a-1}^t \boldsymbol{t}_a}{\boldsymbol{t}_a^t \boldsymbol{t}_a}$$

   (e) Deflate the matrices $\boldsymbol{X}_0$ and $\boldsymbol{y}_0$ by subtracting the information related to the current component:

$$\boldsymbol{E} = \boldsymbol{X}_{a-1} - \boldsymbol{t}_a \boldsymbol{p}_a^t$$

$$\boldsymbol{f} = \boldsymbol{y}_{a-1} - \boldsymbol{t}_a q_a$$

(f) Update the inputs:

$$\boldsymbol{X}_a = \boldsymbol{E}$$

$$\boldsymbol{y}_a = \boldsymbol{f}$$

$$a = a + 1$$

4. Choose k, the number of components to use in the fitted model.

5. Estimate the regression coefficients for the fitted model:

$$\hat{\boldsymbol{\beta}}_{PLSR,k} = \boldsymbol{W}(\boldsymbol{P}^t\boldsymbol{W})^{-1}\boldsymbol{q}$$

where $\underset{p\times k}{\boldsymbol{W}} = \left[\boldsymbol{w}_1\ \boldsymbol{w}_2\ ...\ \boldsymbol{w}_k\right]$, $\underset{p\times k}{\boldsymbol{P}} = \left[\boldsymbol{p}_1\ \boldsymbol{p}_2\ ...\ \boldsymbol{p}_k\right]$, and $\underset{k\times 1}{\boldsymbol{q}} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_k \end{bmatrix}$.

C Tables and data

C.1 Overview of the 16 design points used in the simulations

Table 2: The *simrel*-arguments for each dp

dp	**n**	**p**	**m**	**relpos**	γ	$\mathbf{R^2}$
1	50	25	3	{1, 2, 3}	0.9	0.7
2	50	25	3	{1, 2, 3}	0.2	0.7
3	50	25	3	{3, 5, 7}	0.9	0.7
4	50	25	3	{3, 5, 7}	0.2	0.7
5	15	25	3	{1, 2, 3}	0.9	0.7
6	15	25	3	{1, 2, 3}	0.2	0.7
7	15	25	3	{3, 5, 7}	0.9	0.7
8	15	25	3	{3, 5, 7}	0.2	0.7
9	50	25	3	{1, 2, 3}	0.9	0.2
10	50	25	3	{1, 2, 3}	0.2	0.2
11	50	25	3	{3, 5, 7}	0.9	0.2
12	50	25	3	{3, 5, 7}	0.2	0.2
13	15	25	3	{1, 2, 3}	0.9	0.2
14	15	25	3	{1, 2, 3}	0.2	0.2
15	15	25	3	{3, 5, 7}	0.9	0.2
16	15	25	3	{3, 5, 7}	0.2	0.2

C.2 Summary of negative and upper bound DoF's

Following is an overview of the number of negative and upper bound DoF's occuring in the Krylov DoF estimates. The columns correspond to the number of components included in the fitted model.

```
$dp1
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    0    0    0    0    0    0    0
upperboundDoF    0    0    0    0    0    0    0    0

$dp2
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    3    4    4    4    5    4    3
upperboundDoF    0    0    0    0    0    0    0    0

$dp3
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    1    1    0    0    0    0    0
upperboundDoF    0    0    0    0    0    0    0    0

$dp4
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    0    1    1    2    1    2    1
upperboundDoF    0    0    0    1    1    2    3    2

$dp5
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    1    1    1    0    0    0    0
upperboundDoF    0    0    0    0    0    0    0    0

$dp6
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    3    3    3    3    3    2    1
upperboundDoF    0    2    3    3    4    4    3    3

$dp7
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    1    1    1    2    2    2    3
upperboundDoF    0    0    1    0    0    0    1    1

$dp8
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    0    0    0    0    0    0    0
upperboundDoF    0    1    2    2    1    1    1    1
```

```
$dp9
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    2    3    2    1    0    0    0
upperboundDoF    0    1    1    0    1    1    1    1

$dp10
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    0    1    2    2    1    3    3
upperboundDoF    0    1    2    2    4    3    3    3

$dp11
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    1    0    0    0    0    0    0
upperboundDoF    0    0    0    0    0    1    1    1

$dp12
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    0    0    3    3    3    4    4
upperboundDoF    0    0    1    0    0    0    0    0

$dp13
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    1    1    2    2    1    1    1
upperboundDoF    0    0    0    1    0    1    1    1

$dp14
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    1    1    2    4    3    3    2
upperboundDoF    0    2    2    2    1    1    1    1

$dp15
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    0    0    0    1    0    0    1
upperboundDoF    0    1    2    2    2    2    2    1

$dp16
              [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
negativeDoF      0    0    0    0    1    2    3    3
upperboundDoF    0    1    1    2    0    0    1    1
```

