Norges miljø- og biovitenskapelige universitet
Institutt for Kjemi, Bioteknologi og Matvitenskap

Masteroppgave 2014
60 stp

Truncation-PLS for Variable Selection

— a simulation study

Ying Yao

Abstract

Partial least squares (PLS) is a class of statistical methods for multivariate data analysis. In the PLSR algorithm, regression, reducing dimensions and analyzing correlations among variables are simultaneously performed. In the recent 20 years, as high-dimensional data have emerged in large numbers, PLS has been improved and applied in many fields.

In this research, a variable-selection procedure, which is derived from Lenth method, was embedded into PLSR. This algorithm known as Truncation PLS was tried out on several simulated datasets with different designs for the parameters. In order to simulate dataset with different properties, an R package relsim was applied. Another well-known wrapper method Jackknife PLS was also applied to the same datasets as a reference. The purpose of this research is to evaluate these two methods and explore how the properties of dataset will affect the performance of a specific method.

After applying these two PLS methods to different datasets, the value of root mean squared error of prediction (RMSEP) for every parameter setting was obtained through cross validation. RMSEP is a statistic indicating the capability of a model for prediction. In addition, by comparing the beforehand known relevant variables in the datasets, the accuracies of variable selection were calculated to evaluate the capability of a method for variable selection.

Considering the results, both of these two methods performed well and produced satisfying values of RMSEP and accuracy. However, the truncation PLS showed a better capability of dealing with datasets of high multicollinearity in X-variables and smaller variance in its relevant component. Besides, Truncation-PLS method is more efficient than Jackknife PLS from the aspect of calculation and time consumption.

Acknowledgements

I would like to express my deep gratitude to professor Solve Sæbø and associate professor Trygve Almøy, my research supervisors, for their patient guidance, enthusiastic encouragement, instructive advice, and helpful suggestions on this thesis.

As a multi-disciplinary student with a great interest in statistics, I did not have much background in math and statistics at the beginning. Solve encouraged me to pursue my interests and always be helpful. With his helps, I have improved myself much not only in statistics, but also in scientific writing, programming, and even Norwegian. His realistic attitude, professionalism, dedication, and assiduous work style will consistently inspire me in my future life.

I am also deeply indebted to all the other teachers in the biostatistics group for their direct and indirect help to me. It is a great opportunity for me to study in this professional group for my Master's degree at NMBU.

Special thanks also go to my parents and husband for their continuous supports and encouragement.

Contents

CHAPTER 1 Introduction

Various types of high-dimensional data have appeared in the recent 20 years, such as multimedia graphics video data, time series data, and huge amount of measurement information generated by modern analytical instruments. Especially in the research of bioinformatics, following the development of some high throughput measuring technologies, exponential growth of the amount of nucleotide data leads to much more variables, in contrast to scant number of observations. Therefore, datasets become “wider” and “wider”. The biggest problem in dealing with high-dimensional data is commonly referred to as "the curse of dimensionality" problem, which indicates that when the dimension rises, complexity and cost of data analysis grow at an exponential rate. Moreover, caused by increasing the probability of including irrelevant variables into model, it may become more difficult to explain a complex system with high-dimensional data. Therefore, it is a great challenge to utilize the data effectively in practice.

Multivariate regression models are widely employed to explore possible relationships between responses and variables. Some classic methods, such as least squares regression and hierarchical classification methods, may have some difficulties in dealing with high-dimensional data. The increment of dimensions will lead to enormous amount of computation; the number of samples may be not sufficient to meet the requirement of these multivariate methods.

In the situation where we have many predictor variables but a small number of observations, even if some variables are uncorrelated in the population, they might seemingly appear correlated in small samples. Thus, a problem of multicollinearity may arise. As a result of multicollinearity, some statistics are difficult to achieve asymptotic, and hence give inaccurate parameter estimates.

And worse still, least squares regression sometimes fails to estimate parameters in multivariate model if the number of samples “n” is smaller than the number of predictor variables “p”.

PLS is a statistical method for multivariate analysis. With relatively less constraints of variables, PLS is suitable in many situations where classic low-dimensional method cannot be applied, such as when the number of observations is less than the number of predictor variables or some variables are highly correlated. Consequently, PLS attracts more and more attentions of scientists and statisticians. In the PLSR algorithm, regression, reducing dimensions, and analyzing correlations among variables are simultaneously performed. However, without variable selection, PLSR model may not be stable for prediction and it cannot be easily interpreted.(Tahir Mehmood 2012)

In this thesis, we applied a truncation based variable selection method in the procedure of PLSR algorithm, which was introduced in (Liland et al., 2013). The algorithm was tried out on simulated data. In statistical inferences, people mine the features of the data by different methods. Data simulation is a critical tool to evaluate methods. It provides us a way not only to understand the dynamic processing of these methods but also to check the variety of inferential results against the true information. Different types of datasets containing Y and X are simulated, where some variables in X are relevant to Y while others are irrelevant. On that way, within beforehand known information of the simulated data, we can evaluate different methods by comparing the estimated parameters with the real ones. Vice versa, by applying a method to different types of simulated datasets, we can explore types of datasets to which the specific method performs well.

CHAPTER 2 Background

2.1 Linear Model

2.1.1 General Linear Model

Generalized linear models were formulated by John Nelder and Robert Wedderburn as a way of unifying various other statistical models, including linear regression, logistic regression, and Poisson regression (Nelder and Wedderbu.Rw, 1972). In statistical analyses, the General Linear Model (GLM) is the foundation for various methods, such as analysis of variance (ANOVA), regression analysis, and many of the multivariate methods including least square method, principal component analysis (PCA), and partial least squares (PLS).

Consider n observations noted $Z_i = (X_i, Y_i)$, in which $Y_i \in R, X_i \in R^p, i = 1, \dots, n$

Further, $Y = (y_1, \dots, y_n)^T \in R^n$ is considered as a continuing response.

$X = (X_1, \dots, X_n)^T$ is a matrix with a dimension $n \times p$.

The general linear model (GLM) might be written as

$$Y = \beta_0 + X\beta + E \qquad (1)$$

The distribution of the error term of every observation is often assumed to be the same in GLM, so that E is a matrix containing errors following a normal distribution with a mean 0 and a variance σ^2. $\beta = (\beta_1, \dots, \beta_p)^T$ is a vector of coefficients for X_i $(i = 1, \dots, n)$. β_0 in this model is an intercept. It might be interpreted as the expected value of Y when all the variables in X are setting to 0, which could be unrealistic sometimes.

Therefore, to make the computation and interpretation easier, an alternative way is to center the data by subtracting the mean of every variable from X and Y. In such a way, the intercept β_0 is equal to 0 and can be ignored in the model. Within the

centered data Y_0 and X_0, the model can be expressed in the form

$$Y_0 = X_0\beta + E \qquad (2)$$

The coefficient vector β in form (2) is the same as β in form (1). And the expected β_0 in form (1) should be exactly the same as the mean of values observed in Y.

2.1.2 Ordinary least squares (OLS)

The least squares method is a standard approach to estimate β in a linear regression model. By applying least squares method, the solution should be found to minimize the sum of the squares of the residuals SSR. A residual is defined as the difference between an observed value and the fitted value provided by the model.

$$SSR = (Y - X\beta)^T(Y - X\beta) = \sum_{i=1}^{n}(Y_i - X_i\beta)^2$$

In least squares regressions, estimation of β is calculated by

$$\tilde{\beta} = (X^TX)^{-1}X^TY$$

The solution gives the best approximation of the data. However, least squares regression requires that X^TX to be positive definite, otherwise it fails to estimate parameters in multivariate model in a situation that the number of samples "n" is smaller than the number of predictor variables "p".

Consider the n × p matrix X of sample data. The rank of X is at most the minimum of n and p, thus n in n<p cases. Therefore the rank of p × p matrix X^TX won't be larger than n, which is the rank of X. In respect that sample covariance matrix X^TX is singular and non-invertible if n<p, least squares regression will lose the unique solution and fail to estimate parameters then.

2.2 Variable selection

In statistical modeling and inference, variable selection is an elementary step. The basic logic of these methods is to find an easily interpretable model with a set of predictor variables, which gives a good fit to data. Moreover, the model may be applied for prediction. It has been shown by many researches that including non-informative variables in a model may harm the precision of estimation and prediction(A. J. Miller, 2002). Some serious problems could be brought in by including irrelevant variables, such as colinearity and over-fitting of models.

From the 1970s different methods of variable selection have been proposed. The frequently used methods may be classified into four categories: all subset method, stepwise methods, coefficient shrinkage methods and projection methods.

2.2.1 All Subset Method

In order to select a best subset of predictors from all candidates of predictor variables, All Subset Method compares all the possible combinations of predictors. Several evaluation criteria can be used to compare the candidate models, such as R^2, PRESS, Mallow's Cp, and AIC. Although the method can guarantee the best subset, sometimes it involves too much computation and lead to long computational time. Suppose the number of predictor variables is p, the number of all possible subsets is 2^p, which could be a huge number when p is large. Therefore, subsets method might be applied properly in the cases with a small p.

2.2.2 Stepwise Method

2.2.2.1 Forward Selection

Forward Selection method starts with a model of size 0 and proceeds by adding variables that fulfill a defined criterion. Typically the added variable at each step is the one that minimizes SSE. This can be evaluated also by F- test, defined by

$$F_{in} = \frac{SSE_p - SSE_{p+1}}{SSE_{p+1}/(n-p-2)}$$

where SSE_p and SSE_{p+1} are the sum of squares for the error of the models with p and

p+1 variables respectively. F_{in} is used as a stop criterion, corresponding to the probability α, with the freedom of one degree for the numerator and $(n-p-2)$ for the denominator.

2.2.2.2 Backward Elimination

Backward Elimination method proceeds in an opposite way. It starts from a model of size p where p is the total number of variables. Non-relevant variables will be eliminated step by step. In this case, the detected variable is usually the one that gives a minimum increase in SSE. Analogy to the Forward Selection method might be evaluated by F-test, defined by

$$F_{out} = \frac{SSE_{p-1} - SSE_p}{SSE_p/(n-p-1)}$$

where SSE_{p-1} and SSE_p are the sum of squares for the error of the models with p and p-1 variables respectively. F_{out} is used as a stop criterion, corresponding to the probability α, with the freedom of one degree for the numerator and $(n-p-1)$ for the denominator.

2.2.2.3 Stepwise regression

The original algorithm was later improved by Efroymson in 1960 by combining Forward Selection and Backward Elimination(Efroymson, 1960). It starts with Forward Selection. After each variable is added to the model, a test should be made to check if any of the selected variables could be eliminated without largely increasing the SSE. The variables already been selected in the model could become insignificant after adding a new one which correlated with them. The test here might be F test as well as a test based on other criteria such as R^2. To avoid an infinite loop, the significant level for adding variables should be less than the one for eliminating.

2.2.3 Coefficient shrinkage method

Researchers have proposed some methods that are able to perform both regression and variable selection simultaneously through coefficient shrinkage. In contrast to the discrete process of subset methods, variable selection methods based on coefficient shrinkage are more continuous. Depending on few parameters and

without consuming many degrees of the freedom in the selection process, coefficient shrinkage methods avoid high variability.

2.2.3.1 Lasso (Least Absolute Shrinkage and Selection Operator)

By Lasso, as was introduced in (Tibshirani, 2011), we can select some β that minimize the following function as our estimator

$$\sum_{i=1}^{N}(y_i - \boldsymbol{x_i\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

The first part of this function shows the fitness of model, while the second part can be considered as a penalty term. The general idea is to shrink coefficients to some level that some of them are forced to be 0. λ is a tuning parameter which can be used to decide the model complexity, and hence the number of variables to be excluded from the model. Unlike variable selection methods that are based on subset, Lasso selects variables through a relatively mild way and make the model more stable.

2.2.3.2 Ridge Regression

By Ridge regression, as introduced in(Rubio and Firinguetti, 2002), we can select some β that minimize the following function

$$\sum_{i=1}^{N}(y_i - x_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|^2$$

λ is a tuning parameter ,which decides that to what extent we will shrink the coefficients. But it does not force any coefficient to 0 as Lasso. By penalizing the size of the regression coefficients by $\lambda\sum_{j=1}^{p}|\beta_j|^2$, Ridge regression has an advantage of dealing with the multicollinearity problem. We mention Ridge here as the motivator of the next method.

2.2.3.3 Elastic net

In genomic data, genome sequences are more likely correlated because some of them

tend to operate in molecular pathways. Among a set of strong but correlated variables, the lasso penalty is somewhat indifferent (Trevor Hastie, 2008). It tends to select only one of them but to ignore the others.

Elastic net is another regression method based on coefficient shrinkage, as introduced in (Zou and Hastie, 2005), the estimates of β should minimize the following function.

$$\sum_{i=1}^{N} |y_i - x_i\beta|^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j|^2 + \lambda_2 \sum_{j=1}^{p} |\beta_j|$$

Just as the above, the first term of this function shows the fitness of model. But as a compromise between Lasso regression and Ridge regression, elastic net employs both $\sum_{j=1}^{p} |\beta_j|^2$ and $\sum_{j=1}^{p} |\beta_j|$ as penalty terms to regularize their parameters. An equivalent way to write the penalty term is

$$\lambda \sum_{j=1}^{p} \left(\alpha|\beta_j| + (1-\alpha)|\beta_j|^2\right)$$

The value of λ and α can be chosen by cross-validation. With keeping the first term of penalty function, elastic net shares the same feature of variable selection as Lasso. In other words, some of coefficients can be forced to 0. In contrast to Lasso regression, the second term of the function encourages to shrink the coefficients of highly correlated variables meanwhile.

2.2.4 Projection methods

In some situations, we have massive number of variables and some of them are believed to be collinear. PCR and PLSR represent a class of methods based on projections to latent components. The philosophy of these methods is to produce latent variables by projection, which is designed to optimally describe these correlated ones in the original dataset. On that way, projection methods can be used as dimension reduction technique coupled with a regression model.

2.2.4.1 Principal component regression (PCR)

The names of PCR stems from the fact that we use PCA (Principal Component Analysis) to extract the orthogonal components from the X dataset (Jolliffe, 1982). The concept of principal components implies the most meaningful basis that represents the data. In every step, the component corresponding to the largest eigenvalue of the covariance matrix of residuals is extracted. This procedure guarantees that the component contains the largest variance in the remaining data. In the practical implementations of PCA, the components are sorted in according to their variance information. Then a dataset can be represented well with A components ($A \leq p$). Thus, the dimension of data is reduced at the cost of little information loss. In many cases, these components present a systematic way to understand variables.

Several numerical algorithms lead to the same PCA solution. Instead of presenting it in the most common way, we choose to explain PCA in an alternative algorithm as follows, which is most similar to PLSR.

At first X and y is centered into

$$y_0 = y - 1\bar{y}$$
$$X_0 = X - 1\overline{X}$$

where 1 is a vector of ones which has the same length of y ; $\bar{y}$ is the mean value of y; and $\overline{X}$ is the row vector containing the average values for each of the columns in X.

Suppose the number of components for prediction is chosen to be A ($A \leq p$). For a=1,2…A, the following algorithm are iterated for every component.

The loading-weight vector w_a is defined as the eigenvector with the largest eigenvalue of covariance matrix of X_{a-1}.

The component t_a, which extracts the maximum variance from matrix X_{a-1} is defined as $t_a = X_{a-1}w_a$. In other words, it satisfies the following function.

$$w_a = \arg\max\{\|X_{a-1}w_a\|^2\} = \arg\max\{\|t_a\|^2\} = \operatorname{argmax}\left\{\left(\sum_{i=1}^{n} {t_{ai}}^2\right)\right\}$$

The residual matrix X_a can be found by subtracting the kth principal components from X_{a-1}:

$$X_a = X_{a-1} - t_a X_{a-1} t_a (t_a{}^T t_a)^{-1}$$

In the practical implementations of PCA, we iterate these procedures for A times ($A \leq p$), until they produce a satisfying small residual X_a.

After extracting the orthogonal components, the PCR is obtained by regressing y on these components T={t_a(a=1,2…A)}. The regression coefficients for model $y = XB + E$ is defined as

$$B = P^T Q$$

X-loadings P can be calculated as least squares solution of the model X=TP+E,

$$P = (T^T T)^{-1} T^T X$$

Y-loadings Q can be calculated as least squares solution of the model y=TQ+F,

$$Q = (T^T T)^{-1} T^T y$$

2.2.4.2 Partial least squares regression (PLSR)

Partial least squares was firstly introduced by Herman Wold (Wold., 1973, Wold, 1966), then developed further by his son Svante Wold who applied it to regression (Wold et al., 1984). Although PLS was originally applied in econometrics and social sciences, after being improved by many researchers in these years (Helland, 1988, Martens H, 1989), a variety of PLS methods are more widely used in many other fields, such as bioinformatics, economics, and pharmaceutical science. In chemometrics, PLSR was used as a standard multivariate modeling tool.