C.3 ANOVA tables for the fitted linear mixed model

C.3.1 The full fitted model with all interaction effects

	Df	Sum Sq	Mean Sq	F value	p value
Method	2	1.6588	0.8294	39.0486	0.0000
Comp	7	1.9804	0.2829	13.3201	0.0000
n	1	26.9483	26.9483	1268.7739	0.0000
relpos	1	1.8462	1.8462	86.9230	0.0000
gamma	1	2.1370	2.1370	100.6126	0.0000
R2	1	15.5574	15.5574	732.4713	0.0000
Method:Comp	14	12.0293	0.8592	40.4544	0.0000
Method:n	2	0.6052	0.3026	14.2472	0.0000
Method:relpos	2	2.2433	1.1216	52.8088	0.0000
Method:gamma	2	2.8302	1.4151	66.6251	0.0000
Method:R2	2	6.4358	3.2179	151.5038	0.0000
Comp:n	7	0.8493	0.1213	5.7125	0.0000
Comp:relpos	7	1.9900	0.2843	13.3848	0.0000
Comp:gamma	7	2.1341	0.3049	14.3536	0.0000
Comp:R2	7	8.5817	1.2260	57.7201	0.0000
n:relpos	1	0.0078	0.0078	0.3650	0.5458
n:gamma	1	1.4813	1.4813	69.7419	0.0000
n:R2	1	14.5864	14.5864	686.7552	0.0000
relpos:gamma	1	0.4192	0.4192	19.7344	0.0000
relpos:R2	1	1.3192	1.3192	62.1117	0.0000
gamma:R2	1	3.2848	3.2848	154.6537	0.0000
Method:Comp:n	14	1.0554	0.0754	3.5491	0.0000
Method:Comp:relpos	14	2.7465	0.1962	9.2364	0.0000
Method:Comp:gamma	14	0.5264	0.0376	1.7701	0.0370
Method:Comp:R2	14	3.1934	0.2281	10.7395	0.0000
Method:n:relpos	2	0.0143	0.0072	0.3373	0.7137
Method:n:gamma	2	1.1714	0.5857	27.5758	0.0000
Method:n:R2	2	0.6202	0.3101	14.5991	0.0000
Method:relpos:gamma	2	0.1301	0.0651	3.0631	0.0468
Method:relpos:R2	2	1.1969	0.5985	28.1764	0.0000
Method:gamma:R2	2	1.2430	0.6215	29.2622	0.0000
Comp:n:relpos	7	0.0227	0.0032	0.1525	0.9937
Comp:n:gamma	7	0.4607	0.0658	3.0989	0.0029
Comp:n:R2	7	0.9630	0.1376	6.4770	0.0000
Comp:relpos:gamma	7	0.3646	0.0521	2.4525	0.0164
Comp:relpos:R2	7	1.6277	0.2325	10.9479	0.0000

Comp:gamma:R2	7	0.1485	0.0212	0.9988	0.4298
n:relpos:gamma	1	0.0349	0.0349	1.6431	0.1999
n:relpos:R2	1	0.0084	0.0084	0.3959	0.5292
n:gamma:R2	1	1.0660	1.0660	50.1886	0.0000
relpos:gamma:R2	1	0.1823	0.1823	8.5838	0.0034
Method:Comp:n:relpos	14	0.2410	0.0172	0.8105	0.6585
Method:Comp:n:gamma	14	0.2311	0.0165	0.7772	0.6953
Method:Comp:n:R2	14	0.3248	0.0232	1.0924	0.3586
Method:Comp:relpos:gamma	14	0.3006	0.0215	1.0108	0.4387
Method:Comp:relpos:R2	14	1.4338	0.1024	4.8220	0.0000
Method:Comp:gamma:R2	14	0.5773	0.0412	1.9416	0.0184
Method:n:relpos:gamma	2	0.0578	0.0289	1.3597	0.2568
Method:n:relpos:R2	2	0.0271	0.0135	0.6380	0.5284
Method:n:gamma:R2	2	0.5035	0.2517	11.8527	0.0000
Method:relpos:gamma:R2	2	0.0317	0.0159	0.7465	0.4741
Comp:n:relpos:gamma	7	0.0174	0.0025	0.1173	0.9972
Comp:n:relpos:R2	7	0.0398	0.0057	0.2674	0.9666
Comp:n:gamma:R2	7	0.2784	0.0398	1.8722	0.0697
Comp:relpos:gamma:R2	7	0.3467	0.0495	2.3318	0.0224
n:relpos:gamma:R2	1	0.0109	0.0109	0.5149	0.4730
Method:Comp:n:relpos:gamma	14	0.0481	0.0034	0.1617	0.9998
Method:Comp:n:relpos:R2	14	0.0547	0.0039	0.1840	0.9996
Method:Comp:n:gamma:R2	14	0.2933	0.0210	0.9865	0.4640
Method:Comp:relpos:gamma:R2	14	0.2781	0.0199	0.9352	0.5193
Method:n:relpos:gamma:R2	2	0.0230	0.0115	0.5403	0.5826
Comp:n:relpos:gamma:R2	7	0.0213	0.0030	0.1434	0.9948
Method:Comp:n:relpos:gamma:R2	14	0.0941	0.0067	0.3163	0.9923

C.3.2 The reduced fitted model

	Df	Sum Sq	Mean Sq	F value	p value
Method	2	1.6588	0.8294	39.3455	0.0000000
Comp	7	1.9804	0.2829	13.4213	0.0000000
n	1	26.9483	26.9483	1278.4210	0.0000000
relpos	1	1.8462	1.8462	87.5839	0.0000000
gamma	1	2.1370	2.1370	101.3776	0.0000000
R2	1	15.5574	15.5574	738.0406	0.0000000
Method:Comp	14	12.0293	0.8592	40.7620	0.0000000
Method:n	2	0.6052	0.3026	14.3555	0.0000006
Method:relpos	2	2.2433	1.1216	53.2104	0.0000000
Method:gamma	2	2.8302	1.4151	67.1316	0.0000000