Generally, by giving a loading-weight to each variable, PLS method extracts some orthogonal components, noted as t and u, from dataset X and Y respectively with the

following constraints in every iterative process: (1) optimally present the variance information in X and Y respectively; (2) maximize the covariance between t and u. Then X is regressed on t by least square regression; Y is regressed on u by least square regression. The above procedures are repeated until satisfied residual matrixes are obtained. These components as latent variables are used in regression. Thus dimension reduction is performed at the same time as regression. Moreover, by employing orthogonal components in regression, PLS avoids collinear problem in building models effectively. PLS can also be applied for discrimination as in (Barker and Rayens, 2003).

Among variety of algorithms of PLS, the most commonly used algorithm with orthogonal scores is presented as follows. Suppose y is a single response vector.
At first, X and y is centered into

$$\mathrm{y}_0 = \mathrm{y} - 1\bar{\mathrm{y}}$$
$$\mathrm{X}_0 = \mathrm{X} - 1\overline{\mathrm{X}}$$

A ($A \leq p$) is the number of components chosen for regression. For a=1, 2…A, the following algorithm are iterated for every component.

(1) Compute the loading-weights vector $\mathrm{w_a}$

$$w_{\mathrm{a}} = X_{a-1}^{T} y_{a-1}$$

and scale it into a vector with length equal to 1

$$w_a^* = \frac{\mathrm{w_a}}{\|\mathrm{w_a}\|} = \mathrm{w_a}/\sqrt{\mathrm{w_a^T w_a}}$$

(2) Compute the score vector $\mathrm{t_a}$

$$\mathrm{t_a} = \mathrm{X_{a-1}} w_a^*$$

(3) Compute the X-loadings p_{a} by regressing the variables in $\mathrm{X}_{\mathrm{a}-1}$ on the score vector t_{a}

$$p_{\mathrm{a}} = X_{a-1}^{T} t_{\mathrm{a}} (t_a^T t_{\mathrm{a}})^{-1}$$

Compute the Y-loadings p_{a} by regressing the variables in $y_{\mathrm{a}-1}$ on the score vector t_{a}

$$q_{\mathrm{a}} = y_{a-1}^{T} t_{\mathrm{a}} (t_a^T t_{\mathrm{a}})^{-1}$$

(4) Subtract the information explained by the ath component to compute the residual matrices X_{a}, y_{a}

$$\mathrm{X}_{\mathrm{a}} = \mathrm{X}_{\mathrm{a}-1} - t_{\mathrm{a}} p_a^T$$
$$\mathrm{y}_{\mathrm{a}} = \mathrm{y}_{\mathrm{a}-1} - t_{\mathrm{a}} q_{\mathrm{a}}$$

In the regression procedure, we save the loading weights, scores, and X-loadings above into matrices or vectors: $\mathrm{W} = \{\mathrm{w}_1, \mathrm{w}_2 \dots \mathrm{w}_A\}$, $P = \{p_1, p_2 \dots p_A\}$, $\mathrm{Q} = \{q_1, q_2 \dots q_A\}$. Finally, the vector β of estimated regression coefficients for model $y = \beta_0 + \beta^T X + \varepsilon$ can be computed by

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

The intercept β_0 can be estimated by

$$\widehat{\beta_0} = \bar{y} - \hat{\beta}^T \bar{X}$$

PLSR can also be generalized in the situation with multiple responses in the Y matrix. The algorithm is presented as follows:

Center the matrices X and Y into

$$Y_0 = \mathrm{Y} - 1\overline{\mathrm{Y}}$$
$$\mathrm{X}_0 = \mathrm{X} - 1\overline{\mathrm{X}}$$

where $\overline{\mathrm{Y}}$ is now the row vector containing the average values for every one of the columns in Y. The number of components for regression is A ($A \leq p$). For a=1, 2…A, the following algorithm is iterated for every component.

(1) Initialize u_a to the column of Y_{a-1}matrix of the largest variance.

(2) Compute the loading-weights vector $\mathrm{w_a}$

$$\mathrm{w_a} = X_{a-1}^T u_a$$

and scale it into a vector with length equal to 1

$$w_a^* = \frac{\mathrm{w_a}}{\|\mathrm{w_a}\|} = \mathrm{w_a}/\sqrt{\mathrm{w_a^T w_a}}$$

(3) Compute the score vector $\mathrm{t_a}$

$$\mathrm{t_a} = \mathrm{X_{a-1}} w_a^*$$

(4) Compute the X-loadings p_a by regressing the variables in $\mathrm{X_{a-1}}$ on the score vector t_a

$$p_\mathrm{a} = X_{a-1}^T t_\mathrm{a} (t_a^T t_\mathrm{a})^{-1}$$

Compute the Y-loadings q_a by regressing the variables in $y_{\mathrm{a-1}}$ on the score vector t_a

$$q_\mathrm{a} = Y_{a-1}^T t_\mathrm{a} (t_a^T t_\mathrm{a})^{-1}$$

(5) Update u_a by regressing the variables in $y_{\mathrm{a-1}}$ on the Y-loadings q_a

$$u_a = Y_{a-1}^T q_\mathrm{a} (q_a^T q_\mathrm{a})^{-1}$$

(6) Repeat the above procedures (1)-(4) until the score vector t_a converges. The goal is to maximize the covariance between u_a and t_a.

(7) Subtract the variance caused by the ath component to compute the residual matrices $\mathrm{X_a}$, Y_a

$$\mathrm{X_a} = \mathrm{X_{a-1}} - t_\mathrm{a} p_a^T$$
$$Y_\mathrm{a} = \mathrm{Y_{a-1}} - t_\mathrm{a} q_a^T$$

In the regression procedure, we save the loading weights, scores, and X-loadings above into matrices or vectors: $W = \{w_1, w_2 \dots w_A\}$, $P = \{p_1, p_2 \dots p_A\}$, $Q = \{q_1, q_2 \dots q_A\}$. Finally, the matrix B of regression coefficients for model $y = \beta_0 + B^T X + \varepsilon$ with a components can be estimated by

$$\hat{B} = W(P^T W)^{-1} Q^T$$

and the intercept vector can be estimated by

$$\widehat{\beta_0} = \bar{Y} - \hat{B}^T \bar{X}$$

2.2.4.3 Variable selection in PLSR

Although PLSR has an inherent process of assigning different weights to variables, it does not exclude the directions spanned by noisy variables. It was shown by (Chun and Keles, 2010) that in a situation with large p and small n, PLSR may fail to give asymptotic consistency estimators for responses, thus it produces a predicted response with large variance. Moreover, without variable selection, regression models in PLSR may not be easily interpretable. An ideal model should not only perform well in prediction, but also provide an understanding of how the system works. Therefore, varieties of variable selection methods integrated with PLS are applied in practice.

In (Tahir Mehmood 2012), these methods were presented in 3 categories such as filter methods, wrapper methods, and embedded methods.

Filter methods use the output of the PLS algorithm to select variables. Variables are selected based on the magnitude of the filter measures. The filter measures could for instance be loading weight w_a, PLS regression coefficients $\widehat{\beta}$ and variable importance on projection (VIP). The VIP measure is defined as

$$v_j = \sqrt{p \sum_{a=1}^{A} \left[SS_a \left(w_{aj} / \|w_a\|^2\right)\right] \Big/ \sum_{a=1}^{A} (SS_a)}$$

where SS_a is the variance in y explained by ath component, which could be expressed as $q_a^2 t_a^T t_a$, and w_{aj} is the j'th element in the loading-vector w_a. Hence, $\left(w_{aj}/\|w_a\|^2\right)$ represents the contribution of x_j in the a'th component. Generally, if v_j is larger than 1, x_j is considered to be an important explanatory variable.

Wrapper methods are generally based on iterating procedures between model fitting and variable selection. The variables, which are selected by filter method, are recycled in next PLSR procedure to get an optimal variable set. Some of these methods contain random procedures such as the Genetic algorithm combined with PLS regression which was introduced by (K. Hasegawa, 1997), and Monte-carlo variable elimination with PLS (Han et al., 2008). Another very popular wrapper method is the Jackknife selection method.

Embedded methods nest the variable selection to the PLSR algorithm. During the iterations in PLSR, variables are selected for every component. The best-known methods in this category are interactive variable selection(Lindgren et al., 1994, Lindgren et al., 1995), soft-threshold PLS(Saebo et al., 2008), sparse-PLS(Le Cao et al., 2008), and powered PLS(Indahl et al., 2009).

2.3 Evaluation criteria for model comparison

Various criteria could be used for selecting variables and comparing models. In practice, a criterion should be chosen according to the purpose of research. Here a short overview of common criteria is presented.

2.3.1 Likelihood based criteria

AIC

The Akaike information criterion (AIC) was proposed by Akaike in 1974. (Akaike, 1974), under the name of "an information criterion".
In the general case, the AIC is given by

$$AIC = -2\ln(L) + 2k$$

where k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model. AIC not only rewards the goodness of fit by $-2\ln(L)$, but also includes a penalty which discourages overfitting by increasing AIC as the number of estimated parameters increase. Given a set of candidate models for the data, the preferred model is the one with the minimum AIC value.

BIC

Bayesian information criterion (BIC) is a criterion which was developed by Gideon E. Schwarz, who gave a Bayesian argument for adopting it.(Schwarz, 1978)

The formula for the BIC is:

$$BIC = -2\ln(L) + k\ln(n)$$

where k is the number of parameters in the statistical model, n is the number of observations, and L is the maximized value of the likelihood function for the estimated model.

Both BIC and AIC solved the overfitting problem by introducing a penalty term for the number of parameters in the model. The penalty term increasing with n is larger in BIC than the one in AIC with $\ln(n) \geq 7, n \geq 2$.

2.3.2 R^2 or adjusted R^2

The coefficient of determination is given by

$$R^2 = 1 - \frac{SSE}{SST}$$

Theoretically, models with larger R^2 should be preferred. Since we all know SSE always will decrease when we include more predictors, we should add predictor

variables until R^2 increases significantly.

Adjusted R^2 is a modification of R^2 which is given by

$$R_a^2 = 1 - (n-1)\frac{MSE}{SST}$$

It adjusts for the number of variables in a model. Shown by the formula, this criterion will select the model with smallest MSE. Since MSE, unlike SSE, can increase or decrease while we include more variables, R_a^2 will increase only if the new term improves the model significantly. R_a^2 is always less than or equal to R^2.

2.3.3 Prediction based criteria

A common purpose of modeling is to predict the future value of Y. Therefore there are some criteria based on the error of prediction. The following statistics are used for measuring the error of prediction of a model.

PRESS and RMSEP

The prediction error sum of squares (PRESS), is given by

$$\text{PRESS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$$

After a fitted model is tried out on a test-dataset of n observations, y_i is the actual value of y for the i-th observation in test dataset; $\hat{y}_i$ is the predicted value for y_i with the model under evaluation.

The root mean squared error of prediction (RMSEP):

$$\text{RMSEP} = \sqrt{\frac{\text{PRESS}}{n}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}}$$

Both above statistics give more or less the same information. In practice, RMSEP values are preferred than PRESS, because RMSEP is in the same units as the y, thus it's easier to be interpreted.

2.3.4 Mallows's C_p

Mallows proposed the statistic as a criterion for selecting among many alternative subset regressions (Mallows, 1973). Mallows's C_p is a statistic given by

$$C_p = \frac{SSE_p}{MSE_k} + 2(p+1) - n$$

where SSE_p is the mean squared prediction error for the model with p regressors, calculated by

$$\mathrm{SSE_p} = \sum_{\mathrm{i=1}}^{\mathrm{n}} (\mathrm{Y_i} - \mathrm{Y_{pi}})^2$$

p is the number of predictor variable in the subset model, n is the number of observations, and MSE_k is the MSE for the full model. It is suggested that one should choose a subset that has a smallest C_p. In an ideal state, the value of C_p is expected to approaching p. (Daniel, 1980)

2.4 Validation

Prediction error obtained by residuals of a regression model may be over-optimistic, since we actually use the same dataset to train model and evaluate residuals. Instead, a validation should be performed to qualify the model we assumed. In this validation step, predicted values with the model under evaluation are tested independently with a test dataset, which is different from those for training the model. In practice, we can either choose a K-fold Cross Validation (K-CV) or Leave-One-Out Cross Validation (LOO-CV). In K-CV, a dataset should be divided into K groups. Every of them is used as

testing set once and others as training set. In general, a smaller K will produce a relatively poor model estimate but a smaller variance of prediction error. On the contrary, a larger K will lead to a better estimate with smaller bias but potentially higher variance of prediction error. When K is approaching N (total number of observations), K-fold Cross Validation is getting closer to the limit case: LOO-CV. In LOO-CV, every observation in the data set is used as testing set once, and others as training set. In both K-CV and LOO-CV, we calculated the average residuals in the end to measure prediction error. LOO-CV is more reliable and persuadable than K-CV, since it does not depend on grouping process. On the contrary, K-CV does not cost much computation time. It is preferred when we have large number of observations. The PRESS and RMSEP are simple functions of cross validation. The model with the smallest PRESS and RMSEP should be considered as the best model for prediction.

CHAPTER 3 Material and methods

3.1 Variable selection

3.1.1 Truncation PLS

In this thesis, a variable selection method was integrated with PLS in order to improve the prediction and interpretability of a PLSR model. Truncation PLS here might be considered as one of the embedded methods. As presented in PLSR step (2), for every component in PLS regression, an X loading weight vector w_a is found proportional to $X_{a-1}^T u_a$.

Every element in the loading weight vector corresponding to a specific variable could be considered as a sum of n equally distributed random variables.

$$w_{a1} = (x_{11}u_{a1} + x_{12}u_{a2} + \cdots + x_{1n}u_{an}) * s$$

where s is a scale which makes the length of w_a into 1. According to central limit theorem (CLT), the arithmetic mean and sum of a sufficiently large number of the iterates of independent random variables, each with a well-defined expected value and well-defined variance, will be approximately normally distributed. Approximately, loading weights for uninformative predictors would distribute normally with a mean of 0. On the contrary, those loading weights for important predictors would approach to a normal distribution with a non-zero mean. In truncation PLS here, all loading weights inside a confidence interval, which is believed to be independent of response, are forced to be 0. Therefore, Lenth's method is employed in this research for determining the cut-offs between the inliers and outliers. Lenth's method which was presented firstly by Lenth [Lenth, 1989 #46] is a method for deciding which effects are active in the analysis of non-replicated experiments, when the model is saturated and hence there are no degrees of freedom for estimating the error variance. In the presentation of the method, Lenth showed a reasonable estimator of the standard deviation of contrast when there were only a few significant effects. Similarly, the standard deviations of loading weights in PLS are estimated to determine the confidence interval of the loading weights of unimportant variables. The algorithm is as follows.

Consider a loading weights vector $w \in R^p$ from a PLS regression, every element in the loading weight vector w_j is corresponding to a specific variable x_j. $(j = 1, \dots, p)$

First, let

$$s_0 = 1.5 * median\{|w|\}$$

Exclude those loading weights exceeding $2.5s_0$ and get a new vector w_0

$$w_0 = \{|w| < 2.5s_0\}$$

Then standard deviation is defined as

$$sd = 1.5 * median\{|w_0|\}$$

Then upper and lower cut-off value can be calculated by $t_\alpha * sd$.

Here α denotes a truncation level, which can be set to different values between 0 and 1 ($\alpha \in (0,1]$). A smaller α leads to a larger $t_\alpha * sd$, thus more loading weights in vector w are forced to be 0. Vise versa, if $\alpha = 1$, we will get the same result as in normal PLSR, which does not include variable selection procedure. Different truncation levels were tried out in this thesis to minimize RMSEP of the truncation PLSR model. In every iteration of the truncation PLSR, we employed the truncated loading weight vector instead of the previous one to get a corresponding component.

3.1.2 Jackknife selection

In order to explore the capability of selecting relevant variables of the Lenth truncation PLS method, it is worthwhile to employ some other variable selection methods as a reference. Jackknife method was firstly introduced by (Quenouille, 1949) (Quenouille, 1956)and developed by (Tukey, 1958). As bootstrap method, it is one of the most commonly used methods for estimating variance of a complicated statistics. In addition to Lenth method, Jackknife method was also applied to select variables in

this thesis. Variances of coefficients for every variable in PLSR model are estimated by leave-one-out Jackknife method.

Consider a dataset containing p variables of n random samples from a population, $\beta_j (j = 1, \dots, p)$ is the coefficient for j th variable in PLSR model. To get the variances of the estimated coefficients for every variable in PLSR model, leave-one-out Jackknife method is to fit a PLSR model with a subset omitting the ith (i = 1, 2, …n) sample to obtain p estimated coefficients for n times. By reusing the same data as n sub-samples, $\beta_{ij} (i = 1 \dots n, j = 1 \dots p)$ are obtained. An average of these n estimated coefficients is taken as the Jackknife estimator for $\beta_j (j = 1, \dots, p)$

$$\bar{\hat{\beta}}_J = \frac{1}{n} \sum_{i=1}^{n} \beta_{ij} \quad (j = 1, \dots, p)$$

Estimates of the variances of estimated $\beta_j (j = 1, \dots, p)$ are defined as

$$var(\bar{\hat{\beta}}_J) = \frac{n-1}{n} \sum_{i=1}^{n} (\beta_{ij} - \bar{\hat{\beta}}_J)^2 \quad (j = 1, \dots, p)$$

Thus, statistic T_j $(j = 1, \dots, p)$ for the variables can be calculated as

$$T_j = \frac{\overline{\hat{\beta}_{ij}}}{SD(\hat{\beta}_{ij})} = \frac{\overline{\hat{\beta}_{ij}}}{\sqrt{\frac{n-1}{n} \sum_{i=1}^{n} (\beta_{ij} - \bar{\hat{\beta}}_J)^2}} \quad (j = 1, \dots, p)$$

Variables corresponding to the larger $|T_j|$ are believed to be more relevant. For every truncation level of $alpha$ in this thesis, the $p * alpha$ variables with largest $|T_j|$ are selected.

3.2 Data simulation

In order to explore the relationship between the performances of the truncation - based PLS and the properties of datasets to which the method is applied, some datasets with varying properties were simulated in this thesis using the relsim R package (Saebo, 2014). The structure of these datasets might be noted as $W = (Y, X)$, in which $Y = (y_1, \dots, y_n)^T \in R^n$ is considered as a continuing response of n observations. $X = (X_1, \dots, X_p)$ is a $n \times p$ predictor matrix containing p predictor variables.

Within the relsim function inside the package, some features of a dataset can be fixed in data simulating such as:

- n
  The number of observations used for training data
- $ntest$
  The number of observations used for testing data
- p
  The number of predictors
- q
  The number of relevant predictors
  q out of p predictors in X are simulated relevant to Y, others are irrelevant.
- m
  The number of relevant latent components
  m out of p latent components are simulated relevant to Y, others are irrelevant.
- R^2
  The coefficient of determination, which is defined as the proportion of total variance in Y explained by X
- γ
  A parameter indicates the degree of collinearity in X
- pos
  The position of the relevant components

To put it simple, at most 2 levels for every of these parameters are tried out as follows.

	n	p	m	q	pos	γ	R^2	ntest
Level 1	50	500	2	25	c(1,2)	0.1	0.5	100
Level 2				100	c(4,5)	0.9	0.9	

Moreover, for every combination of the parameters, 10 different random datasets are simulated. In that way, after applying truncation-based PLSR on these datasets, more robust accuracy of variable selection and *rmsep* can be obtained by calculating the mean of these 10 repetitions.

The main steps of the data simulation proceeding are demonstrated as the below.

(1) To create a matrix W=(Y, Z), in which $Y = (y_1 \dots y_n)^T \in R^n$ is considered as a continuing response of n observations. $Z = (z_1 \dots z_p)$ is a n × p matrix. $z_1 \dots z_p$ are p components of the response Y. A normal distribution is assumed for every component in this datasets.

$$\begin{bmatrix} Y \\ z_1 \\ \vdots \\ z_p \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_y \\ \mu_z \end{bmatrix}, \begin{bmatrix} \sigma_y^2 & \sigma_{zy}^t \\ \sigma_{zy} & \Lambda \end{bmatrix} \right)$$

(2) To put it simple, we make all means of variables in Y and Z equal to 0 ($\mu_y = \mu_x = 0$) and variance in Y equal to 1 (σ_y^2=*1*). Since the components are always orthogonal, the covariance matrix of them is a diagonal matrix with all the eigenvalues λ_j ($j = 1, \dots, p$) in the diagonal positions, denoted Λ here. The eigenvalues λ_j are given by a declining function $\lambda_j = e^{-r \cdot j}/e^{-r}$ in this package. In such a way, that larger γ indicates a steeper decline structure of eigenvalues, thus more multi-collinearity in X; so does the smaller γ indicates a more gradual decline structure of eigenvalues, thus

less multi-collinearity in X. To simulate m out of p components to be relevant to Y, for these components z_j $(j = 1 \dots m)$ whose corresponding elements in covariance vector $\sigma_{zy}, \sigma_{z_j y}$ should be simulated to be different from 0; while for the others, $\sigma_{z_j y} = 0$ $(j = p - m, p - m + 1, \dots, p)$. Furthermore, to make the covariance matrix $\sum zy$ to be positive definite, the values of $\sigma_{z_j y}$ $(j = 1 \dots m)$ must be restricted by a given coefficient of determination in [0,1] and satisfying:

$$R^2 = \sigma_{zy}^t \Lambda^{-1} \sigma_{zy} = \sum_{j=1}^{m} \frac{\sigma_{z_j y}^2}{\lambda_j}$$

After that, the covariance matrix $\sum zy$ might be made by combining $\sigma_y^2, \sigma_{zy}, \sigma_{zy,}^t \Lambda$ as assumed in step (1).

(3) The covariance matrix $\sum zy$ might be decomposed as

$$\sum zy = E \Lambda E^t.$$

Here $E = (e_1, e_2, \dots, e_P)$, in which e_j is the orthogonal eigenvector corresponding to λ_j. The square root matrix of $\sum zy$ can be found by

$$\sum\nolimits_{zy}^{1/2} = E \Lambda^{1/2} E^t$$

where $\Lambda^{1/2}$ is a diagonal matrix with $\sqrt{\lambda_j}$ (j=1,...,p) in its diagonal positions. U is generated as a $n \times (p + 1)$ matrix in which all elements are randomly sampled from a standard normal distribution. Matrix W=(Y,Z) can be calculated by $U \sum_{zy}^{1/2}$. Y is the first column of W and Z is the rest columns of W.

(4) In order to make an "observable" n × p matrix X, in which n refers to the number of observations and p refers to the number of predictors in X, QR decomposition is employed in the package to create a random rotation matrix. Instead of using a full ($p\ x\ p$) random rotation matrix, a block-diagonal matrix as the following one is generated in order to make q out of p variables in X simulated relevant to Y, others are irrelevant.

$$R = \begin{bmatrix} R_q & 0^t \\ 0 & R_{p-q} \end{bmatrix}$$

The two rotation matrices $R_q\ and\ R_{p-q}$ are generated separately by decomposing a standard normal data matrix of the corresponding dimension. R_q is a matrix of a dimension $q \times q$ and R_{p-q} is a matrix of a dimension $(p-q) \times (p-q)$. 0 is a null matrix of a dimension $(p-q) \times q$.

(5) At last, the "observable" X is generated as a rotated Z:

$$X = ZR$$

CHAPTER 4 Results

4.1 Factorial two level design in data simulation

In the relsim R package, a total of 7 parameters are chosen to determine the properties of a dataset. 3 of them were investigated in this research: pos, γ, and R^2. In the full factorial two level design, all combination of the levels of the 3 factors were analyzed, hence $2^3 = 8$ types of dataset. Under every parameter setting, 10 replicated datasets were simulated.

In contrast, the number of observations used for testing $ntest$, which will affect the precision of evaluating performance of the models is kept constant equal to 100 all along in this research to make the results comparable.

Some other parameters in the relsim R package were chosen as:
the number of observations used for training data $n = 50$;
the number of predictors $p = 500$;
the number of relevant predictors $q = 25$.

The following table displays the design of parameter setting in data simulation.

	pos	γ	R^2
Design 1	-	-	-
Design 2	-	-	+
Design 3	-	+	-
Design 4	-	+	+
Design 5	+	-	-
Design 6	+	-	+
Design 7	+	+	-
Design 8	+	+	+

Table 1. The design of parameter setting in data simulation

The levels of the factors in Table 1 were set as follows:

	-	+
pos	(1,2)	(4,5)
γ	0.1	0.9
R^2	0.5	0.9

Table 2. The levels of factors in the factorial design

The components were ordered in data simulation, from the component containing the largest eigenvalue to the one containing the smallest eigenvalue. Consequently, for the designs with a "+" in pos, the dataset has relevant components in the position 4 and 5. In other words, the components containing the 4th and 5th largest eigenvalue are relevant to the response. Similarly, for these design with a "-" in pos, the dataset has its 1st and 2nd components relevant to the response.

The models were evaluated from two perspectives: the capability of predicting new observations and the capability of selecting the true relevant variables. According to these purposes, truncation-PLS was applied to every simulated dataset with various designs for the parameters. RMSEP and accuracy of variable selection for a certain component, truncation level, and type of dataset were obtained by calculating the mean value of the 10 results from the 10 replications. Meanwhile, RMSEP and accuracy of variable selection of another method PLS with Jackknife selection were calculated as a reference.

4.2 ANOVA of RMSEP

In order to apply the Truncation PLS methods, a certain number of components ($comp$) and truncation level (α) should be set. In this research, all the component numbers from k=1 to k=8 were tried out. The truncation levels α were chosen to be 0.01, 0.05, 0.1, and 0.25.

In the process of applying Jackknife PLS method, a normal PLS regression model with a certain number of components was fitted before any variable selection procedure. Based on the estimated coefficients of the model, Jackknife method was used to select the most relevant variables. The number of relevant predictors to be selected was chosen to be $p * \alpha$, where p is the number of predictors; α is the same value as the corresponding truncation level in Truncation PLS methods. And then, only the selected variables were used to refit the model. The best number of components, which lead to the smallest PRESS (or RMSEP), was chosen to predict afterwards. The number of components in the results for Jackknife PLS regression indicates the number of components that we used to fit the regression model before variable selection. The optional number of components used in the refit varied from dataset to dataset.

The 5 parameters R^2, pos, γ, α, and comp were set as factors. Then a linear model was fitted in R with a set of these 5 main factors, the terms obtained by taking all the second order interactions of them, and a response of RMSEP. RMSEP values were obtained earlier by applying Truncation-PLS method to the various datasets. The second order ANOVA model might be written in R syntax as

$$\text{RMSEP} \sim (R^2 + \text{pos} + \gamma + \alpha + \text{comp})\text{^}2 \qquad (4.1)$$

To study the effects of these factors, ANOVA was used to analyze the linear model above. The output is as follows.

s: 0.08196 on 188 degrees of freedom

Multiple R-squared: 0.9443,

Adjusted R-squared: 0.9245

F-statistic: 47.59 on 67 and 188 DF, p-value: $<$ 2.2e-16

As can be seen from the output, Multiple R-squared is larger than 0.94. Therefore, most variance in RMSEP can be explained by some of these factors and their interactions. Furthermore, the result of ANOVA F-test shows an extremely small p-value, which is smaller than 2.2e-16. In general, these factors are significant.

	DF	Sum Sq	Mean Sq	F value	Pr(>F)
Intercept	1	2.45814	2.45814	365.9482	< 2.2e-16 ***
α	3	0.03716	0.01239	1.8441	0.1406366
comp	7	2.61224	0.37318	55.5556	< 2.2e-16 ***
γ	1	0.08347	0.08347	12.4257	0.0005319 ***
pos	1	0.01314	0.01314	1.9563	0.1635534
R^2	1	1.13471	1.13471	168.9258	< 2.2e-16 ***
α:comp	21	0.45404	0.15135	22.5315	1.656e-12 ***
α: γ	3	1.7706	0.59020	15.7040	3.728e-09 ***
α:pos	3	0.00379	0.00126	0.1880	0.9044428
α: R^2	3	0.16054	0.05351	7.9664	4.995e-05 ***
comp: γ	7	1.88343	0.26906	40.0556	< 2.2e-16 ***
comp:pos	7	0.14219	0.02031	3.0241	0.0049132 **
comp: R^2	7	0.66004	0.09429	14.0373	1.341e-14 ***
γ:pos	1	0.06128	0.06128	9.1230	0.0028752 **
γ: R^2	1	0.75209	0.75209	111.9657	< 2.2e-16 ***
pos: R^2	1	0.02140	0.02140	3.1865	0.0758594
Residuals	188	1.26283	0.00672		
Significant. Codes: '***' 0.001 '**' 0.01 '*' 0.05 '					

Table 3. The analysis of variance for the linear model (4.1)

According to the figures shown in the Table 3, we can see that $comp$, γ, and R^2 as main factors are highly significant with p values smaller than 0.001. The other main factors, α and pos are not significant as a main factor, but both of them have a strong interaction with some other parameters, for instance, the interaction between pos and $comp$, pos and γ, $comp$ and γ, α and γ, & α and R^2.

4.3 RMSEP in Truncation-PLS regression

The following figures show interaction plots for various choices of the ANOVA model factors. The values in the plots are mean values of RMSEP under the chosen factor values.

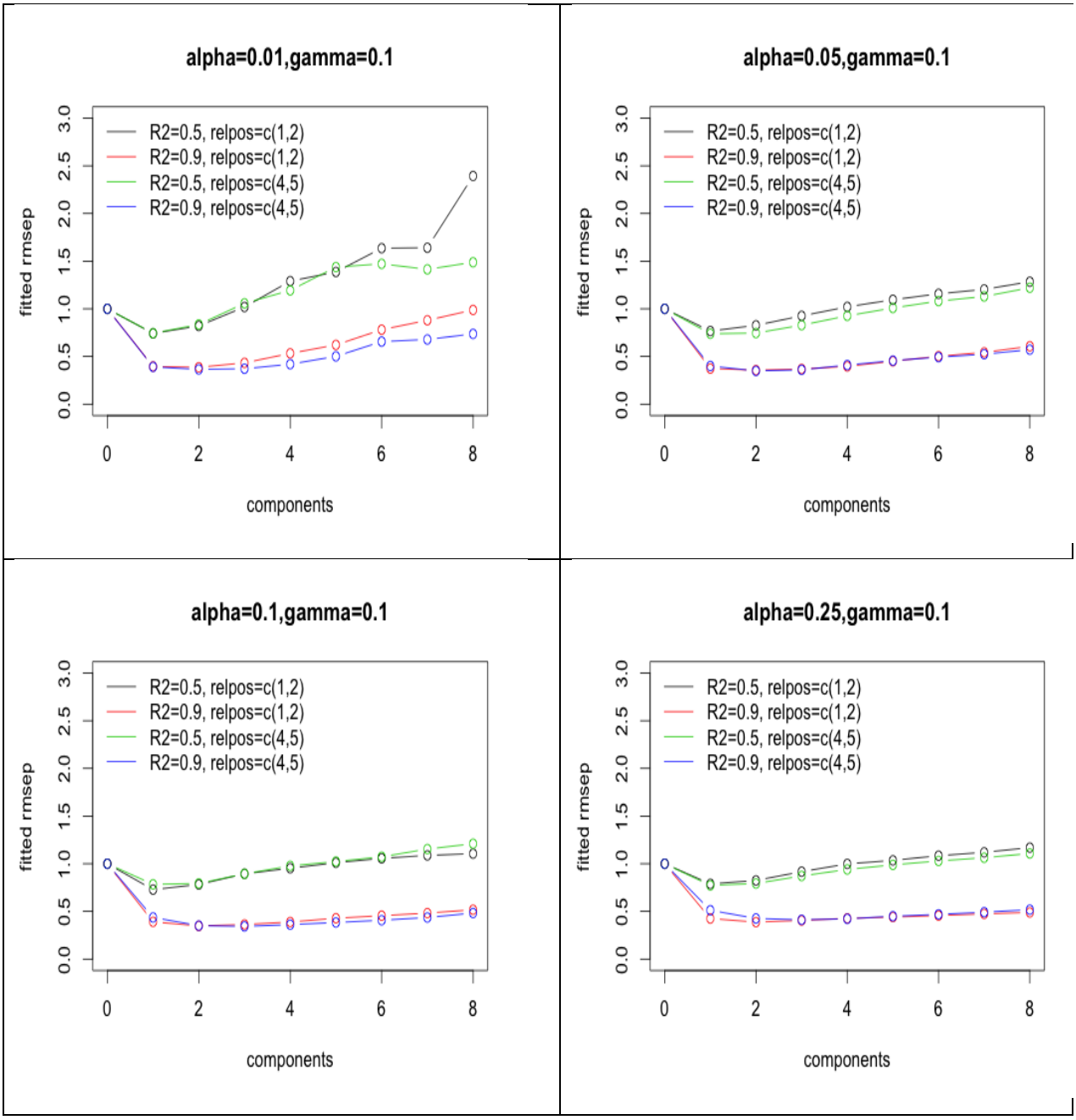

Figure 1. RMSEP in Truncation-PLS regression when γ equals to 0.1. The plots present the mean RMSEP in Truncation-PLS regression with four different truncation-levels. The plot in upper left panel corresponds to the truncation level (α) of 0.01. The others correspond to the Truncation-PLS regression with a truncation level of 0.05, 0.1, and 0.25 respectively. The

scales in vertical axis indicate the value of RMSEP in Truncation-PLS regression model. The scales in the horizontal axis indicate the number of components being used in the regression. At the position with 0 component, RMSEP is always set to 1, which is the assumed unconditional variance of the response in data simulation. The coefficient of determination (R^2) and the position of relevant components (pos) are distinguished by colors.