Method:R2	2	6.4358	3.2179	152.6558	0.0000000
Comp:n	7	0.8493	0.1213	5.7560	0.0000012
Comp:relpos	7	1.9900	0.2843	13.4866	0.0000000
Comp:gamma	7	2.1341	0.3049	14.4627	0.0000000
Comp:R2	7	8.5817	1.2260	58.1590	0.0000000
n:gamma	1	1.4813	1.4813	70.2722	0.0000000
n:R2	1	14.5864	14.5864	691.9770	0.0000000
relpos:gamma	1	0.4192	0.4192	19.8844	0.0000083
relpos:R2	1	1.3192	1.3192	62.5839	0.0000000
gamma:R2	1	3.2848	3.2848	155.8296	0.0000000
Method:Comp:n	14	1.0554	0.0754	3.5761	0.0000062
Method:Comp:relpos	14	2.7465	0.1962	9.3067	0.0000000
Method:Comp:gamma	14	0.5264	0.0376	1.7836	0.0350659
Method:Comp:R2	14	3.1934	0.2281	10.8212	0.0000000
Method:n:gamma	2	1.1714	0.5857	27.7855	0.0000000
Method:n:R2	2	0.6202	0.3101	14.7101	0.0000004
Method:relpos:gamma	2	0.1301	0.0651	3.0864	0.0457214
Method:relpos:R2	2	1.1969	0.5985	28.3907	0.0000000
Method:gamma:R2	2	1.2430	0.6215	29.4847	0.0000000
Comp:n:gamma	7	0.4607	0.0658	3.1225	0.0027132
Comp:n:R2	7	0.9630	0.1376	6.5263	0.0000001
Comp:relpos:gamma	7	0.3646	0.0521	2.4712	0.0156488
Comp:relpos:R2	7	1.6277	0.2325	11.0311	0.0000000
Comp:gamma:R2	7	0.1485	0.0212	1.0064	0.4243251
n:gamma:R2	1	1.0660	1.0660	50.5702	0.0000000
relpos:gamma:R2	1	0.1823	0.1823	8.6490	0.0032820
Method:Comp:relpos:R2	14	1.4338	0.1024	4.8587	0.0000000
Method:Comp:gamma:R2	14	0.5773	0.0412	1.9564	0.0172648
Method:n:gamma:R2	2	0.5035	0.2517	11.9428	0.0000066
Comp:relpos:gamma:R2	7	0.3467	0.0495	2.3495	0.0214297

D Additional plots

D.1 Plots of the averages of the σ^2-estimates

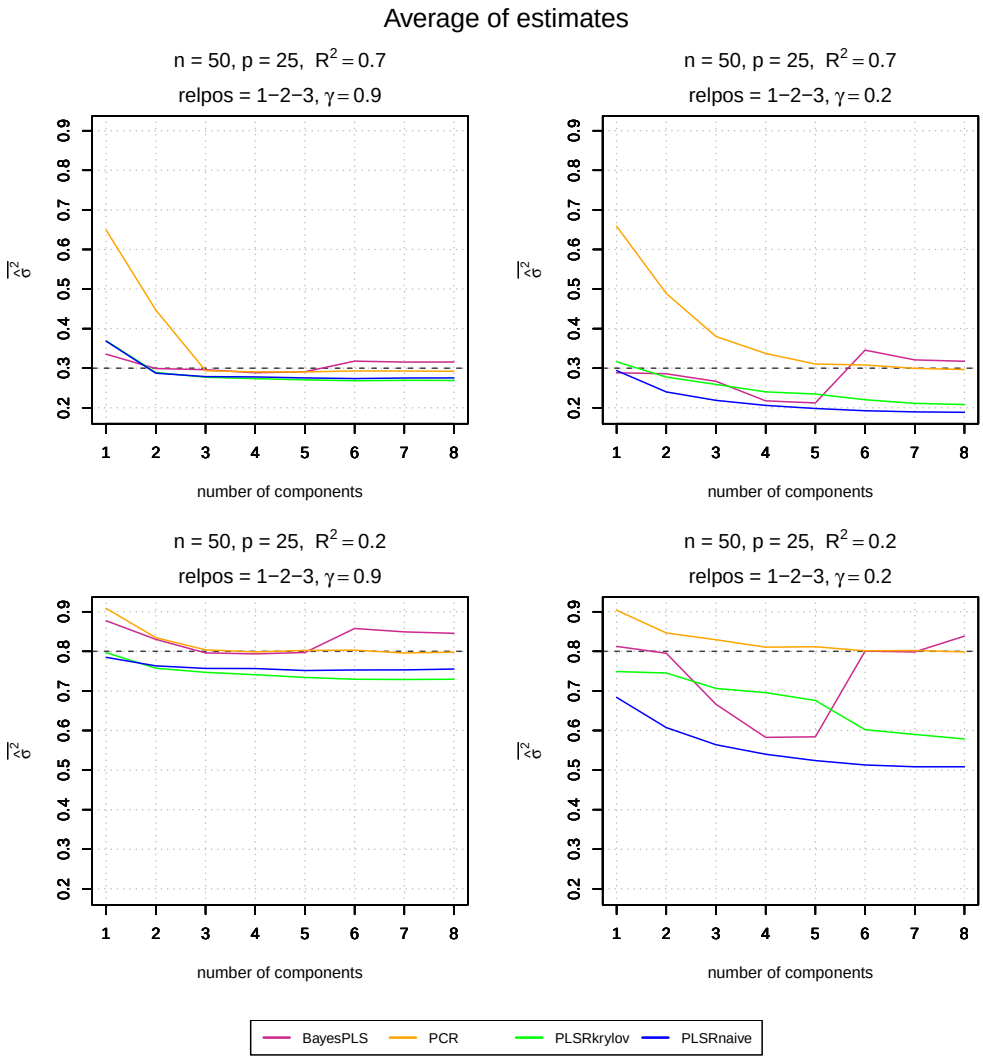


Figure 14: Average estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The dotted line is the true noise variance. The plots belong to dp1, dp2, dp9 and dp10, all having $n = 50$, $p = 25$ and $relpos = \{1, 2, 3\}$.