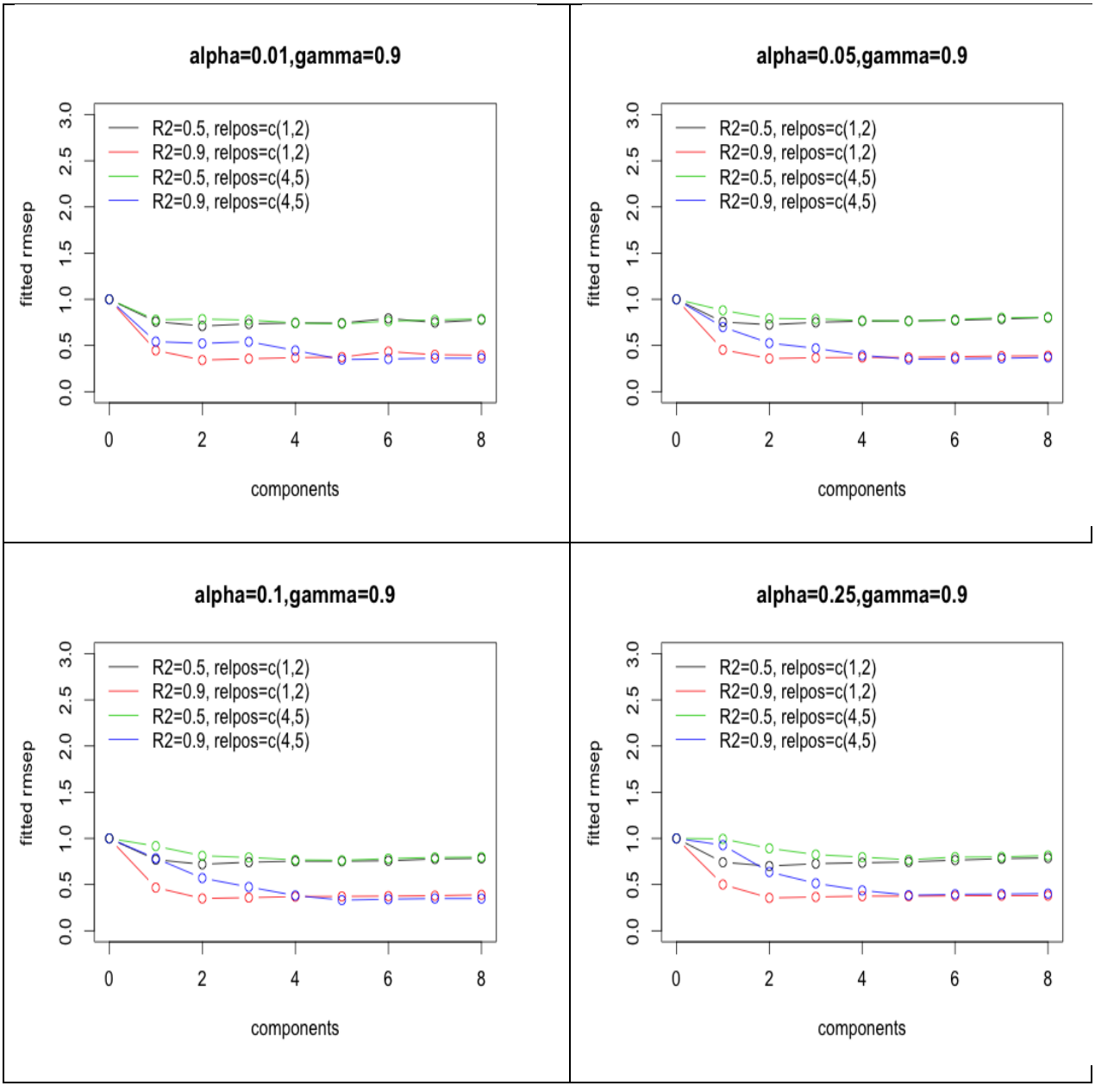

Figure 2. RMSEP in Truncation-PLS regression when γ equals to 0.9. The plots here have the same structure as those in Figure 1.

A list of main features can be read from the plots.

1. The effect of R^2

   The Figure shows that the RMSEP values on the datasets of R^2=0.9 (red and blue lines) are significantly smaller than those on the datasets of R^2=0.5(black and green lines).

2. The effect of γ

   After considering the information in the two above figures, we might conclude that the RMSEP values in Figure 1 where $\gamma = 0.1$ are generally larger than those ones in Figure 2 where $\gamma = 0.9$.

3. Interaction between comp and γ

   In Figure 1 where $\gamma = 0.1$, the plots reveal that the RMSEP values are smaller with less components but increase while using more components. In contrast, in Figure 2 where r=0.9, RMSEP values do not increase much as more components are used.

4. Interaction between pos and γ

   In Figure 1 where $\gamma = 0.1$, the position of relevant components is not very important, from the fact that green lines and black lines are close to each other; the red lines are close to the blue lines. In contrast, in Figure 2 where r=0.9, the black and red lines reach their best prediction earlier than the green and blue lines. In other words, by comparison with a situation of $pos = (1,2)$, more components are needed to get the minimum RMSEP when $pos = (4,5)$.

5. The effect of α

   The following figures illustrate the effect of α in in Truncation-PLS regression.

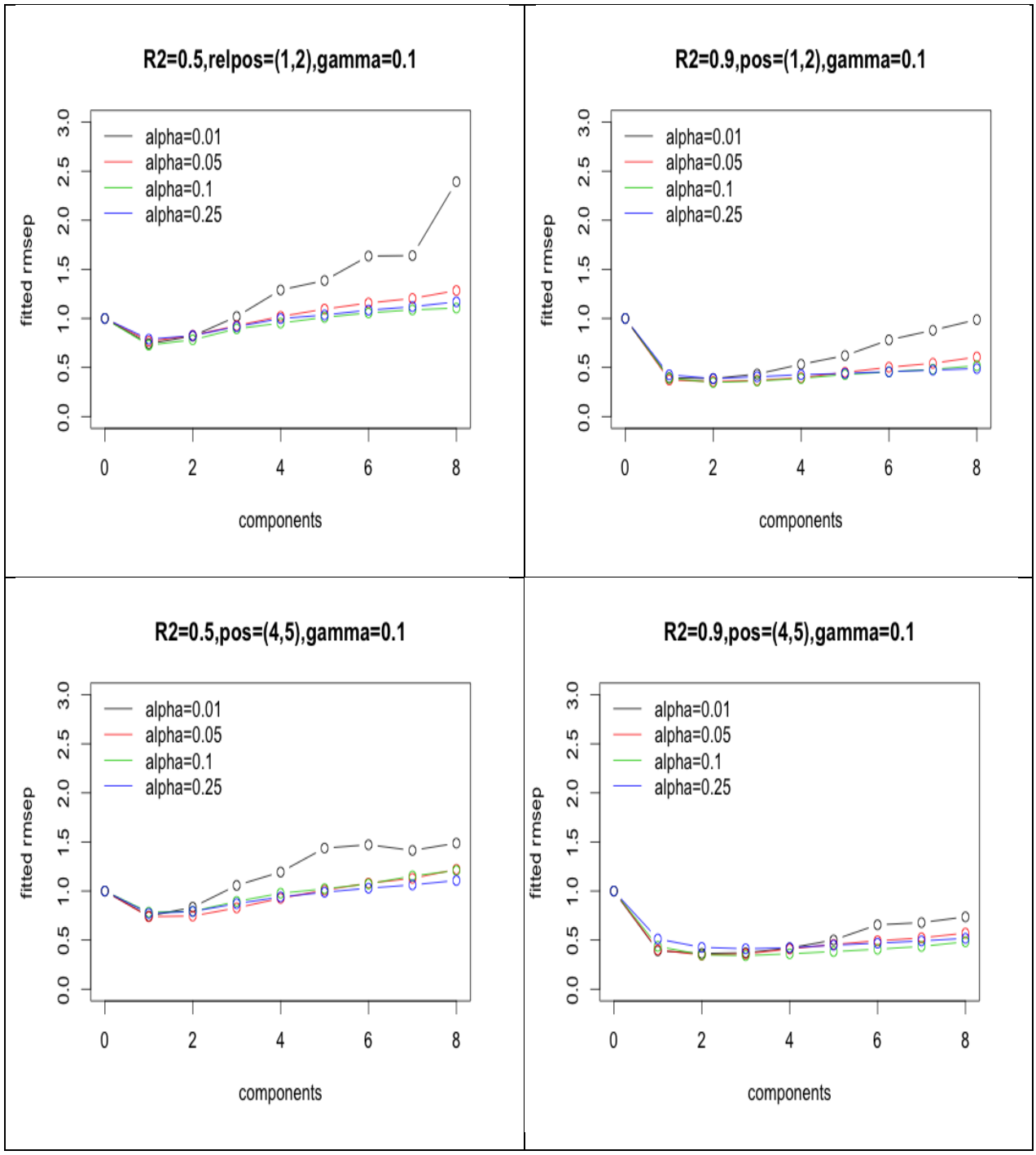

Figure 3. Effect of truncation level in Truncation-PLS regression when γ equals to 0.1.

The plots present the mean RMSEP in Truncation-PLS regression obtained from datasets with different parameter setting. The scales in vertical axis indicate the value of RMSEP in Truncation-PLS regression model. The scales in the horizontal axis indicate the number of components being used in the regression. The four truncation levels are distinguished by colors.

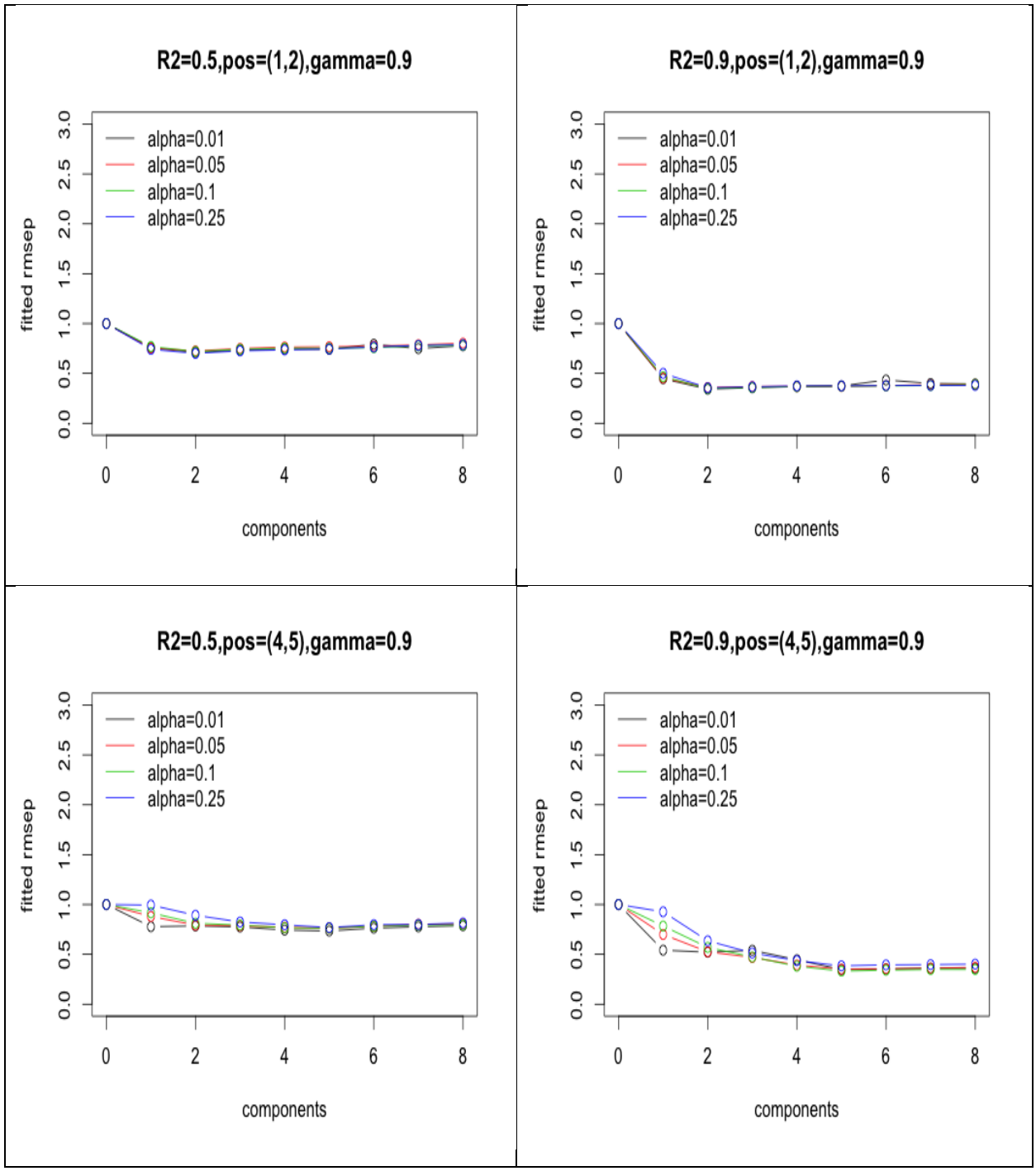

Figure 4. Effect of truncation level in Truncation-PLS regression when γ equals to 0.9. The plots here have the same structure as those in Figure 3.

In contract to the plots in Figure 4 where $\gamma = 0.9$, the plots in Figure 3 where $\gamma = 0.1$ show a marked increase in RMSEP as more and more components are included in the truncation-PLS regression.

In general, the truncation level (α) does not show any significant level as a main factor. But from the two plots at the bottom of the figure where the $pos = (4,5)$ and $\gamma = 0.9$, which is the most difficult situation for predicting, the truncation level (α) shows an effect on RMSEP. With α = 0.01, the method reaches a satisfying RMSEP by using only one component.

4.4 Comparison with Jackknife method

In the following figure, the RMSEP values of Jackknife-PLS method are plotted against those ones of the Truncation-PLS method as a contrast.

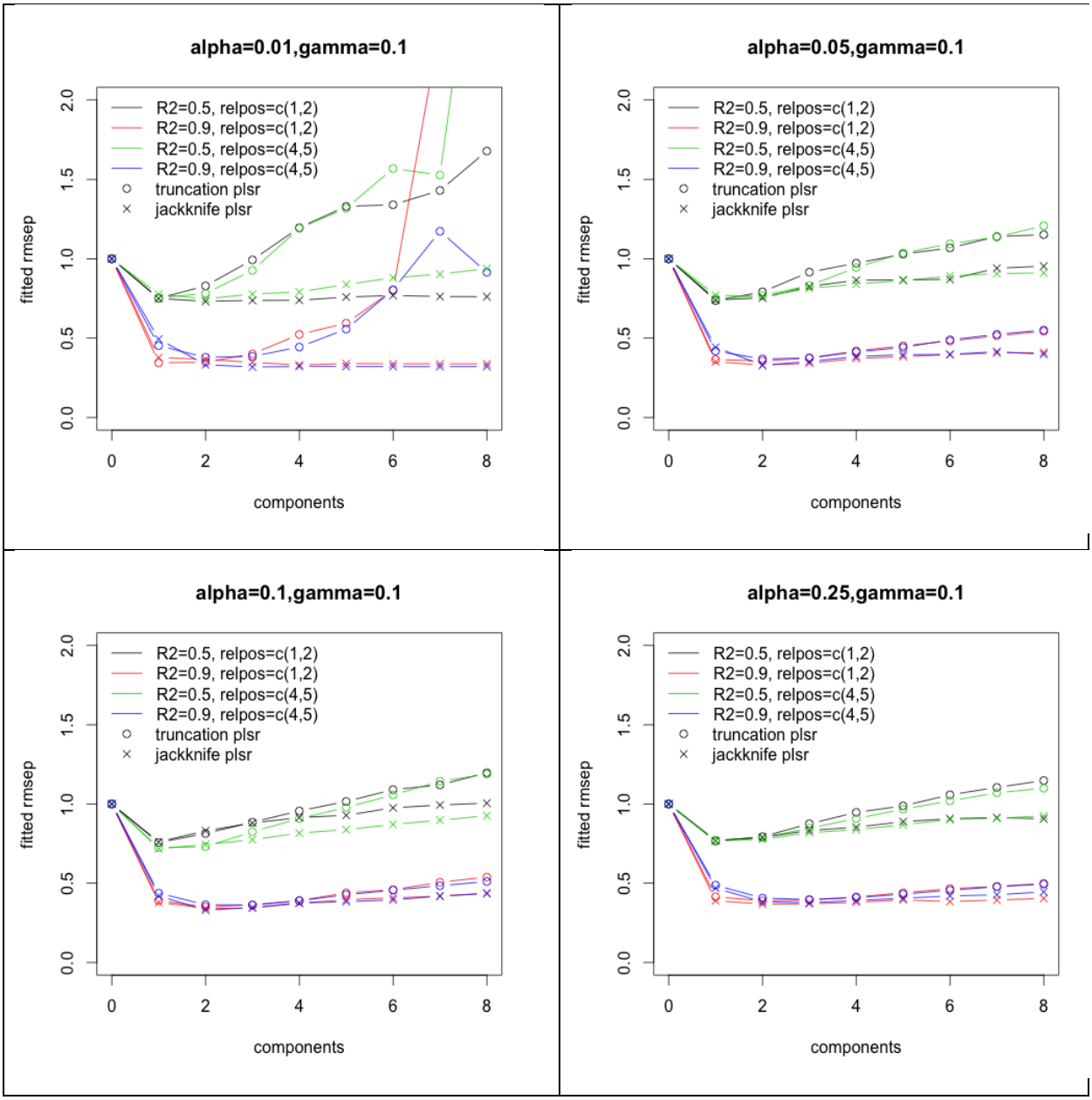

Figure 5. RMSEP in Truncation-PLS and Jack-knife PLS regression when γ equals to 0.1.

The four plots present the mean RMSEP in Truncation-PLS regression and Jackknife PLS regression with four different truncation-levels. The plot in upper left panel corresponds to the truncation level (α) of 0.01. The others correspond to the Truncation-PLS regression with a truncation level of 0.05, 0.1, and 0.25 respectively. Results from Truncation-PLS regression are labeled with ∘, while the results from Jack-knife PLS regression are labeled with ×. The scales in the vertical axis indicate the value of RMSEP. The scales in the horizontal axis indicate the number of components being used in the regression. The coefficient of determination (R^2) and the position of relevant components (pos) are distinguished by colors. At the position with 0 component, RMSEP is always equal to 1, which is the assumed variance of response in data simulation.

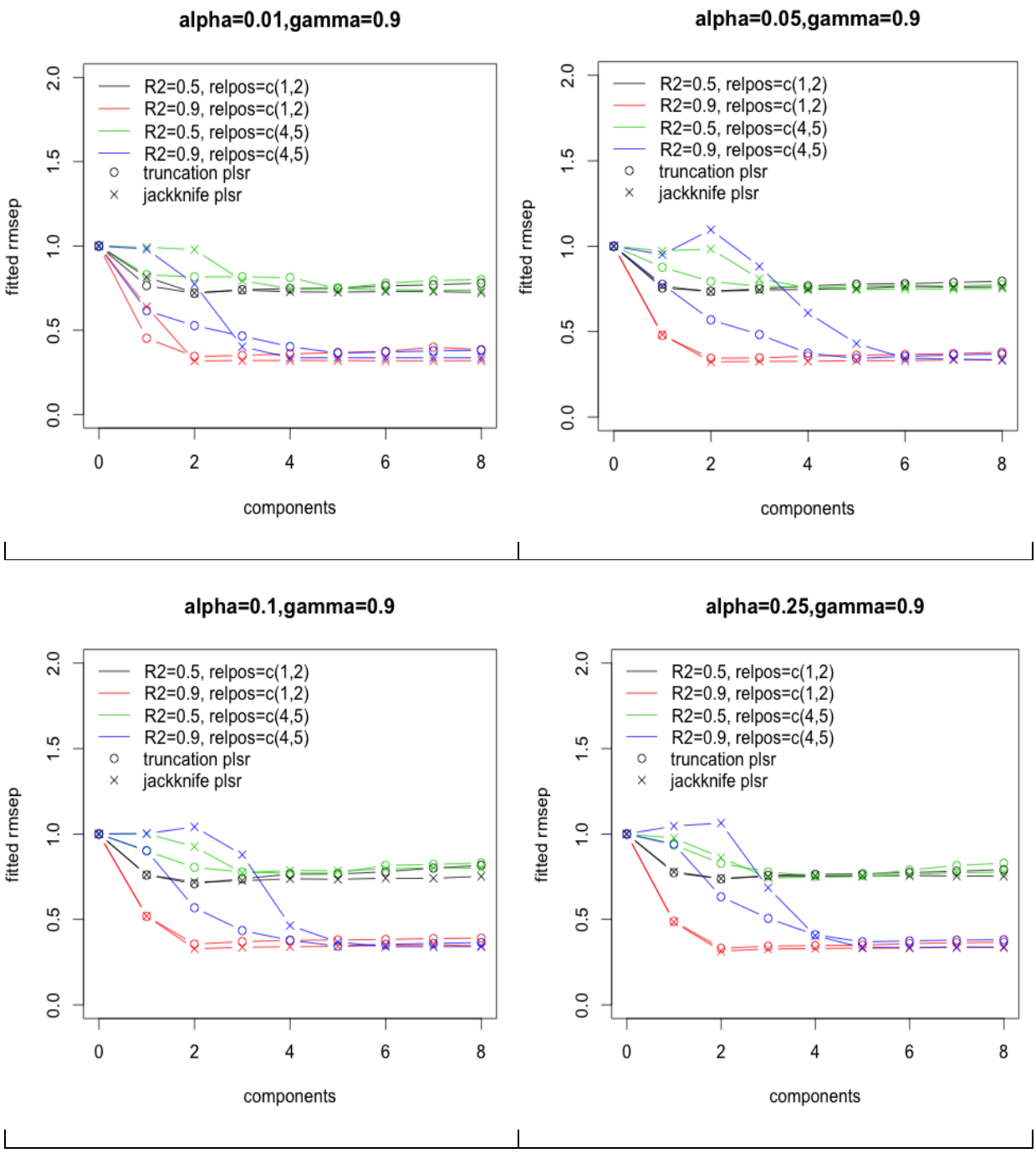

Figure 6. RMSEP in Truncation-PLS and Jackknife PLS regression when γ equals to 0.9. The plots here have the same structure as those in Figure 5.

As we can see in Figure 5 where γ =0.1, in contrast to the results from Truncation-PLS regression, the RMSEP values in Jackknife PLS regression do not increase much

with the number of components. In figure 6 where $\gamma = 0.9$, no increase in RMSEP is shown for both of the two methods as more components are used.

Moreover, we have noted from figure 6 that in Jackknife method there are more problems than in Truncation-PLS method when $\gamma = 0.9$ and $pos = (4,5)$. With a dataset of such a feature, both of the methods need more components to achieve the minimum RMSEP. As it is shown thoroughly in Table 4 and Table 5, both of the methods often achieve more or less the same minimum RMSEP, but Jackknife method needs even more components.

The best predictions of the two methods are more or less the same; the RMSEP values of Jackknife-PLS method is slightly lower than those ones of Truncation-PLS method in most design of datasets. (See section 4.6, Table 4 and Table 5 for the exact values)

4.5 Accuracies of variable selection in Truncation-PLS and Jackknife PLS regression

The accuracy of variable selection is calculated by the percentage of the variables that are classified correctly as relevant and irrelevant.

$$Accuracy = \frac{nr + ni}{p}$$

where nr indicates the number of selected true relevant predictors; ni indicates the number of non-selected true irrelevant predictors; p is the number of all predictors.

In the following figure, the accuracies of variable selection of Jackknife-PLS method are plotted against those ones of Truncation-PLS method as a contrast. The 8 plots present the accuracy results from datasets of 8 different parameter setting.

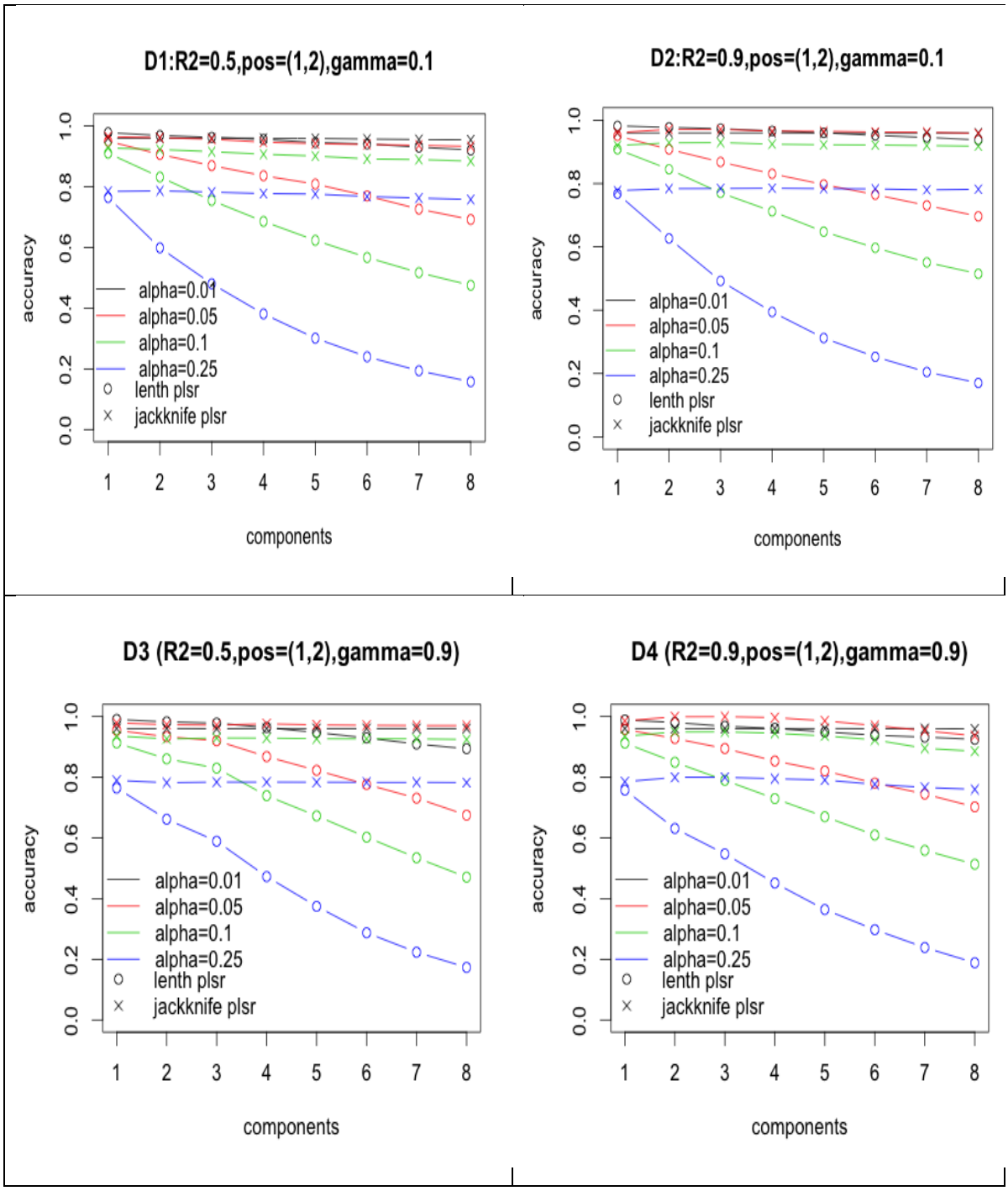

Figure 7. Accuracies of variable selection in Truncation-PLS and Jackknife PLS regression.

Results from Truncation-PLS are labeled with ∘; results from Jackknife PLS method are labeled with ×. The scales in the vertical axis indicate the value of accuracy in Truncation-PLS regression model. The scales in the horizontal axis indicate the number of components being used in the regression. The four different truncation levels are distinguished by colors.

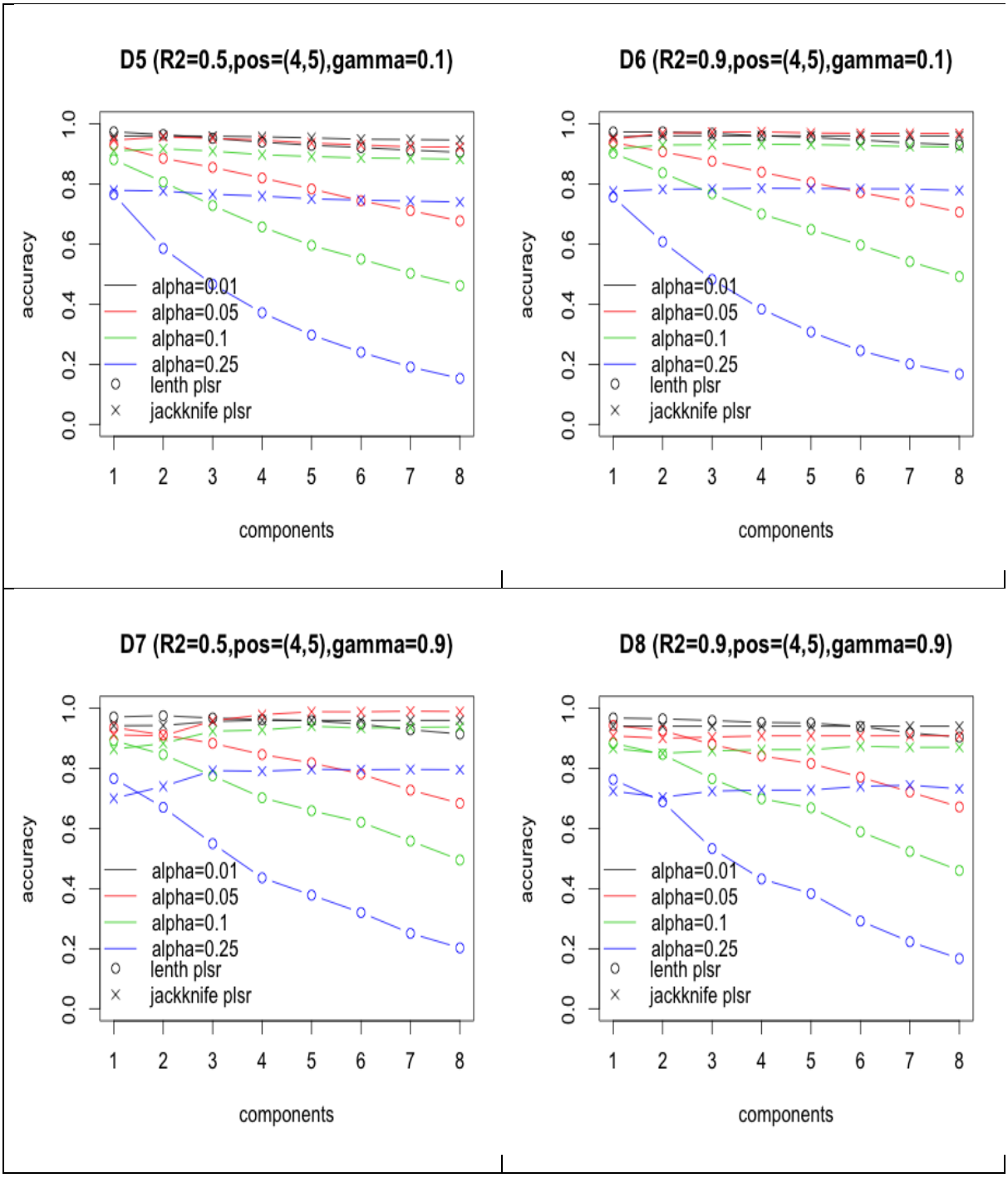

Figure 8. Accuracies of variable selection in Truncation-PLS and Jackknife PLS regression. The plots here have the same structure as those in Figure 7.

After considering the information in Figure 7 and Figure 8, the accuracies of variable selection calculated by Truncation PLS method with different α steadily decline with the number of components raised, while the curve of Jackknife PLS method has leveled off.

Considering all the different settings of truncation level, the accuracies of variable selection calculated by Truncation PLS method with a truncation level $\alpha = 0.01$ are always higher than those with other truncation levels. The accuracies of variable selection calculated by Jackknife PLS method with a test level $\alpha = 0.01$ are close to those with $\alpha = 0.05$.

Moreover, we have noted from the bottom plots of figure 8 that Jackknife method needs more components to find the correct variables than Truncation-PLS method when γ =0.9 and $pos = (4,5)$.

The best accuracies of variable selection of the two methods are more or less the same; the accuracy values of Truncation-PLS method is at its maximum slightly higher than those ones of Jackknife-PLS method in most design of datasets. (See section 4.6, table 6 and table 7 for the exact values)

4.6 The best choice of α and $comp$

For every design of datasets, the best choice of a truncation level and number of the components of truncation-PLS method were chosen by identifying the minimum RMSEP, and its corresponding truncation level and the number of components.

The following table shows the result of best combination of truncation level and number of component of truncation-PLS for different types of datasets.

	Truncation level	Number of components	Minimum RMSEP	Minimum achievable RMSEP
D1: R^2=0.5,pos=(1,2),γ =0.1	0.05	1	0.74	0.71
D2: R^2=0.9,pos=(1,2),γ =0.1	0.01	1	0.34	0.32
D3: R^2=0.5,pos=(1,2),γ =0.9	0.1	2	0.71	0.71
D4: R^2=0.9,pos=(1,2),γ =0.9	0.25	2	0.33	0.32
D5: R^2=0.5,pos=(4,5),γ =0.1	0.1	1	0.72	0.71
D6: R^2=0.9,pos=(4,5),γ =0.1	0.1	3	0.36	0.32
D7: R^2=0.5,pos=(4,5),γ =0.9	0.01	5	0.75	0.71
D8: R^2=0.9,pos=(4,5),γ =0.9	0.05	5	0.34	0.32

Table 4. The truncation level and number of components corresponding to the smallest RMSEP for Truncation-PLS method. The minimum achievable RMSEP for a certain dataset is given by $\sigma^2 = \sqrt{1 - R^2}$, where R^2 is the coefficient of determination.