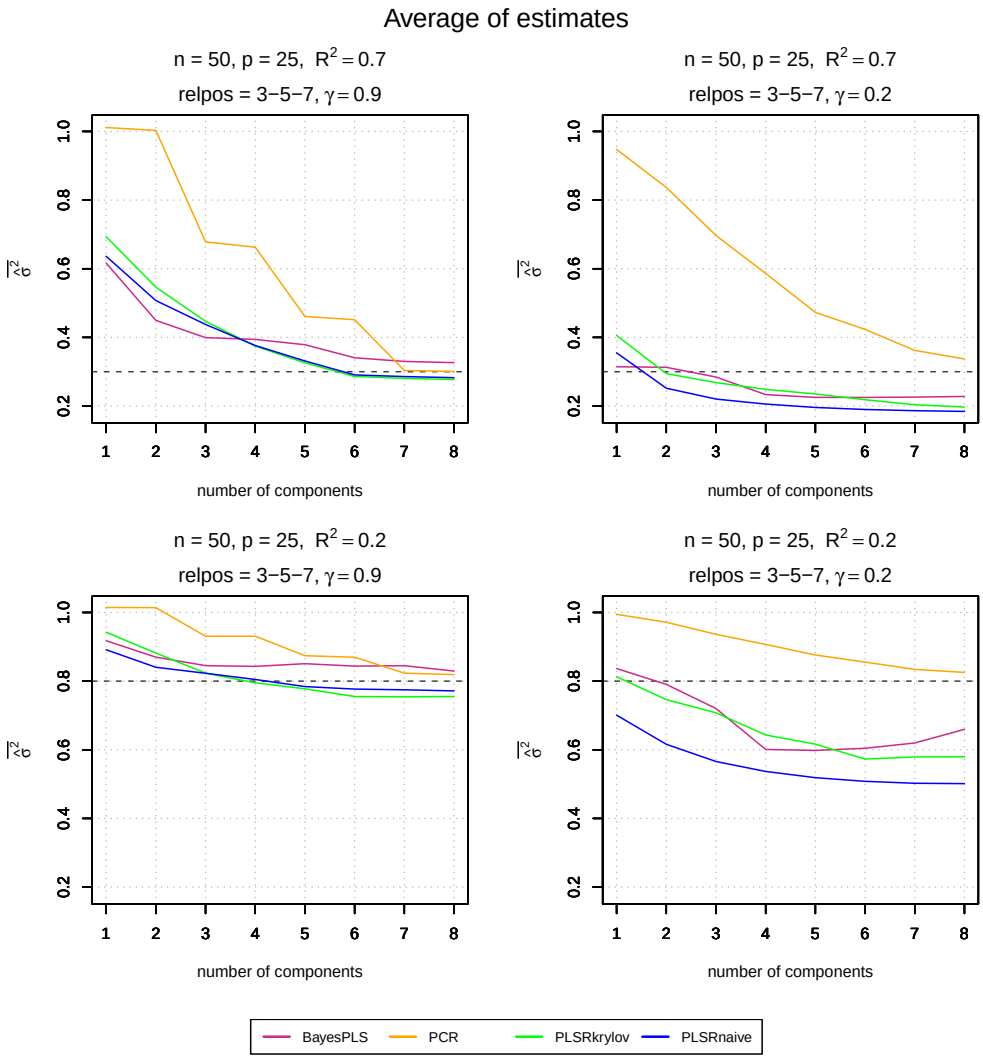


Figure 15: Average estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The dotted line is the true noise variance. The plots belong to dp3, dp4, dp11 and dp12, all having $n = 50$, $p = 25$ and $relpos = \{3, 5, 7\}$.