After considering the information in Table 4, it might be concluded that when $pos = (4,5)$ more components are needed for Truncation-PLS method to achieve the minimum RMSEP. Especially in the designed datasets D7 and D8, where γ =0.9 and $pos = (4,5)$, 5 components are used to obtain the best prediction.

Similarly, the best combinations of test level and number of components for the Jackknife-PLS method were chosen by identifying the minimum RMSEP, and its corresponding truncation level and number of components. In the Jackknife method, the test level was used to decide the number of variables to be selected. The result is shown as follows.

	Test level	Number of components	Minimum RMSEP	Minimum achievable RMSEP
D1: R^2=0.5,pos=(1,2),γ =0.1	0.01	2	0.73	0.71
D2: R^2=0.9,pos=(1,2),γ =0.1	0.05	2	0.33	0.32
D3: R^2=0.5,pos=(1,2),γ =0.9	0.1	2	0.72	0.71
D4: R^2=0.9,pos=(1,2),γ =0.9	0.25	2	0.32	0.32
D5: R^2=0.5,pos=(4,5),γ =0.1	0.1	1	0.72	0.71
D6: R^2=0.9,pos=(4,5),γ =0.1	0.01	3	0.32	0.32
D7: R^2=0.5,pos=(4,5),γ =0.9	0.01	7	0.74	0.71
D8: R^2=0.9,pos=(4,5),γ =0.9	0.05	8	0.33	0.32

Table 5. The test level and number of components corresponding to the smallest RMSEP for Jackknife-PLS method. The minimum achievable RMSEP for a certain dataset is given by $\sigma^2 = \sqrt{1 - R^2}$

By comparison with Table 4, the minimum RMSEP values of Jackknife-PLS method in table 5 are slightly lower than those of truncation-PLS method in most designs of datasets except for D3 and D5. However, Jackknife-PLS needs more components than truncation-PLS to achieve the best prediction in all the designs of datasets. Especially in the designed datasets D7 and D8, where γ =0.9 and $pos = (4,5)$, 7 and 8 components are used to obtain the best prediction respectively.

	Truncation level	Number of components	Maximum accuracy
D1: R^2=0.5,pos=(1,2),γ =0.1	0.01	1	0.978
D2: R^2=0.9,pos=(1,2),γ =0.1	0.01	1	0.982
D3: R^2=0.5,pos=(1,2),γ =0.9	0.01	1	0.99
D4: R^2=0.9,pos=(1,2),γ =0.9	0.01	1	0.988
D5: R^2=0.5,pos=(4,5),γ =0.1	0.01	1	0.973
D6: R^2=0.9,pos=(4,5),γ =0.1	0.01	1,2	0.973
D7: R^2=0.5,pos=(4,5),γ =0.9	0.01	2	0.971
D8: R^2=0.9,pos=(4,5),γ =0.9	0.01	1	0.968

Table 6. The truncation level and number of components corresponding to the highest accuracy of variable selection for Truncation-PLS method

As it is shown in table 6, in most designs of datasets, truncation-PLS method achieves the maximum accuracy of variable selection with a truncation level equals to 0.01, and only one component, or at most two.

	Test level	Number of components	Maximum accuracy
D1: R^2=0.5,pos=(1,2),γ =0.1	0.05	1	0.965
D2: R^2=0.9,pos=(1,2),γ =0.1	0.05	2,3	0.971
D3: R^2=0.5,pos=(1,2),γ =0.9	0.05	1	0.978
D4: R^2=0.9,pos=(1,2),γ =0.9	0.05	3	0.999
D5: R^2=0.5,pos=(4,5),γ =0.1	0.01	2	0.96
D6: R^2=0.9,pos=(4,5),γ =0.1	0.05	4	0.973
D7: R^2=0.5,pos=(4,5),γ =0.9	0.05	7	0.990
D8: R^2=0.9,pos=(4,5),γ =0.9	0.01	1:8	0.94

Table 7. The test level and number of components corresponding to the highest accuracy of variable selection for Jackknife-PLS method

By comparison with Table 6, the maximum accuracy of Jackknife-PLS method in Table 7 is lower than the maximum accuracy of truncation-PLS method in most designs of datasets except for D4 and D7. Moreover, Jackknife-PLS needs more components than truncation-PLS to achieve the maximum accuracy of variable selection.

4.7 The inconsistency in the best choice of truncation level and number of components

After applying Truncation PLS with all the component numbers from 1 to 8 to different designs of datasets, the best number of components, which is used to achieve a minimum RMSEP, is not always identical to the one leading to the maximum accuracy of variable selection. The following plot shows the inconsistency.

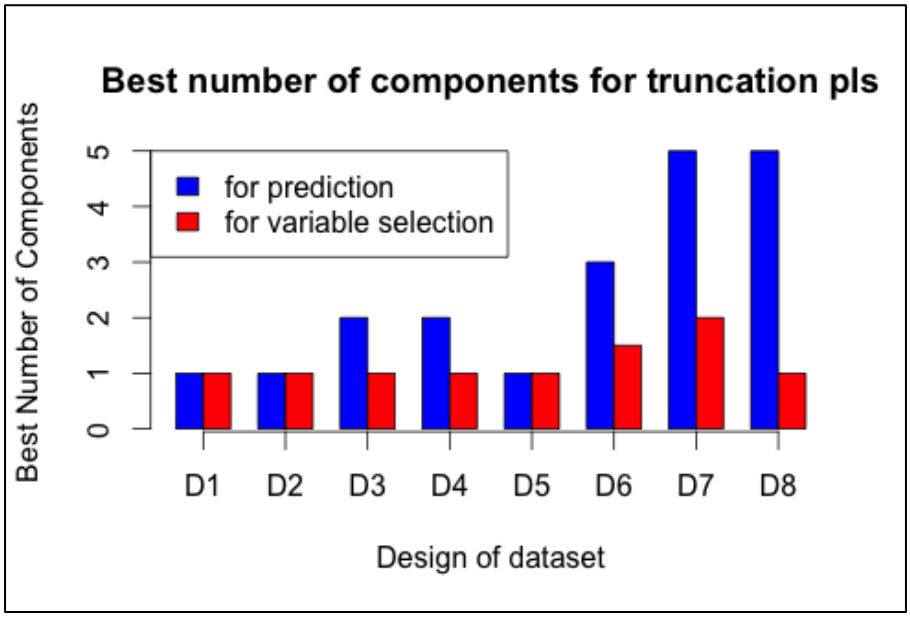

Figure 9. The best number of components for truncation PLS method.

This bar plot compares the best number of components for prediction with the best number of components for variable selection in all 8 designs of datasets. The blue bars indicate the number of components, which is used to achieve a minimum RMSEP; the red ones indicate the number of components, which is used to achieve the maximum accuracy of variable selection.

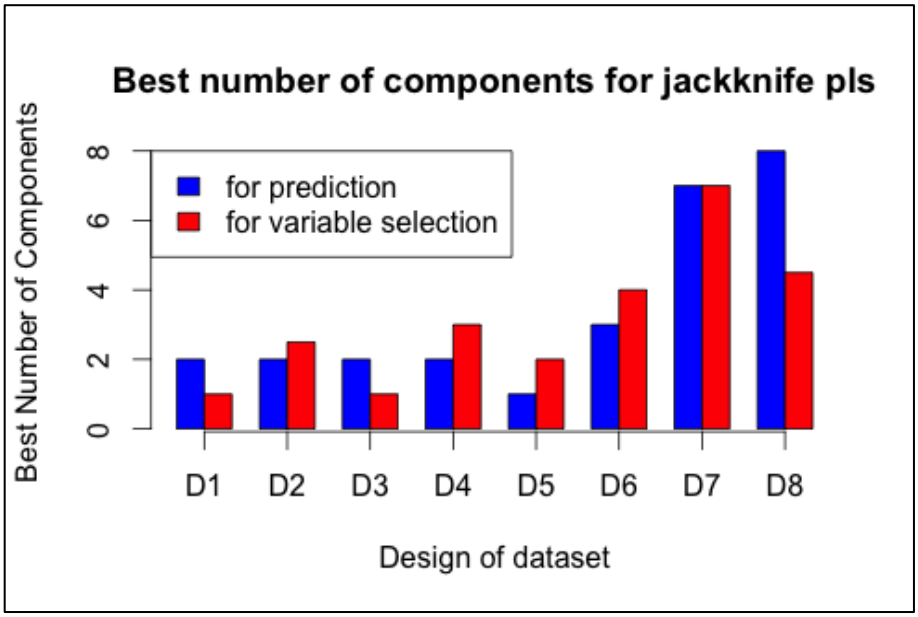

Figure 10. The best number of components for Jackknife PLS method. This plot has the same structure as Figure 9.

Figure 9 shows that in most designs of dataset, such as D3, D4, D6, D7 and D8, the best number of components for prediction is larger than the best number of components for variable selection. In the other datasets such as D1, D2, D5, the number of components for prediction is the same as the one for variable selection.

In contrast, for Jackknife PLS method, the best number of components for variable selection and for prediction is comparable with one another in every design of datasets as it is shown in Figure 10.

The following tables demonstrate some extreme examples in this research showing the inconsistency. Table 8 shows the number of variables selected by truncation PLS with a truncation level $\alpha = 0.01$ and 1 component which produced the maximum accuracy of variable selection in datasets of design 8, while Table 9 shows the number of variables selected by truncation PLS with a truncation level $\alpha = 0.05$ and 5 components which produced the minimum RMSEP in datasets of design 8. On the contrary, in datasets of design 2, truncation PLS achieved both the minimum RMSEP and the maximum accuracy of variable selection with a truncation level $\alpha = 0.01$ and 1 component. The number of selected variables is shown in Table 10.

	True relevant	True irrelevant	Sum
Estimated relevant	5	2	7
Estimated irrelevant	20	473	493
Sum	25	475	500

Table 8. The number of variables selected by truncation PLS which produced the maximum accuracy in datasets of design 8.

$$\text{Sensitivity} = \frac{5}{25} = 0.2$$

$$\text{Specificity} = \frac{473}{475} = 0.996$$

$$\text{Accuracy} = \frac{5 + 473}{500} = 0.956$$

Since the most variables are irrelevant in the simulated data, the accuracy can be high (0.956) even if the sensitivity is low (0.2).

	True relevant	True irrelevant	Sum
Estimated relevant	21	77	98
Estimated irrelevant	4	398	402
Sum	25	475	500

Table 9. The number of variables selected by truncation PLS which produced the minimum RMSEP in datasets of design 8.

$$\text{Sensitivity} = \frac{21}{25} = 0.84$$

$$\text{Specificity} = \frac{398}{475} = 0.83$$

$$\text{Accuracy} = \frac{21 + 398}{500} = 0.838$$

The specificity and accuracy from Table 9 are lower than those from Table 8. But the sensitivity is relatively higher which means 21 of the 25 true relevant variables are selected in the model.

In some datasets of a parameter setting like design 2, the best choice of truncation level and number of components are consistent for the truncation PLS to achieve its minimum RMSEP as well as its maximum accuracy of variable selection.

	True relevant	True irrelevant	Sum
Estimated relevant	22	1	23
Estimated irrelevant	3	474	477
Sum	25	475	500

Table 10. The number of variables selected by truncation PLS which produced the minimum RMSEP (or the maximum accuracy) in datasets of design 2.

$$\text{Sensitivity} = \frac{22}{25} = 0.88$$

$$\text{Specificity} = \frac{474}{475} = 0.998$$

$$\text{Accuracy} = \frac{22 + 474}{500} = 0.992$$

According to the figures shown in Table 10, the variables selected by the model have high sensitivity, specificity, and accuracy.

4.8 The effect of q

In order to explore the effect of q, we repeated the experiment with every parameter being set at the same value as before except the true number of relevant predictors q is set at 100 instead of 25 in all the simulated datasets. The following tables (Table 11, 12, 13, and14) show the results of the new experiment with $q = 100$. And they are in the same structure as Table 4,5,6,7 in section 4.6.

	Truncation level	Number of components	Minimum RMSEP	Minimum achievable RMSEP
D1:R^2=0.5,pos=(1,2),γ =0.1	0.25	1	0.72	0.71
D2: R^2=0.9,pos=(1,2),γ =0.1	0.25	2	0.36	0.32
D3: R^2=0.5,pos=(1,2),γ =0.9	0.25	2	0.72	0.71
D4: R^2=0.9,pos=(1,2),γ =0.9	0.05	2	0.32	0.32
D5: R^2=0.5,pos=(4,5),γ =0.1	0.05	1	0.74	0.71
D6: R^2=0.9,pos=(4,5),γ =0.1	0.1	3	0.36	0.32
D7: R^2=0.5,pos=(4,5),γ =0.9	0.25	5	0.74	0.71
D8: R^2=0.9,pos=(4,5),γ =0.9	0.05	5	0.35	0.32

Table 11. The truncation level and number of components corresponding to the minimum RMSEP for Truncation-PLS method when q=100.

	Test level	Number of components	Minimum RMSEP	Minimum achievable RMSEP
D1: R^2=0.5,pos=(1,2),γ =0.1	0.25	1	0.72	0.71
D2: R^2=0.9,pos=(1,2),γ =0.1	0.1	2	0.32	0.32
D3: R^2=0.5,pos=(1,2),γ =0.9	0.25	2	0.72	0.71
D4: R^2=0.9,pos=(1,2),γ =0.1	0.05	5	0.32	0.32
D5: R^2=0.5,pos=(4,5),γ =0.1	0.05	2	0.74	0.71
D6: R^2=0.9,pos=(4,5),γ =0.1	0.05	6	0.34	0.32
D7: R^2=0.5,pos=(4,5),γ =0.9	0.01	8	0.74	0.71
D8: R^2=0.9,pos=(4,5),γ =0.9	0.01	5	0.32	0.32

Table 12. The test level and number of components corresponding to the minimum RMSEP for Jackknife-PLS method when p=100.

	Truncation level	Number of components	Maximum accuracy
D1: R^2=0.5,pos=(1,2), γ =0.1	0.05	1	0.88
D2: R^2=0.9,pos=(1,2), γ =0.1	0.01	1	0.9
D3: R^2=0.5,pos=(1,2), γ =0.9	0.01	2	0.98
D4: R^2=0.9,pos=(1,2), γ =0.9	0.01	2	0.988
D5: R^2=0.5,pos=(4,5), γ =0.1	0.05	1	0.87
D6: R^2=0.9,pos=(4,5), γ =0.1	0.01	1	0.9
D7: R^2=0.5,pos=(4,5), γ =0.9	0.01	4	0.99
D8: R^2=0.9,pos=(4,5), γ =0.9	0.01	1	0.97

Table 13. The truncation level and number of components corresponding to the maximum accuracy for Truncation PLS method when q=100.

	Test level	Number of components	Maximum accuracy
D1: R^2=0.5,pos=(1,2), γ =0.1	0.1	1	0.87
D2: R^2=0.9,pos=(1,2), γ =0.1	0.1	2	0.90
D3: R^2=0.5,pos=(1,2), γ =0.9	0.1	1:4	0.90
D4: R^2=0.9,pos=(1,2), γ =0.9	0.25	2	0.92
D5: R^2=0.5,pos=(4,5), γ =0.1	0.1	2	0.87
D6: R^2=0.9,pos=(4,5), γ =0.1	0.1	3:4	0.90
D7: R^2=0.5,pos=(4,5), γ =0.9	0.1	5	0.89
D8: R^2=0.9,pos=(4,5), γ =0.9	0.05	8	0.99

Table 14. The test level and number of components corresponding to the maximum accuracy for Jackknife PLS method when q=100.

In general, after comparing the results form Table 11-14 (q=100) with the results form Table 4-7 (q=25), both the Truncation PLS method and the Jackknife PLS method achieved a satisfying minimum RMSEP, which was approaching to the minimum achievable RMSEP. When q=100, both of the methods had smaller accuracies of variable selection than those from previous experiment when q=25 in most designs of datasets.

After comparing the information in Table 13 and Table 14 where q=100, to achieve the maximum accuracy, Jackknife PLS needed larger test level, whereas a small truncation level was still preferred in Truncation PLS.

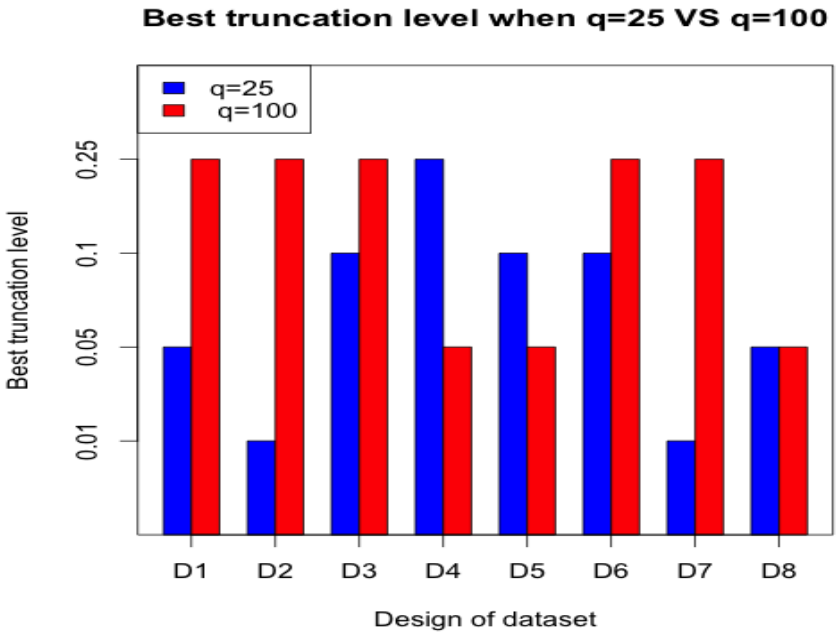

Figure 11. Comparison of the best truncation level which leads to minimum RMSEP for 8 designs of datasets with different q.