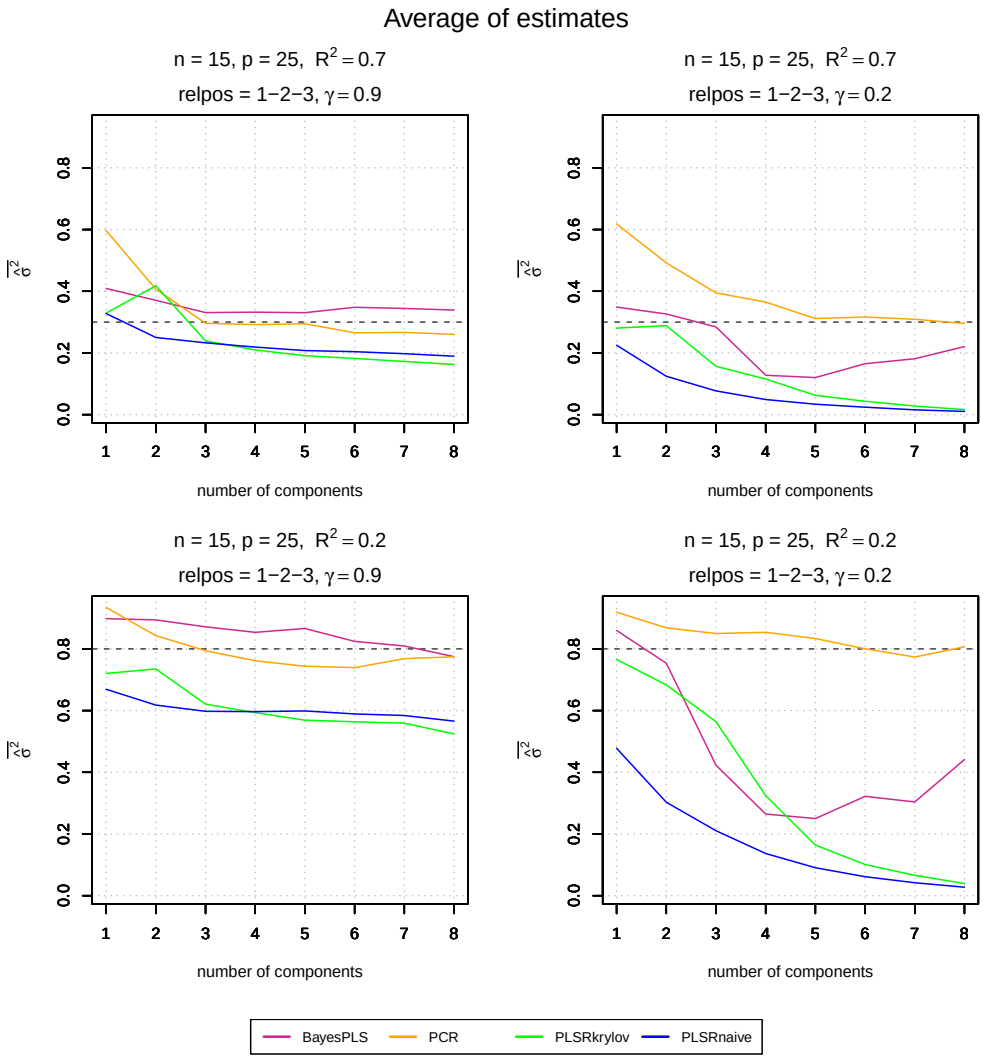


Figure 16: Average estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The dotted line is the true noise variance. The plots belong to dp5, dp6, dp13 and dp14, all having $n = 15$, $p = 25$ and $relpos = \{1, 2, 3\}$.

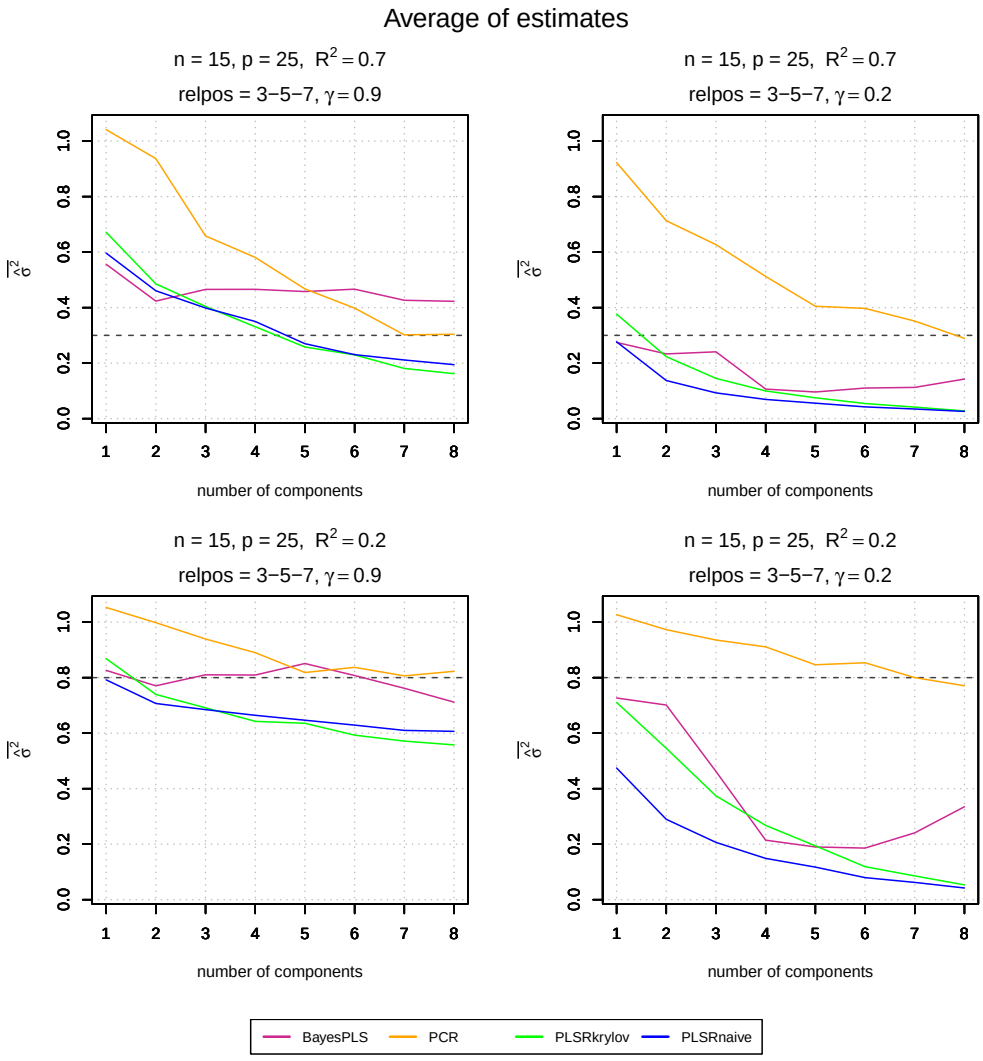


Figure 17: Average estimates vs. number of components. All PLSRkrylov estimates with negative or upper bound DoF have been removed. The dotted line is the true noise variance. The plots belong to dp7, dp8, dp15 and dp16, all having $n = 15$, $p = 25$ and $relpos = \{3, 5, 7\}$.

D.2 Interaction effect plots

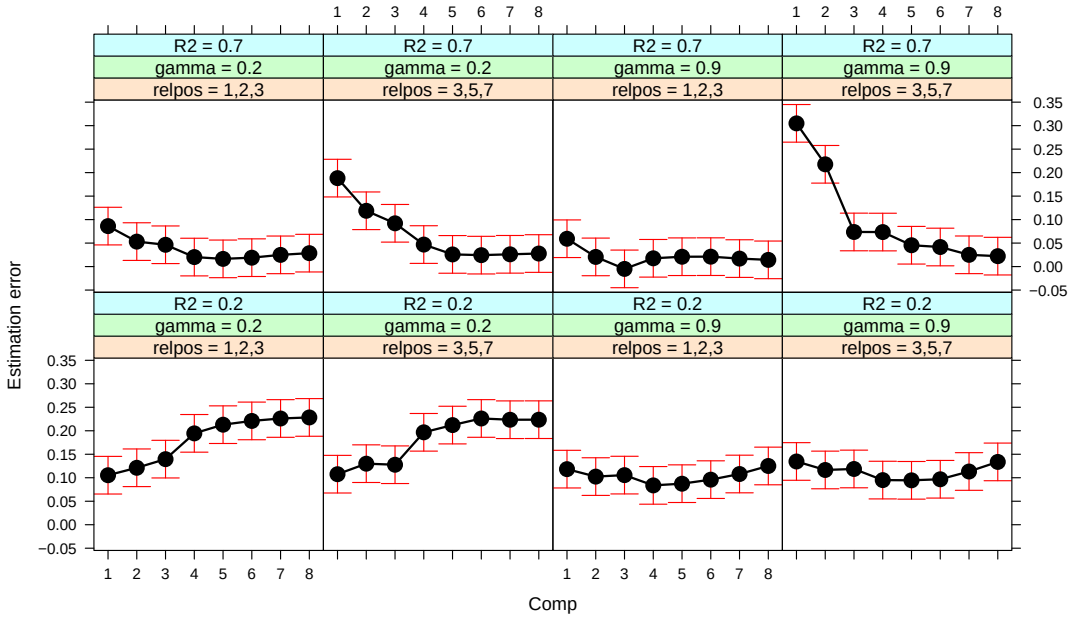


Figure 18: Interaction effect of components, *relpos*, γ and R^2. The red lines are confidence intervals.

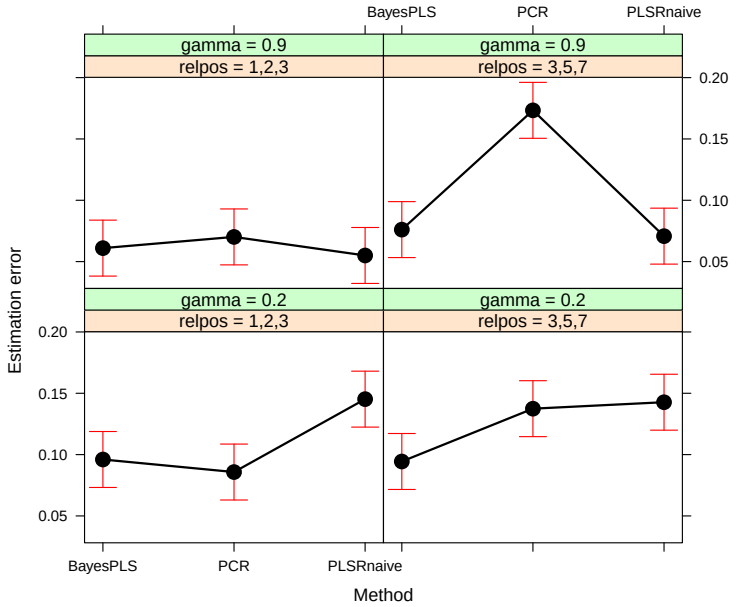


Figure 19: Interaction effect of method, *relpos* and γ. The red lines are confidence intervals.