This bar plot compares the best truncation level of Truncation PLS which leads to minimum RMSEP when q = 25 with the one when q = 100 in every design of datasets. The horizontal axis indicates the 8 designs of datasets. The vertical axis indicates 4 truncation levels in this research. The scales in the vertical axis are adjusted to show the distinctness better.

The bar plot of Figure 11 shows that for the truncation PLS to achieve the minimum RMSEP, a larger truncation level is preferred when q=100 in most datasets except design 4 and design 5.

CHAPTER 5 Discussion of results

5.1 The effect of the factors and their interactions

By applying truncation PLSR on the 8 different designs of simulated datasets, the performance of truncation-PLS method might be evaluated by RMSEP. The output of the analysis of variance for the linear model (Table 3) with a response of RMSEP shows that $comp, \gamma,$ and R^2 affect the RMSEP significantly, so do the interactions between pos and $comp$, pos and γ, $comp$ and γ, α and γ, & α and R^2. Figure 1 and Figure 2 demonstrate the detail of the effect. After considering the information from Figure 1 and Figure 2, we might draw conclusions as follows.

1. The effect of R^2

   It is easier for truncation PLSR to make good prediction when a dataset has a higher coefficient of determination (R^2). Likewise, this is obviously also true for most other methods, because we can only make a prediction with the limited information contained in the datasets. Hence, the minimum achievable RMSEP equals to $\sqrt{1-R^2}$, which is lower when the coefficient of determination increases.

2. Interaction between pos and γ

   $relpos = (4,5)$ means that the relevant components have smaller variances.

   More components are needed to get the minimum RMSEP when the relevant components have smaller variance (smaller eigenvalue), especially when there is a great disparity in the eigenvalues of the components.

   The reason is the below.

$$cov(z_i, y) = corr(z_i, y) \cdot \sqrt{var(z_i)} \cdot \sqrt{var(y)}$$

   where z_i is a component ; $cov(z_i, y)$ is the covariance of response y and component z_i; $var(y)$ is the variance of response y; $var(z_i)$ is the variance of component z_i; $corr(z_i, y)$ is the correlation of response y and component z_i.

PLS method firstly chooses the component with largest $cov(z_i, y)$. Since $var(y)$ is constant in a dataset, if a component has smaller $var(z_i)$, even if $corr(z_i, y)$ is large, $cov(z_i, y)$ could still be too small to be chosen by PLS at the prior stage. For certain, the larger the disparity in the eigenvalues of the components is, the more extents the selection process could be impacted by $var(z_i)$.

3. The effect of γ

   Conversely, when the eigenvalues of the components are relatively even (as for $\gamma = 0.1$), the variance of the components does not impact the selection process much. Hence, the components chosen by PLS at the prior stage are more likely to be of higher correlation to the response. If we continually select more components into the model, it could be noisy. It is also the reason why in Figure 1 where $\gamma = 0.1$, the RMSEP values are smaller with less components but increased while using more components.

   The same tendency is shown in Figure 5 for Jackknife PLS method, whereas the increment of RMSEP is slower than those for truncation PLSR when more components are used. This is probably due to the fact that Jackknife PLS provides the second chance to fit the model with selected variables, and then makes prediction with its best number of components chosen by cross-validation. Therefore, even if some irrelevant components are included in the model at the latter stage, the damage could be compensating by choosing only the best number of components for prediction.

   When $\gamma = 0.1$, there is a larger error as the number of components increases, especially when an extremely small truncation level is chosen such as α=0.01 in this research. The reason is as follows.

   As it is shown in Figure 7 and Figure 8, the truncation PLS method tends to bring more irrelevant variables when using more components anyhow. When $\gamma = 0.1$, the predictors in X are relatively independent. To form the most relevant component, the normal PLS tends to select more variables by putting more even loading weights

on them. However, a too small truncation level toughly makes it select only a few of them while ignoring the others.

When $\gamma = 0.9$, the components chosen by PLS at the prior stage might not be the most correlated to the response; however after the Lenth truncation process, the most important variables in the component are retained. After several components were added into the model, we reached more or less the same RMSEP value as we got in the prior stage when $\gamma = 0.1$.

4. The effect of α

   In general, the truncation level (α) does not show any significant level as a main factor. But from the two plots at the bottom of the figure where the $pos = (4,5)$ and $\gamma = 0.9$, which is the most difficult situation for prediction, the truncation level (α) shows an effect on RMSEP. With α =0.01, the method reaches a satisfying RMSEP by using only one component.

   The reason is similar as before. When $\gamma = 0.9$ and $pos = (4,5)$, the components chosen by PLS at the prior stage might not be the most correlated to the response, actually it is the worst situation in this research to get a correlated component. But the Lenth truncation process helped here to retain the most important variables in the component. Therefore, as it is shown in Table 4, when $\gamma = 0.9$ and $pos = (4,5)$, it is better to choose a small truncation level (α=0.01 or 0.05 in this research), because it could be more irrelevant variables in one component.

5. The effect of *comp*

   After comparing the best number of components for Truncation-PLS method in Figure 9, the best number of components, which is used to achieve a minimum RMSEP, is not always identical to the one leading to the maximum accuracy of variable selection. Sometimes, although a certain number of components for a highest accuracy are selected, the Truncation PLS tends to select more components to make the best prediction, especially in the datasets with a larger γ and small variances in the relevant components such as design 7 and design 8. Table 8, Table 9,

and Table 10 demonstrate some extreme examples in this research showing the inconsistency. In order to make a good prediction, it is more important for Truncation PLS to have a high sensitivity in variable selection than a high specificity, due to the property that PLS algorithm can alleviate the impact of irrelevant variables to some extents by assigning small loading weights to them.

6. The effect of q

   In general, the true number of relevant predictors q did not show a significant effect on prediction in both Truncation PLS and Jackknife PLS models. However, when there are more relevant predictors in the dataset, the accuracies of variable selection for both methods decreased.

   In the datasets with larger q, to achieve the maximum accuracy, Jackknife PLS needs larger test level α, whereas a small truncation level is still preferred in Truncation PLS. Jackknife PLS is a wrapper method, which has iterating procedures between model fitting and variable selection. The number of selected variables is decided by the test level directly. Truncation PLS is an embedded method, in which variables are selected for every component. Hence, number of selected variables is not only related to the truncation level but also the number of components and the distribution of all the loading weights. But still, to achieve the minimum RMSEP, a larger truncation level is preferred in Truncation PLS when there are more relevant predictors in datasets.

5.2 Comparison with Jackknife-PLS method

We have noted from Figure 6 that Jackknife method has more problems than Truncation PLS method when $\gamma = 0.9$ and $pos = (4,5)$. Both of the methods often reach more or less the same minimum RMSEP, but Jack-knife method needs more components.

The phenomenon is due to the different variable selection process in the two methods. In order to select variables in every component, truncation-PLS method uses the loading weight as a filter, which is proportional to the covariance of X and y.

Although a component used in the prior stage is not relevant, the variables in the component most relevant to the response could be retained. On the other hand, Jackknife method selects variables by the coefficient obtained by normal PLS. If some components used in the prior stage are not relevant, normal PLS might produce some inaccurate coefficients. As a result, the variable selection process in Jackknife method does not make things better at the earlier stage until the relevant components are included into the model. The two bottom plots in Figure 8 reveal the fact that Jackknife method does not have a good performance in variable selection until more components are used.

5.3 Conclusion

1. The best truncation level for prediction will depend on the number of true relevant predictors q. If q is smaller in the comparison with p, probably a small α is preferred. Conversely, if q is larger in comparison with p, probably a large α is preferred. Anyway, the best truncation level will vary from one case to another. In practice, the value of α must be determined by cross-validation.

2. The simulation in this research also confirmed that Truncation-PLS increases the number of selected variables for every component, while Jackknife PLS keeps a more constant size of the set of selected variables.

3. Considering the information of minimum RMSEP and maximum accuracy of variable selection obtained by the two methods, both of them performed well and produced satisfying values. However, the truncation PLS showed a better capability of dealing with datasets of high multicollinearity in X-variables and smaller variance in its relevant component.

4. The truncation-PLS method is more efficient than Jackknife PLS from the aspect of calculation and time consumption, due to the fact that the Jackknife PLS method fits a PLS model twice and runs the cross-validation twice.

5.4 Further research

Besides the parameters investigated in this research, there are more parameters determining the property of the simulated datasets, such as the number of observations used for training data (noted n), the number of predictors (noted p), the number of observations used for testing data (noted $ntest$), and the number of relevant components (noted m). As it is verified in many other researches, a comparatively smaller n to p could lead to larger variances of estimation, thus frustrate prediction sometimes. Since the Truncation PLS estimates a model with some components instead of variables, hypothetically this frustration could be diminished somehow. It might be interesting to observe the effects of those parameters in the further research. Moreover, the range of parameter setting is limited, only 2 levels for each in this research. In practice, more levels of α and $comp$ could be tried out in applying Truncation PLS.

Appendices

Tables

ANOVA of RMSEP

```
Call:
lm(formula = y ~ (R2 + pos + gamma + alpha + comp)^2, data = rmsep.tmp)

Residuals:
     Min           1Q           Median           3Q           Max
-0.23565      -0.02847       0.00297       0.02950       0.62607

Coefficients:
                    Estimate Std.    Error    t value     Pr(>|t|)
(Intercept)              0.86238     0.03973   21.705     < 2e-16 ***
R2(0.9)                 -0.47201     0.03474  -13.585     < 2e-16 ***
pos(1)                  -0.01442     0.03474   -0.415     0.678634
gamma(0.9)              -0.22063     0.03474   -6.350     1.58e-09 ***
alpha(0.05)             -0.13815     0.04520   -3.056     0.002565 **
alpha(0.1)              -0.14125     0.04520   -3.125     0.002059 **
alpha(0.25)             -0.11947     0.04520   -2.643     0.008903 **
comp(2)                  0.10958     0.05099    2.149     0.032917 *
comp(3)                  0.26882     0.05099    5.272     3.68e-07 ***
comp(4)                  0.42153     0.05099    8.267     2.44e-14 ***
comp(5)                  0.53515     0.05099   10.495     < 2e-16 ***
comp(6)                  0.66749     0.05099   13.090     < 2e-16 ***
comp(7)                  0.68267     0.05099   13.388     < 2e-16 ***
comp(8)                  0.90464     0.05099   17.741     < 2e-16 ***
R2(0.9):pos(1)           0.02980     0.01927    1.546     0.123674
R2(0.9):gamma(0.9)       0.22968     0.01927   11.918     < 2e-16 ***
R2(0.9):alpha(0.05)      0.07139     0.02726    2.619     0.009532 **
R2(0.9):alpha(0.1)       0.06868     0.02726    2.520     0.012573 *
R2(0.9):alpha(0.25)      0.10080     0.02726    3.698     0.000285 ***
R2(0.9):comp(2)         -0.08861     0.03854   -2.299     0.022606 *
R2(0.9):comp(3)         -0.15786     0.03854   -4.095     6.26e-05 ***
R2(0.9):comp(4)         -0.21042     0.03854   -5.459     1.50e-07 ***
R2(0.9):comp(5)         -0.24179     0.03854   -6.273     2.38e-09 ***
R2(0.9):comp(6)         -0.25348     0.03854   -6.576     4.64e-10 ***
R2(0.9):comp(7)         -0.25245     0.03854   -6.549     5.37e-10 ***
R2(0.9):comp(8)         -0.30492     0.03854   -7.911     2.13e-13 ***
pos(1):gamma(0.9)        0.10411     0.01927    5.402     1.98e-07 ***
pos(1):alpha(0.05)       0.05521     0.02726    2.026     0.044226 *
pos(1):alpha(0.1)        0.08417     0.02726    3.088     0.002318 **
pos(1):alpha(0.25)       0.10003     0.02726    3.670     0.000316 ***
```

```
pos(1):comp(2)          -0.03740    0.03854   -0.970     0.333120
pos(1):comp(3)          -0.07562    0.03854   -1.962     0.051245 .
pos(1):comp(4)          -0.11881    0.03854   -3.082     0.002363 **
pos(1):comp(5)          -0.12945    0.03854   -3.358     0.000949 ***
pos(1):comp(6)          -0.14566    0.03854   -3.779     0.000211 ***
pos(1):comp(7)          -0.14606    0.03854   -3.789     0.000203 ***
pos(1):comp(8)          -0.18966    0.03854   -4.921     1.88e-06 ***
gamma(0.9):alpha(0.05)   0.21703    0.02726    7.963     1.56e-13 ***
gamma(0.9):alpha(0.1)    0.24959    0.02726    9.157     < 2e-16 ***
gamma(0.9):alpha(0.25)   0.25407    0.02726    9.322     < 2e-16 ***
gamma(0.9):comp(2)      -0.10094    0.03854   -2.619     0.009547 **
gamma(0.9):comp(3)      -0.18178    0.03854   -4.716     4.68e-06 ***
gamma(0.9):comp(4)      -0.27892    0.03854   -7.236     1.14e-11 ***
gamma(0.9):comp(5)      -0.36232    0.03854   -9.400     < 2e-16 ***
gamma(0.9):comp(6)      -0.41324    0.03854  -10.721     < 2e-16 ***
gamma(0.9):comp(7)      -0.44158    0.03854  -11.456     < 2e-16 ***
gamma(0.9):comp(8)      -0.53264    0.03854  -13.819     < 2e-16 ***
alpha(0.05):comp(2)     -0.04451    0.05451   -0.816     0.415260
alpha(0.1):comp(2)      -0.06530    0.05451   -1.198     0.232474
alpha(0.25):comp(2)     -0.07689    0.05451   -1.411     0.160025
alpha(0.05):comp(3)     -0.08746    0.05451   -1.604     0.110307
alpha(0.1):comp(3)      -0.11310    0.05451   -2.075     0.039357 *
alpha(0.25):comp(3)     -0.13939    0.05451   -2.557     0.011345 *
alpha(0.05):comp(4)     -0.11951    0.05451   -2.192     0.029576 *
alpha(0.1):comp(4)      -0.15826    0.05451   -2.903     0.004134 **
alpha(0.25):comp(4)     -0.18375    0.05451   -3.371     0.000910 ***
alpha(0.05):comp(5)     -0.14397    0.05451   -2.641     0.008958 **
alpha(0.1):comp(5)      -0.19518    0.05451   -3.581     0.000437 ***
alpha(0.25):comp(5)     -0.22782    0.05451   -4.179     4.47e-05 ***
alpha(0.05):comp(6)     -0.20448    0.05451   -3.751     0.000234 ***
alpha(0.1):comp(6)      -0.26510    0.05451   -4.863     2.43e-06 ***
alpha(0.25):comp(6)     -0.29741    0.05451   -5.456     1.52e-07 ***
alpha(0.05):comp(7)     -0.17938    0.05451   -3.291     0.001193 **
alpha(0.1):comp(7)      -0.24007    0.05451   -4.404     1.78e-05 ***
alpha(0.25):comp(7)     -0.28213    0.05451   -5.176     5.80e-07 ***
alpha(0.05):comp(8)     -0.26809    0.05451   -4.918     1.90e-06 ***
alpha(0.1):comp(8)      -0.34643    0.05451   -6.355     1.53e-09 ***
alpha(0.25):comp(8)     -0.39012    0.05451   -7.157     1.80e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.07709 on 188 degrees of freedom
Multiple R-squared: 0.9525,
Adjusted R-squared: 0.9356
F-statistic: 56.26 on 67 and 188 DF,  p-value: < 2.2e-16
```