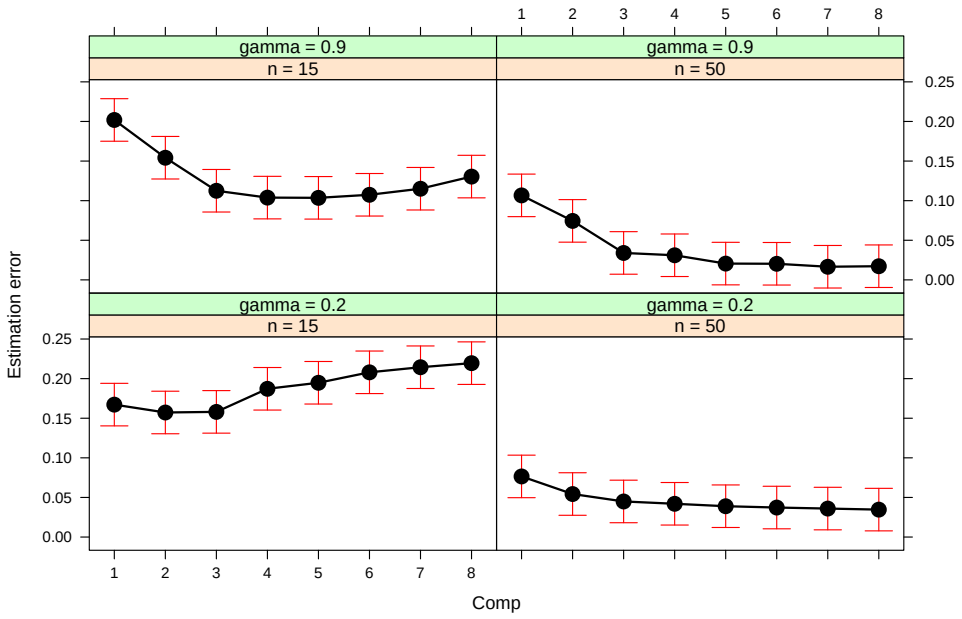


Figure 20: Interaction effect of component, n and γ. The red lines are confidence intervals.

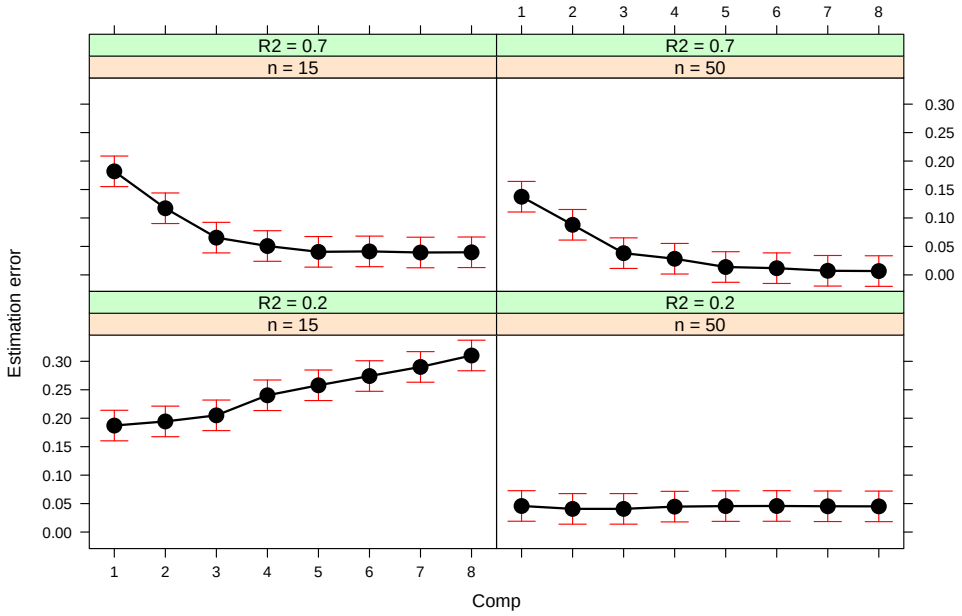


Figure 21: Interaction effect of component, n and R^2. The red lines are confidence intervals.