R-code

```
#1. LENTH FUNCTION
    lenth.trunc <- function(w, alpha){
      s0<-1.5*median(abs(w))
      w0<-w[abs(w)<2.5*s0]
      sd<-1.5*median(abs(w0))
      upper<-qnorm((1-alpha/2),0,sd)
      lower<-qnorm(alpha/2,0,sd)
      lo<-sum(w>lower&w<upper)
      w[w>lower&w<upper]<-rep(0,lo)
      idx<-which(w!=0)#idx of improtant variables
     if (lo==length(w)){
        idx<-which.max(abs(w))
        w<-w[idx]}
      return(list(sd=sd,upper=upper,lower=lower,w=w,idx))}

#2. TRUNCATION PLS FUNCTION
    NIPALS<- function(Y, X, ncomp, lenth.alpha){
      X0<-scale(X,scale=FALSE)
      Y0<-scale(Y,scale=FALSE)
      meanX<-attr(X0,"scaled:center")#meanX<-apply(X,2,mean)
      meanY<-attr(Y0,"scaled:center")#meanY<-apply(Y,2,mean)
      m<-ifelse(is.null(dim(Y)), 1, dim(Y)[2])
      n<-dim(X)[1]
      p<-dim(X)[2]
      T<-matrix(nrow=n,ncol=ncomp)
      W<-matrix(nrow=p,ncol=ncomp)
      P<-matrix(nrow=p,ncol=ncomp)
      Q<-matrix(nrow=m,ncol=ncomp)
      U<-matrix(nrow=n,ncol=ncomp)
      B <- array(0, dim = c(p,m, ncomp))
      res <- array(0, dim = c(n,m, ncomp))
      mse<-matrix(nrow=m,ncol=ncomp)
      X<-X0
      Y<-Y0
      for (i in 1:ncomp)
      {
        if (m == 1) {u <- Y;t.old<-0}
        else {
```

```
    u<- Y[,which.max(colSums(Y * Y))]
    t.old <- 0
  }
  repeat{
    w<-t(X)%*%u%*%solve(t(u)%*%u)
      w<-w/as.numeric(sqrt(t(w)%*%w))
    t<-X%*%w
    if(sum(abs((t - t.old)/t))<1.0e-5)break
    else{q<-t(Y)%*%t%*%solve(t(t)%*%t)
         p<-t(X)%*%t%*%solve(t(t)%*%t)
         u<-Y%*%q%*%solve(t(q)%*%q)
         t.old<-t}
  }
   w<-lenth.trunc(w,alpha=lenth.alpha)$w #alpha=0.1
   W[,i]<-w
   T[,i]<-t
   P[,i]<-p
   Q[,i]<-q
   U[,i]<-u
   X<-X-T[,i]%*%t(P[,i,drop=F])
   Y<-Y-T[,i]%*%t(Q[,i,drop=F])
   res[,,i]<-Y
B[,,i]<W[,1:i,drop=F]%*%solve(t(P[,1:i,drop=F])%*%W[,1:i,drop=F])%*%t
(Q[,1:i,drop=F])
   for (j in 1:m){mse[j,i]<-(res[,j,i])%*%res[,j,i]/n}
 }
return(list(coefficients = B, scores = T, loadings = P,
loading.weights = W,
Yscores = U, Yloadings = Q, meanY=meanY, meanX=meanX,
ncomp=ncomp))}

#3. PREDICT FUNCTION
predicttrunc<-function(fit,newX){
#testX0<-newX-fit$meanX
  testX0 <- scale(newX,center=fit$meanX, scale=FALSE)
#newY<-matrix(0,ncol=length(fit$meanY),nrow=dim(newX)[1])
  newY<-array(0,c(nrow=dim(newX)[1],length(fit$meanY),fit$ncomp))
  for (i in 1:fit$ncomp){
   newY[,,i] <- testX0%*%fit$coefficients[,,i]}
 return(newY)}
```

```
#4. RMSEP FUNCTION FOR CONTINUOUS DATA
      rmsep <- function(A, B){
        sqrt(mean((A-B)^2))
      }

#5. ERROR RATE FOR CATEGORICAL DATA
      er<-function(A,B){
        A<-as.vector(A)
        B<-as.vector(B)
        B[B<1.5]=rep(1,sum(B<1.5))
        B[B>=1.5]=rep(2,sum(B>=1.5))
       return(sum(abs(A-B))/length(A))
      }

# 6. CROSS VALIDATION
      comps <- 8
      alphavek <- rev(c(0.25,1.0e-1, 0.05, 0.01))
      #Remove som null variables
      #sumtest <- apply(X,2,sum)
      #keep <- which(sumtest!=0)
      N <- dim(X)[1]
      K <- 10
      segs <- cvsegments(N,K)
      rmsepmat <- matrix(0,length(alphavek), comps)
      for(j in 1:length(alphavek)){
        rmsepvek <- rep(0,comps)
        for(i in 1:comps){
        rmsep.c<-rep(0,K)
        for(k in 1:K){
          testY <- Y[segs[[k]],,drop=F]
          testX <- X[segs[[k]],,drop=F]
          trainY <- Y[-segs[[k]],,drop=F]
       trainX <- X[-segs[[k]],,drop=F]
       trainY0<-scale(trainY,scale=FALSE)
       meanY<-attr(trainY0,"scaled:center")
       testY0 <- scale(testY,center=meanY, scale=FALSE)
       newY0<-predict(fit=NIPALS(trainY,trainX,comps,
      lenth.alpha=alphavek[j]),newX=testX)
```

```
 rmsep.c[k]<-rmsep(testY0, newY0[,,i])
 }
 rmsepvek[i]<-mean(rmsep.c)
  cat(paste("Component",i,", alpha-value",alphavek[j]," finished\n"))
  }
  rmsepmat[j,] <- rmsepvek
}
matplot(t(rmsepmat), type="b")

#7. FIT MODEL AND CALCULATE RMSEP, ACCURACY
library(pls)
#D1 (R2=0.5,pos=(1,2),gamma=0.1)
sim1<-
relsim(n=50,p=500,m=2,q=25,relpos=c(1,2),gamma=0.1,R2=0.5,ntest=100)
alpha<-c(0.01,0.05,0.1,0.25)
#betamat1<-array(0,dim=c(500,8,4))
rrmsep1<-rmsep1<-racc1<-acc1<-array(0,dim=c(10,4,8))
for (j in 1:4)
{
  for(i in 1:10){
    sim<-
relsim(n=50,p=500,m=2,q=25,relpos=c(1,2),gamma=0.1,R2=0.5,ntest=100,s
im=sim1)
    fit<-NIPALS(sim$Y,sim$X,8,alpha[j])
    fit0<-plsr(sim$Y~sim$X,ncomp=8,validation="LOO",jackknife=TRUE)
  #betamat1[,,j]<-betamat1[,,j]+drop(fit$coefficient)
    tru1<-sim1$relpred
    for(k in 1:8){
    #variable selection by lenth-trunc
    est1<-which(drop(fit$coefficient[,,k])!=0)
    tp1<-length(intersect(tru1,est1))#number of true positive
    tn1<-500-length(unique(c(tru1,est1)))#number of true negative
    acc1[i,j,k]<-(tp1+tn1)/500
    predY<-predicttrunc(fit,sim$TESTX)
    rmsep1[i,j,k]<-rmsep(drop(sim$TESTY),predY[,,k])
    #variable selection by jackknife
    p.beta<-jack.test(fit0,k)
    rest1<-order(p.beta$pvalues)[1:(alpha[j]*500)]
    rtp1<-length(intersect(tru1,rest1))
    rtn1<-500-length(unique(c(tru1,rest1)))
```

```
    racc1[i,j,k]<-(rtp1+rtn1)/500
    #refit model with jackknife selected variables
    refit<-plsr(sim$Y~sim$X[,rest1],ncomp=min(k,length(rest1)-
1),validation="LOO")
    bestk <- which.min(refit$validation$PRESS)
    tX <- sim$TESTX[,rest1,drop=FALSE]
    rpredY<-predict(refit,tX,ncomp=bestk)
    rrmsep1[i,j,k]<-rmsep(drop(sim$TESTY),rpredY[,,1])}
    cat(paste("iteration j=",j,"i=",i,"\n"))}
  #betamat1[,,j]<-betamat1[,,j]/10 # p*ncomp
  }
rmsep1<-apply(rmsep1,c(2,3),mean)
rrmsep1<-apply(rrmsep1,c(2,3),mean)
acc1<-apply(acc1,c(2,3),mean)
racc1<-apply(racc1,c(2,3),mean)

#8. PLOT ACCURACY RESULTS
plot(1:8,acc1[1,],type="b", ylim=c(0,1),ylab="accuracy",
xlab="components",main="D1:R2=0.5,pos=(1,2),gamma=0.1")
points(1:8,acc1[2,],type="b", col=2)
points(1:8,acc1[3,],type="b", col=3)
points(1:8,acc1[4,],type="b", col=4)
points(1:8,racc1[1,],type="b", pch=4)
points(1:8,racc1[2,],type="b", pch=4,col=2)
points(1:8,racc1[3,],type="b", pch=4,col=3)
points(1:8,racc1[4,],type="b", pch=4,col=4)
legend(0.5,0.55,legend=c("alpha=0.01",
                         "alpha=0.05",
                         "alpha=0.1",
                         "alpha=0.25"),lty=1,col=1:4,bty="n")
legend(x="bottomleft",legend=c("   lenth plsr","   jackknife
plsr"),pch=c(1,4),bty="n")

#9. PLOT RMSEP RESULTS
#tp1
plot(0:8,c(1,rmsep1[1,]) ,type="b", ylim=c(0,2.0),ylab="fitted
rmsep", xlab="components",main="alpha=0.01,gamma=0.1")
points(0:8,c(1,rmsep2[1,]),type="b", col=2)
points(0:8,c(1,rmsep5[1,]),type="b", col=3)
```

```
points(0:8,c(1,rmsep6[1,]),type="b", col=4)
points(0:8,c(1,rrmsep1[1,]),type="b",pch=4,col=1)
points(0:8,c(1,rrmsep2[1,]),type="b",pch=4,col=2)
points(0:8,c(1,rrmsep5[1,]),type="b",pch=4,col=3)
points(0:8,c(1,rrmsep6[1,]),type="b",pch=4,col=4)
legend(x="topleft", legend=c("R2=0.5, relpos=c(1,2)",
                             "R2=0.9, relpos=c(1,2)",
                             "R2=0.5, relpos=c(4,5)",
                             "R2=0.9,
relpos=c(4,5)"),lty=1,col=1:4,bty="n")
legend(0,1.57,legend=c("   truncation plsr","   jackknife
plsr"),pch=c(1,4),bty="n")

#10. GROUPED BAR PLOT FOR TRUNCATION PLS
barplot(rbind(c(1,1,2,2,1,3,5,5), c(1,1,1,1,1,1.5,2,1)), main="Best
number of components for truncation pls",
        ylab="Best Number of Components",
        xlab=c("Design of dataset"),
        col=c("blue","red"),
        , legend=c("for prediction","for variable selection"),
beside=TRUE,
        args.legend=list(x="topleft"))
axis(1, at=seq(2,23,by=3), labels=paste("D",1:8, sep=""))

#11. GROUPED BAR PLOT FOR JACKKNIFE PLS
barplot(rbind(c(2,2,2,2,1,3,7,8), c(1,2.5,1,3,2,4,7,4.5)), main="Best
number of components for jackknife pls",
        ylab="Best Number of Components",
        xlab=c("Design of dataset"),
        col=c("blue","red"),
        legend=c("for prediction","for variable selection"),
beside=TRUE,
        args.legend=list(x="topleft"))
axis(1, at=seq(2,23,by=3), labels=paste("D",1:8, sep=""))

#12. GROUPED BAR PLOT FOR q=25 VS q=100
barplot(rbind(c(2,1,3,4,3,3,1,2), c(4,4,4,2,2,4,4,2)),
        main="Best truncation level when q=25 VS q=100",
```

```
        ylab="Best truncation level",
        xlab="Design of dataset",
        col=c("blue","red"),
        legend=c("q=25","q=100"), beside=TRUE,
        args.legend=list(x="topleft"),
        ylim=c(0,5),axes=F)
axis(1, at=seq(2,23,by=3), labels=paste("D",1:8, sep=""))
axis(2,at=1:4, labels=c(0.01, 0.05, 0.1, 0.25))
```

References

A. J. MILLER 2002. *Subset Selection in Regression,* London, Chapman and Hal.

AKAIKE, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19**,** 716-723.

BARKER, M. & RAYENS, W. 2003. Partial least squares for discrimination. *Journal of Chemometrics,* 17**,** 166-173.

CHUN, H. & KELES, S. 2010. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society Series B-Statistical Methodology,* 72**,** 3-25.

DANIEL, C. W., F. 1980. *Fitting Equations to Data (Rev. ed.),* New York, Wiley & Sons,lnc.

EFROYMSON 1960. *Multiple regression analysis,* New York, Wiley.

HAN, Q. J., WU, H. L., CAI, C. B., XU, L. & YU, R. Q. 2008. An ensemble of Monte Carlo uninformative variable elimination for wavelength selection. *Analytica Chimica Acta,* 612**,** 121-125.

HELLAND, I. S. 1988. On the Structure of Partial Least-Squares Regression. *Communications in Statistics-Simulation and Computation,* 17**,** 581-607.

INDAHL, U. G., LILAND, K. H. & NAES, T. 2009. Canonical partial least squares-a unified PLS approach to classification and regression problems. *Journal of Chemometrics,* 23**,** 495-504.

JOLLIFFE, I. T. 1982. A Note on the Use of Principal Components in Regression. *Applied Statistics-Journal of the Royal Statistical Society Series C,* 31**,** 300-303.

K. HASEGAWA, Y. M., K. FUNATSU, 1997. GA strategy for variable selection in QSAR studies: GA-based PLS analysis of calcium channel antagonists. *Journal of Chemical Information and Computer Sciences* 306–310.

LE CAO, K. A., ROSSOUW, D., ROBERT-GRANIE, C. & BESSE, P. 2008. A Sparse PLS for Variable Selection when Integrating Omics Data. *Statistical Applications in Genetics and Molecular Biology,* 7.

LILAND, K. H., HØY, M., MARTENS, H. & SÆBØ, S. 2013. Distribution based truncation for variable selection in subspace methods for multivariate regression. *Chemometrics and Intelligent Laboratory Systems,* 122**,** 103-111.

LINDGREN, F., GELADI, P., BERGLUND, A., SJOSTROM, M. & WOLD, S. 1995. Interactive Variable Selection (Ivs) for Pls .2. Chemical Applications. *Journal of Chemometrics,* 9**,** 331-342.

LINDGREN, F., GELADI, P., RANNAR, S. & WOLD, S. 1994. Interactive Variable Selection (Ivs) for Pls .1. Theory and Algorithms. *Journal of Chemometrics,* 8**,** 349-363.

MALLOWS, C. L. 1973. Some Comments on Cp. *Technometrics* 15**,** 661–675.

MARTENS H, N. T. 1989. *Multivariate Calibration,* New York, Wiley.

NELDER, J. A. & WEDDERBU.RW 1972. Generalized Linear Models. *Journal of the Royal Statistical Society Series a-General,* 135**,** 370-&.

QUENOUILLE, M. H. 1949. Problems in Plane Sampling. *Annals of Mathematical Statistics,* 20**,** 355-375.

QUENOUILLE, M. H. 1956. Notes on Bias in Estimation. *Biometrika,* 43**,** 353-360.

RUBIO, H. & FIRINGUETTI, L. 2002. The distribution of stochastic shrinkage parameters in ridge regression. *Communications in Statistics-Theory and Methods,* 31**,** 1531-1547.

SÆBØ, S. 2014. relsim: Package for data simulation and design of computer experiments.(to be published)

SÆBØ, S., ALMØY, T., AARØE, J. & AASTVEIT, A. H. 2008. ST-PLS: a multi-directional nearest shrunken centroid type classifier via PLS (vol 22, pg 54, 2008). *Journal of Chemometrics,* 22**,** 423-423.

SCHWARZ, G. E. 1978. "Estimating the dimension of a model". 6 (2): 461–464. doi:10.1214/aos/1176344136. MR 468014. *Annals of Statistics* 6**,** 461–464.

TAHIR MEHMOOD , K. H. L., LARS SNIPEN, SOLVE SÆBØ, 2012. A review of variable selection methods in Partial Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems,* 118**,** 62-69.

TIBSHIRANI, R. 2011. Regression shrinkage and selection via the lasso: a retrospective. *Journal Of The Royal Statistical Society Series B-Statistical Methodology* 73**,** 273-282.

TREVOR HASTIE, R. T., JEROME FRIEDMAN 2008. *The elements of statistical learning,* California, springer.

TUKEY, J. W. 1958. Bias and confidence in not quite large samples. *The Annals of Mathematical Statistics*.

WOLD, H. 1966. *"Estimation of principal components and related models by iterative least squares. ,* New York, Academic Press.

WOLD, S., RUHE, A., WOLD, H. & DUNN, W. J. 1984. The Collinearity Problem in Linear-Regression - the Partial Least-Squares (Pls) Approach to Generalized Inverses. *Siam Journal on Scientific and Statistical Computing,* 5**,** 735-743.

WOLD., H. 1973. *Nonlinear Iterative Partial Least Squares (NIPALS) Modelling: Some Current Developments Multivariate Analysis. Academic Press, New york, 1973.,* New york, Academic Press.

ZOU, H. & HASTIE, T. 2005. Regularization and variable selection via the elastic net (vol B 67, pg 301, 2005). *Journal of the Royal Statistical Society Series B-Statistical Methodology,* 67**,** 768-768.

Norges miljø- og
biovitenskapelige
universitet

Postboks 5003
NO-1432 Ås
67 23 00 00
www.nmbu.no