D.3 True and estimated eigenvalues and covariances

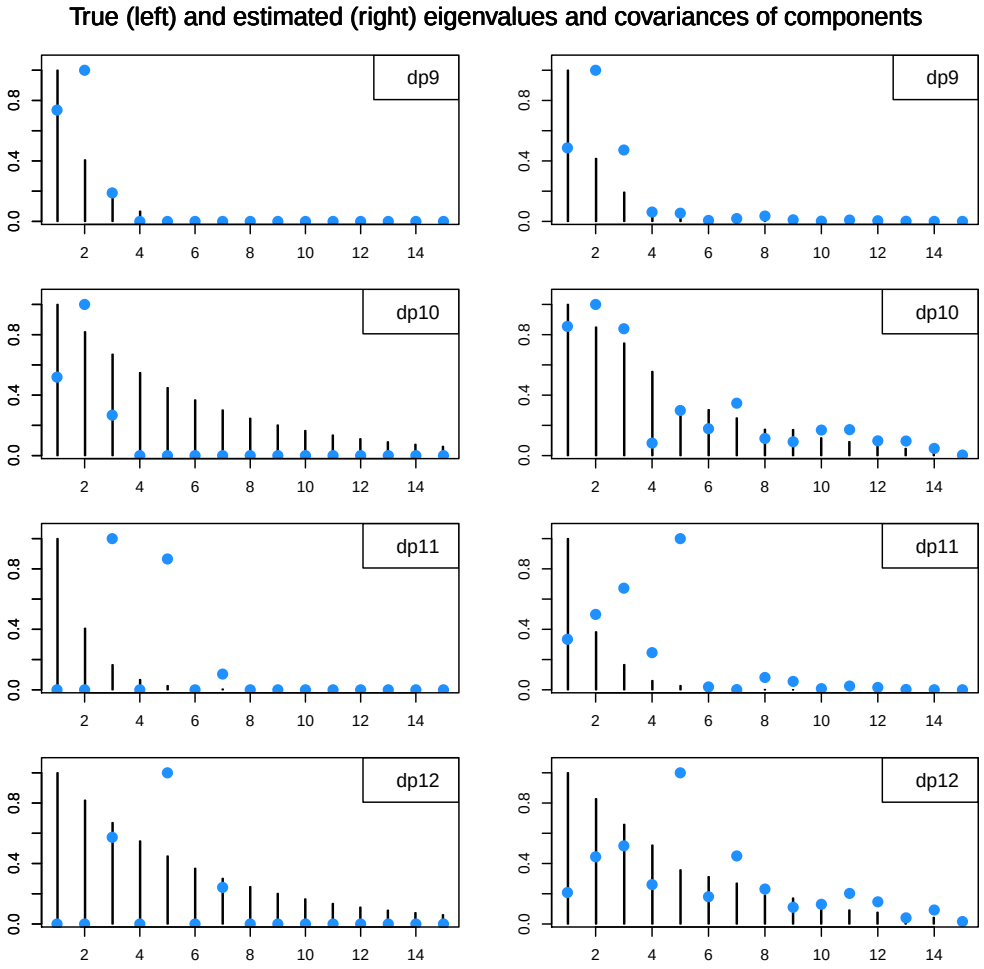


Figure 22: The true eigenvalues (bars) and covariances (blue dots) are illustrated in the plot to the left, and the corresponding estimates are illustrated in the plot to the right. All values are scaled by the largest occuring value. The design points featured here all have $n = 50$ and $R^2 = 0.2$.

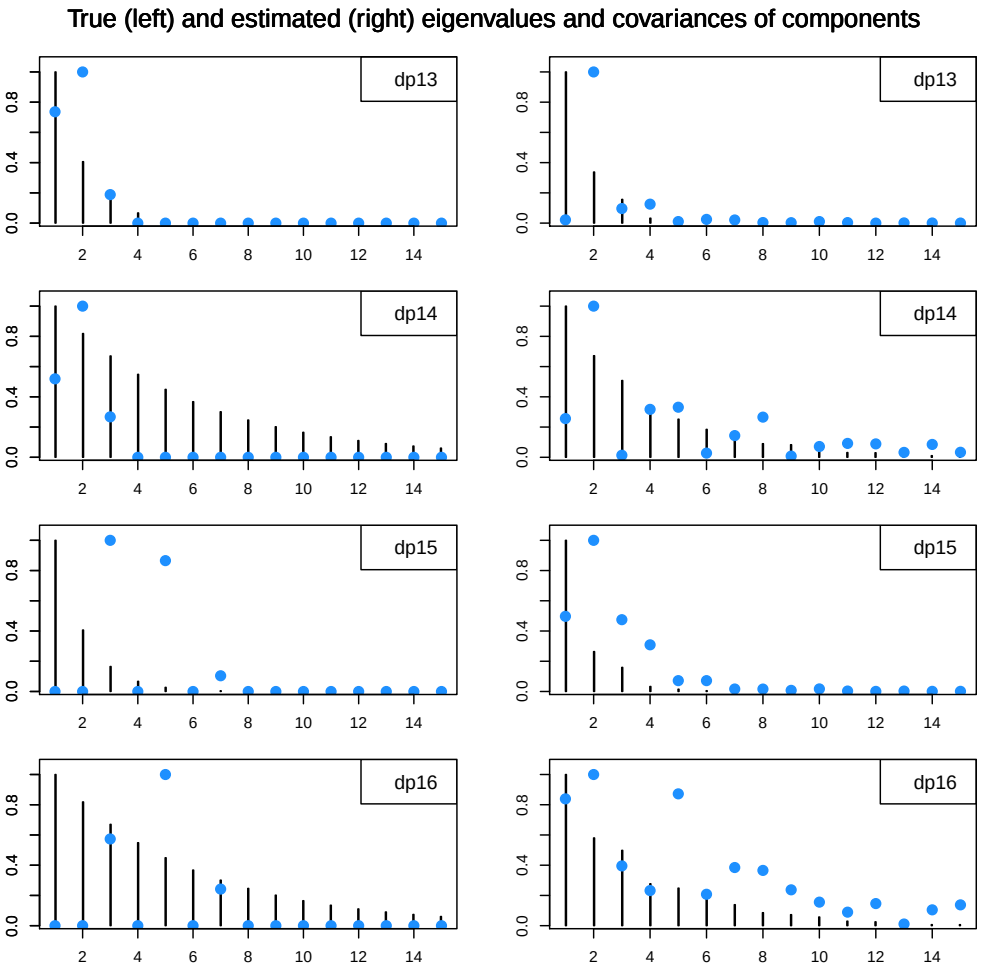
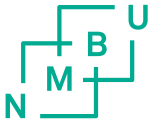


Figure 23: The true eigenvalues (bars) and covariances (blue dots) are illustrated in the plot to the left, and the corresponding estimates are illustrated in the plot to the right. All values are scaled by the largest occuring value. The design points featured here all have $n = 15$ and $R^2 = 0.2$.

E Software

This thesis was written with LaTeX (Sweave) in RStudio version 1.0.143.
All programming and plotting is done in RStudio.
The R scripts are available at `https://github.com/siriskodvin/masterthesis`.



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway