Norwegian University of Life Sciences
Faculty of Veterinary Medicine and Biosciences
Department of Chemistry, Biotechnology and Food Science

Master Thesis 2015
60 credits

# A simulation study: Hierarchical PLS for multi-group classification

# En simuleringsstudie: Hierarkisk PLS for multi-gruppe klassifisering

Hanne Brit Hetland

**Abstract**

Hierarchically Ordered Taxonomic Partial Least Squares (Hot PLS) is a method for classifying data in a hierarchical structure. Since Hot PLS is a relatively new method, we want to study strengths and weaknesses of this. This was done by simulated data with known parameters by using the R package, Simrel.

The simulated data was then classified by Hot PLS. Classification error was used as the measure on how good the a method is to classify the data. For finding out which effect the different simulated parameters had on the classification error an ANOVA model was made, where the classification error was the response and the simulatated parameters and methods was the treatments. The simulated data were also classifies by other classifiers PLS, LDA, QDA and KNN, so one could check if the Hot PLS did perform better than the other classifiers. First the Hot PLS was only compared with PLS.

The results from these analysis show us that the Hot PLS is a good method for classifying data which has a hierarchical structure.

**Sammendrag**

Hierarchically Ordered Taxonomic Partial Least Squares (Hot PLS) er en metode for å klassifisere data som har en hierarkisk struktur. Siden Hot PLS er en relativt ny metode, ønsker man å studere styrkene og svakhetene ved denne metoden. Dette ble gjort ved å simulere data med kjente simuleringsparametre ved hjelp av R-pakken, Simrel.

De simulerte dataene ble deretter klassifisert av Hot PLS og klassifikasjonsfeil ble brukt som mål på hvor god metoden er på å klassifisere dataene. For å finne ut hvilken virkning på de ulike simuleringsparameterne har på klassifikasjonsfeil ble en ANOVA modell laget, hvor klassifikasjonsfeil var responsen og simuleringsparameterne og metodene var forklarende variabler. De simulerte dataene var også klassifisert av andre klassifisering metoder, disse metodene er PLS, LDA, QDA og KNN. Dette ble gjort slik at man kunne sjekke om Hot PLS gjorde det bedre enn de andre klassifiseringsmetodene. Først ble Hot PLS bare sammenlignet med PLS.

Resultatene fra disse analyser viser at Hot PLS er en god metode for å klassifisere data som har en hierarkisk struktur.

# Acknowledgement

This thesis written at IKBM at NMBU as a part of the Biostatistics group.

I would like to express my gratitude to my supervisor Solve Sæbø for all the guidance and help. I also would thank my co-supervisor Trygve Almøy. Thank you for always have the door open and taking time answer my questions.

Kristian Liland also deserves a big thank you for leting my use his R code for the Hot PLS.

To my boyfriend, Magne, thanks for giving me many great tips and for pushing me to work harder. Betty, thank you for taking the time to read through the thesis and give helpful feedback. I would also like to express my gratitude to my parents who has always encouraged me to study.

---

Hanne Brit Hetland

# Contents

# Chapter 1

# Introduction

## 1.1 Problem

The goal of this thesis is to study the strengths and the weaknesses of Hierarchically Ordered Taxonomic Partial Least Squares (Hot PLS) method for classification [Liland et al. 2014]. Hot PLS may be used in cases where the data has a hierarchical structure. Hot PLS starts at the top level and works its way trough the structure to the lowest level. Unlike other classifiers, Partial Least Squares [Wold 1966], Linear discriminant analysis (LDA) [e.g. Johnson and Wichern 2007a], Quadratic discriminant analysis (QDA) [e.g.Johnson and Wichern 2007b], k-Nearest Neighbors (KNN) [e.g.James et al. 2013a], which will classify on the lowest level. In this thesis the data will be simulated so that one can predetermine parameters to be what you want them to be. The results of the Hot PLS will be compared with the results from other classifiers PLS, LDA, QDA, KNN.

## 1.2 Classification

In statistics classification is used to predict some qualitative response. Another word for qualitative variable is categorical variable. Categorical variables can be, for instance be the eye colour or the gender of an individual. To predict the qualitative response there are different classifiers such as LDA, QDA, and KNN, to mention a few.

The classification begins with one or more observed input features, x-variables, which then are ran through a classifier. The classifier will chooses which class the observation will belong to. Besides other approaches, the class which gives the observation the highest probability can be chosen. Binary classification is the most known classification scheme, where the observations are classified as either A or B. Occasionally the problem may requires a classification into more than two groups, and which is known as multi-group classification. Such a classification is a frequent problem in machine learning, since it can be hard to separate one class from another. One strategy to deal with multi-group classification is OvA (One versus All) [Har-peled et al. 2002]. This method makes use of the standard binary classifiers to find the correct class. OvA assumes that there single separators for each class to separate it from all the others. By using this, the OvA implies a WTA-strategy (winner takes all). WTA uses a real-value function to determine which class the observations belong to. Machine learning is divided into two groups, supervised and unsupervised learning. Unsupervised learning models seek for natural groups in the observations. While in supervised learning the classes are known, and one divides the observations into test data and training data. The training data is used to fit the classifier and then it is

tested on the test data.

### 1.2.1 Classification rules

Classification rules are also known as classifiers. In the following some classifiers which are used later on for comparison in this thesis are described.

### 1.2.2 Linear discriminant analysis (LDA)

The LDA classifier is a stable classifier, even if when the number $n$ of observations is low, and the number explanatory variables, $p$, are approximately normal distributed in each of the classes. This means that LDA is working even if the condition for multinomial distribution is not completely fulfilled and there are just a few observations available. LDA is also working well when the observations are linearly separated. LDA is based on the assumption that the covariance matrix for each class to equal, $\boldsymbol{\Sigma_1} = \boldsymbol{\Sigma_2} = \ldots = \boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$, where $k$ is the number of groups. This give the multivariate Gaussian density defined as

$$f_j(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^t \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)}$$

where $p$ is the number of explanatory variables, $\boldsymbol{x}$ is a vector of an observation, $\boldsymbol{\Sigma}$ is the covariance matrix of $\boldsymbol{X}$, $cov(\boldsymbol{X})$ and the $\boldsymbol{\mu}_j$ is the expected mean for class j. In a LDA case with two classes and known $\boldsymbol{\mu}$s and the $\boldsymbol{\Sigma}$ are distributed as:

For class 1

$$\boldsymbol{x}_1 \sim N_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = f_1(\boldsymbol{x}_1)$$

For class 2

$$\boldsymbol{x}_2 \sim N_p\left(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}\right) = f_1\left(\boldsymbol{x}_2\right)$$

A new observation in a two classes case belongs to class 1 if and only if

$$f_1(\boldsymbol{x}_1) > f_2(\boldsymbol{x}_2)$$

$$\boldsymbol{x}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_1^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 > \boldsymbol{x}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \frac{1}{2}\boldsymbol{\mu}_2^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2$$

In most cases $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}$ are unknown, so that these parameters have to be estimated. If $n_1 + n_2 - 2 \geq p$ one can use

$$\begin{aligned} \overline{\boldsymbol{x}}_1 &= \frac{1}{n_1}\sum_{i=1}^{n_1}\boldsymbol{x}_{1j} \quad \boldsymbol{S_1} = \frac{1}{n_1 - 1}\sum_{i=1}^{n_1}\left(\boldsymbol{x}_{1j} - \overline{\boldsymbol{x}}_1\right)\left(\boldsymbol{x}_{1i} - \overline{\boldsymbol{x}}_1\right)^t \\ \overline{\boldsymbol{x}}_2 &= \frac{1}{n_2}\sum_{i=1}^{n_2}\boldsymbol{x}_{2i} \quad \boldsymbol{S_2} = \frac{1}{n_2 - 1}\sum_{i=1}^{n_2}\left(\boldsymbol{x}_{2i} - \overline{\boldsymbol{x}}_2\right)\left(\boldsymbol{x}_{2i} - \overline{\boldsymbol{x}_2}\right)^t. \end{aligned} \tag{1.1}$$

where $\overline{\boldsymbol{x}}_1$ is a vector with the mean of group 1, $\overline{\boldsymbol{x}}_2$ is a vector the mean of group 2, $\boldsymbol{x}_{1i}$ is x vector for group 1 and observation $i$, $\boldsymbol{x}_{1i}$ is x vector for group 2 and observation $j$, $\boldsymbol{S}_1$ is the estimated variance for group 1 and $\boldsymbol{S}_2$ is the estimated variance for group 2. Since in LDA $\boldsymbol{\Sigma}$ is assumed to be equal for both groups the estimated $\mathbf{S_1}$ and $\mathbf{S_2}$ be combined to one matrix, $\boldsymbol{S}_{pooled}$

$$\boldsymbol{S}_{pooled} = \frac{\left(n_1 - 1\right)\boldsymbol{S_1} + \left(n_2 - 1\right)\boldsymbol{S_2}}{\left(n_1 - 1\right) + \left(n_2 - 1\right)}.$$

When a new observation are added to data it belongs to class 1 if

$$\left(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2\right)^t\boldsymbol{S}_{pooled}^{-1}\boldsymbol{x}_0 - \frac{1}{2}\left(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2\right)^t\boldsymbol{S}_{pooled}^{-1}\left(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2\right) \geq ln\left(\frac{\pi_2}{\pi_1}\right) \tag{1.2}$$

if equation (1.2) does not hold the new observation belongs to group 2. Where the $\frac{\pi_2}{\pi_1}$ is the prior probability ratio, which is used if the probability of belonging to one class is higher than the probability of belonging to the other

class. In most cases with two classes the prior is set equal to 0.5, by doing so the probabilities of belonging to a certain class are equal and thus the prior cancels.

As mentioned before the LDA classifier works well with small $n$, nevertheless it requires $n > p$. If $n < p$ the inverse covariance matrix, $\mathbf{\Sigma}^{-1}$ cannot be found hence the calculation of the probability is not possible. During the analysis of this thesis LDA is used even if $n < p$. In order to use LDA under this condition it is necessary to use principal components (PCA) instead of x. The principal components will have $a$ components where $a < n$. In this thesis $a = 8$ since 8 always will be smaller than the smallest $n$ used in this thesis. One also expects that 8 components will retain most of the variation in $\boldsymbol{X}$. Later on, in subsection 1.4.1, principal components will be explained in more details.

### 1.2.3 Quadratic discriminant analysis (QDA)

The LDA classifier can be used when the covariances are equal in all $k$ classes i.e. $\mathbf{\Sigma_1} = \mathbf{\Sigma_2} = \ldots = \mathbf{\Sigma}_k = \mathbf{\Sigma}$. In cases where this is not true, then the covariances are assumed different from each other, $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2 \neq \ldots \neq \mathbf{\Sigma}_k$, $k$ is the number of classes, the LDA classifier cannot be used which lead to the QDA classifier. QDA is similar to LDA since both classifiers assume the classes the classes to be multivariate normal distributed. Figure 1.1 shows an example of two distributions with different means and variances.

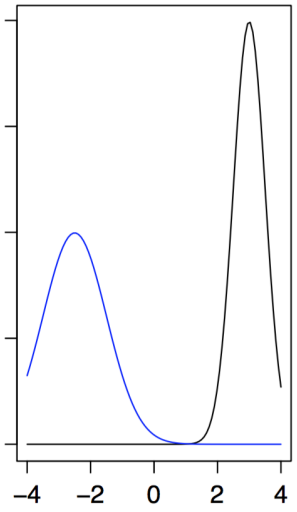



Figure 1.1: *The density plot showing two probability distribution. The blue one has* $\mu = -2.5,\ \sigma = 1$*, and the black one has* $\mu = 3,\ \sigma = 0.5$

For a QDA case with two classes, they will be distributed as

For class 1

$$\boldsymbol{x_1} \sim N_p\left(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right) = f_1\left(\boldsymbol{x_1}\right)$$

For class 2

$$\boldsymbol{x_2} \sim N_p\left(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right) = f_2\left(\boldsymbol{x_2}\right)$$

where $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are the covariance matrices for class 1 and class 2, respectively. If the model parameters are known, a new observation is classified to class 1 if

$$f_1\left(\boldsymbol{x_1}\right) > f_2\left(\boldsymbol{x_2}\right)$$

$$\frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\mid \mathbf{\Sigma_1} \mid^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\boldsymbol{x}^* - \mathbf{1}\mu_1)^t \mathbf{\Sigma}_1^{-1} (\boldsymbol{x}^* - \mathbf{1}\mu_1)} > \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\mid \mathbf{\Sigma_2} \mid^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\boldsymbol{x}^* - \mathbf{1}\mu_2)^t \mathbf{\Sigma}_2^{-1} (\boldsymbol{x}^* - \mathbf{1}\mu_2)} \tag{1.3}$$

In order to simplify equtaion (1.3) the logarithm is taken which leads to

$$\begin{aligned} &-\frac{1}{2}\log\left(\mid \mathbf{\Sigma}_1 \mid\right) - \frac{1}{2}\left(\boldsymbol{x}^* - \mathbf{1}\mu_1\right)^t \mathbf{\Sigma}_1^{-1} \left(x^* - \mathbf{1}\mu_1\right) \\ &> -\frac{1}{2}\log\left(\mid \mathbf{\Sigma}_2 \mid\right) - \frac{1}{2}\left(\boldsymbol{x}^* - \mathbf{1}\mu_2\right)^t \mathbf{\Sigma}_2^{-1} \left(\boldsymbol{x}^* - \mathbf{1}\mu_2\right) \end{aligned}$$

Using the second squared sentence gives

$$\begin{aligned} &-\frac{1}{2}\log\left(\mid \mathbf{\Sigma}_1 \mid\right) - \frac{1}{2}\left(\boldsymbol{x}^{*t}\mathbf{\Sigma}_1^{-1}\boldsymbol{x}^* - 2\mathbf{1}\mu_1^t\mathbf{\Sigma}_1^t\boldsymbol{x}^* + \mathbf{1}\mu_1^t\mathbf{\Sigma}^{-1}\mathbf{1}\mu_1\right) \\ &> -\frac{1}{2}\log\left(\mid \mathbf{\Sigma}_2 \mid\right) - \frac{1}{2}\left(\boldsymbol{x}^{*t}\mathbf{\Sigma}_2^{-1}\boldsymbol{x}^* - 2\mathbf{1}\mu_2^t\mathbf{\Sigma}_2^t\boldsymbol{x}^* + \mathbf{1}\mu_2^t\mathbf{\Sigma}^{-2}\mathbf{1}\mu_2\right) \end{aligned}$$

and thus

$$\begin{aligned} &\log\left(\mid \mathbf{\Sigma}_1 \mid\right) + \left(\boldsymbol{x}^{*t}\mathbf{\Sigma}_1^{-1}\boldsymbol{x}^* - 2\mathbf{1}\mu_1^t\mathbf{\Sigma}_1^t\boldsymbol{x}^* + \mathbf{1}\mu_1^t\mathbf{\Sigma}^{-1}\mathbf{1}\mu_1\right) \\ &> \log\left(\mid \mathbf{\Sigma}_2 \mid\right) - \left(\boldsymbol{x}^{*t}\mathbf{\Sigma}_2^{-1}\boldsymbol{x}^* + 2\mathbf{1}\mu_2^t\mathbf{\Sigma}_2^t\boldsymbol{x}^* + \mathbf{1}\mu_2^t\mathbf{\Sigma}^{-2}\mathbf{1}\mu_2\right) \end{aligned}$$

If it is known in advance that there is a higher probability to belong to one class than to the other, the prior is also used to classify the class, namely

$$f_1\left(\mathbf{x_1}\right) \times \pi_1 > f_2\left(\mathbf{x_2}\right) \times \pi_2$$

where $\pi_1$ is the probability for belonging class 1 and $\pi_2$ is the probability for belonging to class 2. The sum of the priors are always equal to 1.

In the most cases the $\mu$s and the $\Sigma$s are unknown. Hence these values have to be estimated as $\overline{\mathbf{x}}_\mathbf{1}, \overline{\mathbf{x}}_\mathbf{2}, \mathbf{S_1}$ and $\mathbf{S_2}$, see Equation 1.1. A newly added observation, $\boldsymbol{x}_0$, belong to group 1 if

$$-\frac{1}{2}\mathbf{x_0^t}\left(\mathbf{S_1^{-1}} - \mathbf{S_2^{-1}}\right)\mathbf{x_0} + \left(\overline{\mathbf{x}}_\mathbf{1}^\mathbf{t}\mathbf{S_1^{-1}} - \overline{\mathbf{x}}_\mathbf{2}^\mathbf{t}\mathbf{S_2^{-1}}\right)\mathbf{x_0} - k \geq ln\left(\frac{\pi_2}{\pi_1}\right),$$

Where the $k$ is

$$k = \frac{1}{2} ln \left( \frac{|\boldsymbol{S}_1|}{|\boldsymbol{S}_2|} \right) + \frac{1}{2} \left( \overline{\boldsymbol{x}}_1^t \boldsymbol{S}_1^{-1} \overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2^t \boldsymbol{S}_2^{-1} \overline{\boldsymbol{x}}_2 \right)$$

else the observation belongs to group 2.

As mentioned in subsection 1.2.2 about LDA $n > p$ has to be fulfilled to be able to calculate the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ or $\boldsymbol{S}^{-1}$. This also applies to QDA. Moreover QDA uses PCA for reduce the number explanatory variables.

### 1.2.4 k-Nearest Neighbors (KNN)

The KNN classifier is a basic and simple classifier. When working on a data set with little or no knowledge on the data before starting the classification, it is wise to use the KNN classifier. The KNN classifier finds the K nearest observations and the new observations are allocated to the group which is most frequent among the K neighbors. $K = 3$ that the distances from the new observations to all the observations in the training set are calculated and then the three samples with the smallest distances are identified. The new observation belongs to the class to which most of the three identified observations belong to, see Figure 1.2.

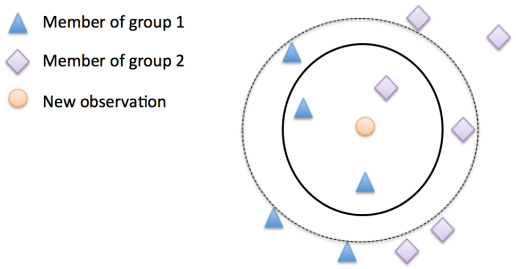



Figure 1.2: *The figure shows the borders for 3-NN (whole line) and 5-NN (dotted line)-classifiers. The blue triangles are observations belonging to group 1, the pink squares are the observations in group 2 and the orange circle is the new observation. The new observation is allocated to group 1 by both the 3-NN and the 5-NN classifiers.*

The euclidean distance is used most frequently to calculate the distance between the test observation and the training samples in KNN. The euclidean distance between a new observation $\mathbf{x_i}\,(i = 1, 2, \ldots, n)$ and an observation from the training samples $\mathbf{x_l}\,(l = 1, 2, \ldots, n)$ is calculated via

$$d\left(\mathbf{x_i}, \mathbf{x_l}\right) = \sqrt{\left(x_{i1} - x_{l1}\right)^2 + \left(x_{i2} - x_{l2}\right)^2 + \ldots + \left(x_{ip} - x_{lp}\right)^2}$$

where $p$ is the number of predictors in the model, $n$ is the total number of input samples and $\mathbf{x_l}$ is an observation of the already classified in the feature space.

## 1.3 Ordinary least squares (OLS)

As an introduction to other methods we will present the ordinary least squares (OLS) model.

**Linear model estimation:**

A linear model is defined as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon$$

where $\beta_0$ is the intercept. We assume the data to have centred variables which then give $\hat{\beta}_0 = 0$. Centred variables is when the mean of the variables are subtracted from variables, $(x_i - \overline{x})$, and from the response, $(y_i - \overline{y})$. The linear model in matrix form is given by

$$\underset{n\times 1}{\boldsymbol{y}} = \underset{n\times p}{\boldsymbol{X}}\ \underset{p\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{\epsilon}}$$

where $\epsilon \sim N(0, \sigma^2)$, $\boldsymbol{y}$ is a vector of responses, $\boldsymbol{\beta}$ is a vector of parameters and $\boldsymbol{X}$ is a matrix of $p$ explanatory variables and $n$ observations. When fitting a linear model the least square estimator is commonly used. It is given by

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X^t X}\right)^{-1} \boldsymbol{X^t y}$$

If $n > p$ the invertion of $X^t X$ will be possible and the estimation of $\hat{\beta}$ will be easy. Today's data often confront one with $n < p$, which makes it impossible to estimate $\hat{\beta}$ since the inversion $X^t X$ is not possibles.

## 1.4 Dimension reduction in regression

Methods based on dimenstion reduction reduce $\mathbf{X}$ to $\mathbf{Z}$ which has $a$ variables instead of $p$, $a < p$ and $a < n$, see Figure (1.3). The vectors $\mathbf{Z} = [\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_a}]$ in $\mathbf{Z}$ are orthogonalized. Hence $\mathbf{z_i^t z_j} = \mathbf{0}$ where $i \neq j$. All $\mathbf{z_i}$s are a linear combination of $\mathbf{x_{i,\ldots,p}}$

$$\mathbf{z_i} = r_1 \mathbf{x_1} + r_2 \mathbf{x_2} + \ldots + r_p \mathbf{x_p} \tag{1.4}$$

where $r_j (j = 1, 2, \ldots, p)$ are some numbers. This gives a new model for the regression of $\mathbf{y}$

$$\mathbf{y} = \mathbf{Z}\alpha + \mathbf{f}$$

where $\boldsymbol{f}$ is the error term. With OLS estimator:

$$\hat{\alpha} = \left(\mathbf{Z^t Z}\right)^{-1} \mathbf{Z^t y} \tag{1.5}$$

In the following Figure 1.3 an illustration of the dimension reduction process is shown.

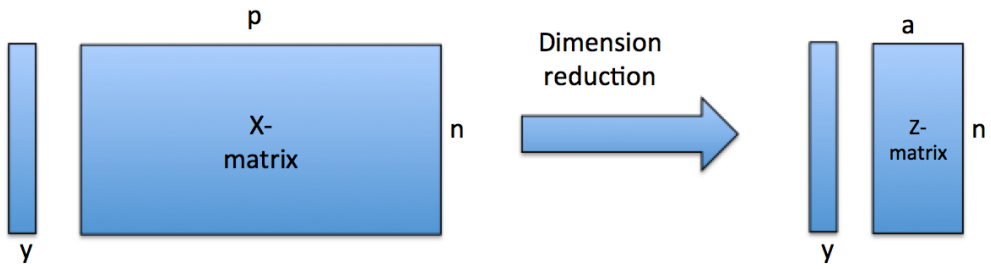



Figure 1.3: *The figure shows how dimension reduction reduces the X-matrix,* $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}]$ *to* $\mathbf{Z} = [\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_a}]$*, where the X-matrix contains p variables and the Z-matrix contains a variables, and* $a < p$*.*

For finding the $\boldsymbol{Z}$ variables there are many methods, two of them is PCR or PLSR. PCR looks for the $\pm Z$s which have the maximum variance. PLSR will find the $\boldsymbol{Z}$s with highest covariance to the $\boldsymbol{Y}$, the $\boldsymbol{Z}$s that are most related to the $\boldsymbol{Y}$.

### 1.4.1 Principal component regression (PCR)

PCR [Kendall 1957] compress the data by finding the direction with maximum variance, called $\mathbf{Z_1}$. Further $\mathbf{Z_2}$ is orthogonalized to $\mathbf{Z_1}$, still maximizing the variance in the x-space. The general way to find the $\mathbf{Z}$s is by

$$\mathbf{Z} = \mathbf{X}\mathbf{E_a}$$

where $\mathbf{E_a} = [\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_a}]$ it is found as the $a$ first eigenvectors of the covariance matrix of $\mathbf{X}$, $(cov(\mathbf{X}))$. It is known (see equation (1.5)) which gives

$$\hat{\beta}_{\mathbf{PCR}} = \mathbf{E_a}\hat{\alpha}$$

In this thesis the PCR is not used in the classification, but PCR is used for calculating the $\mathbf{Z}$ as input to LDA and QDA.

### 1.4.2 Partial Least Squares Regression (PLSR)

As mentioned before the PLSR [Martens and Næs 1989] will find the $\mathbf{Z}$s that are most related to $\mathbf{Y}$. The PLSR algorithm is consisting of five steps. Let $\mathbf{y_0}$ and $X_0$ be centred variables.

1. Compute the loading weights
   $\mathbf{w_1} = \mathbf{X_0^t y_0}$ this is replaced by normalized $\mathbf{w_1} \leftarrow \frac{\mathbf{w_1}}{\sqrt{\mathbf{w_1^t w_1}}}$.
   $\mathbf{w_i}$s in a PLSR model will correspond to the $\mathbf{r}$s in Eq 1.4.

2. Compute the $\mathbf{z_1}$ scoresvector:
   $\mathbf{z_1} = \mathbf{X_0 w_1}$
   by this the $|cov(\mathbf{y_0}, \mathbf{z_1})|$ is maximised.

For finding the $\mathbf{z_2}$ to $\mathbf{z_a}$ we need to remove the information found in $\mathbf{z_1}$.

3. Find the loadings for $y$ and $x$
   The x loadings: $\mathbf{p_1} = \mathbf{X_0^t z_1} \left(\mathbf{z_1^t z_1}\right)^{-1}$
   The y loadings: $\mathbf{q_1} = \mathbf{y_0^t z_1} \left(\mathbf{z_1^t z_1}\right)^{-1}$

4. The inflation step. Find the residual matrix for both $\boldsymbol{X}$ and $\boldsymbol{y}$:
   The residual matrix for $\boldsymbol{X}$: $\mathbf{X_1} = \mathbf{X_0} - \mathbf{z_1 p^t}$
   The residual matrix for $\boldsymbol{y}$: $\mathbf{y_1} = \mathbf{y_0} - \mathbf{z_1 q^t}$

5. Return to step 1 for finding the next $\mathbf{w_i}$ and $\mathbf{z_i}$. Repeat this algorithm until all $w_i$ and $i(i = 1, \ldots, a)$ are found.

The PLSR algorithm will be repeated $a$ times. The number of components $a$ are decided by

- crossvalidation or testing data prediction
- trying different values of $a$, predict $y$ and then choosing the a value which gives the lowest prediction error.

The $\hat{\boldsymbol{\beta}}$ is estimated via

$$\hat{\boldsymbol{\beta}}_{PLSR} = \mathbf{W} \left(\mathbf{P^t W}\right)^{-1} \mathbf{Q} \tag{1.6}$$

where $\underset{p\times a}{\mathbf{W}} = [\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_a}]$, $\underset{p\times a}{\mathbf{P}} = [\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_a}]$ and $\underset{a\times 1}{\mathbf{Q}} = [\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_a}]^{\mathbf{t}}$.
One can now find the $\hat{\boldsymbol{\beta}}_0$ for the original variables by
$\hat{\boldsymbol{\beta}}_0 = \overline{\boldsymbol{y}} - \hat{\boldsymbol{\beta}}^t_{PLSR} \overline{\boldsymbol{x}}$

Moreover the prediction model for a new $\boldsymbol{x}^*$ is

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\beta}}_0 + \hat{\boldsymbol{\beta}}^t_{PLSR}\boldsymbol{x}^*$$

## 1.5 Validation

**Evaluating a classification rule**

When evaluating a classification rule it is common to calculate the classification error also known as Apparent Error Rate (APER) which gives the percentage of misclassification. The closer the APER is to zero, the better the classification model is.

The APER is defined as:

$$APER = \frac{n_{1M} + n_{2M}}{n_1 + n_2}$$

where

$n_{1M}$: the number of correctly classified observations in group 1

$n_{2M}$: the number of correctly classified observations in group 2

$n_1$: the number of observations in group 1

$n_2$: the number of observations in group 2

### 1.5.1 Test data

Evaluating these classifiers requires both training data and test data because when creating a model for a dataset there is always the risk of overfitting i.e. the model is fitted to the certain dataset but does not work well when used

for another dataset. Hence the dataset is divided into training data and test data. The model is fitted to the training data and then tested on the test data. The purpose of using training data is to fit the model to these data while knowing the answer to which group the different observation belongs to. The test data are used to test the model by predicting which group the different observations in the test data belong to.

### 1.5.2 Cross validation (CV)

The risk of overfitting can be reduced by using cross validation. This is done by dividing the data into segments of which one is left out of the fitting process. Later on the left-out-segment is used as test data for the fitted model. After that another segment is left out as test data. This rutine is repeated until all segments have been left out as test data. For example, if there are ten segment, the model will be fitted ten times with different data, each time leaving out another observation as test observation.

#### General case: K-fold cross validation

The K-fold cross validation is a CV which divides the observation into K groups, also called folds. Of these groups one group is left out and the rest is used as training data set. The group that is left out is the test data set for which the classification error is estimated. After doing so for the first fold, one continues to the next fold. This process is repeated until each group served as test data. After finishing all the fittings the K-Fold CV error is calculated.

This error is the average of all APERs $(APER_1, APER_2, \ldots, APER_K)$

$$CV_K - error = \frac{1}{K}\sum_{i=1}^{K} APER_i.$$

The number K of groups affects the results. Choosing a large K gives a large training data set that proviedes a good estimation of the model but a poor estimation of the classification error due to a small test data set. On the other hand if K is chosen small it gives a poor parameter estimation but the estimation of the classification error is quite good due to the large test data set. In order to get the best possible estimation one should choose a medium sized K. Thus K=10 is a frequent choice in literature.

**Special case: leave-one-out cross validation, $K = n$**

A special case of K-fold CV is $K = n$, which is also known as leave-one-out cross validation (LOOCV). LOOCV divides the data set into parts, but unlike other cross validation methods will LOOCV leave one of the observations out and the rest of the observations will constitute the training data.For the first run $(x_1, y_1)$ is left out and the remaining $[(x_2, y_2), \ldots, (x_n, y_n)]$ are the training data set. The model is then fitted to the n-1 training observations. $\hat{y}_1$ is predicted separately from the others since it uses $x_1$, then will also the $APER_1$ calculated. $APER_1$ it will be unbiased, the prediction error $APER_1$ will have high variance because it is predicted on only one observation. After the $APER_1$ is predicted $APER_2$ is predicted on $(x_2, y_2)$ with the training data set $[(x_1, y_1), (x_3, y_3), \ldots, (x_n, y_n)]$. This is repeated until all observations have served as test data. LOOCV will then find find the average of all

the test APER's which will be the estimate for test error

$$CV_n - error = \frac{1}{n}\sum_{i=1}^{n} APER_i.$$

One of the main advantages of LOOCV is that it has less bias than other CVs. Moreover there is less risk of overfitting[James et al. 2013b]. Another advantage is that LOOCV gives less randomness in results compared to other CVs. That is because LOOCV just leaves out one observation instead of a set of observations. Because of this LOOCV will have has similar results in every run. Other CV methods have some randomness in their results due to the fact that they divide the training set and the test set into larger groups.

## 1.6 Analysis of variance

### 1.6.1 ANOVA

The analysis of variance (ANOVA) (see e.g. Montgomery 2009a) is a statistical method used to make comparisons between two or more groups with regard to their effect on some response variable. By doing this it is possible to test and determine which variables are significant and if there exists a significant relation between the groups. In a one-way ANOVA the observations will be divided into different groups also called treatments. A one-way effects model can be formulated as

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij}$$

where $y_{ij}$ is the response, $\mu$ is the overall mean, the $\tau_i$ is the effect of the treatment with level i, $\varepsilon_{ij}$ is the error term. The indices are $i = 1, 2, \ldots, a$

and $j = 1, 2, \ldots, n_i$. An example is that the response is the weight of pigs on a different diets (treatments). In this thesis ANOVA is used to test the effects of classifiers and simulated parameters on classifications error.

Usually an ANOVA table is made to list the sources of variation in the data. SST is a measure of overall variability in the data with the formula

$$SS_T = \Sigma_{i=1}^{a} \Sigma_{j=1}^{n_i} \left(y_{ij} - \overline{y}_{..}\right)^2$$

with degrees of freedom, $DF = N - 1$. SSTreatment is a measure of the variability between the treatments, with the formula

$$SS_{Tr} = \Sigma_{i=1}^{a} n_i \left(\overline{y}_{i.} - \overline{y}_{..}\right)^2$$

with DF = a - 1. SSE is the variability within the treatments, with the formula

$$SS_E = \Sigma_{i=1}^{a} \Sigma_{j=1}^{n} \left(y_{ij} - \overline{y}_{i}.\right)^2$$

with DF = N - a.

Often there are more than one factor in the model. In such models it is common to also consider the interactions between the factors. Interaction is when the effect of a factor on the response is depending on the level of another factor. .

A model with two treatments and interaction is: $y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk}$ where number of groups for $\tau$ is $i = 1, 2, ..., a$, number of groups for $\beta$ is $j = 1, 2, ..., b$ and number of observations in the groups $k = 1, 2, ..., n$.

### 1.6.2 Experimental design

#### Two-level factorial design

In order to study the joint effect of the factors on the response the factorial design is widely used. Two-level factorial design (see e.g. Montgomery 2009b) means that the factors have only two levels each, "high" or "low". Often are the levels denoted as "+" and "−" for high and low levels. These designs are often called $2^k$-designs where k is the number of factors and the $2^k$ will be the total number of runs in the design, across all k factors.

For a $2^2$ factorial design there will be two different factors and four different treatments combinations. Lets say there are two factors, A and B. Then their combinations will be like in Table 2.1. This is also known as a design matrix

| Total | A | B |
|---|---|---|
| (1) | − | − |
| a | + | − |
| b | − | + |
| ab | + | + |

Table 1.1: *Overview of treatment combinations in a $2^2$ factorial design. The "+" means the high level of a factor, and the "−" means the low level of a factor.*

Calculations of the main effect in a $2^2$ factorial design is done by finding the difference between the average when the factor is high level and when the factor is low level (see Table 1.1), this will only be optimal when it is balanced design where $n_i = n$, and the formulas will be

The main effect of factor A

$$A = \overline{Y}_{A^+} - \overline{Y}_{A^-} = \frac{1}{2n}\left([ab - b] + [a - (1)]\right)$$

The main effect of factor B

$$B = \overline{Y}_{B^+} - \overline{Y}_{B^-} = \frac{1}{2n}\left([ab - a] + [b - (1)]\right)$$

The main effect of interaction AB

$$AB = \frac{1}{2n}\left([ab - b] - [a - (1)]\right)$$

The general $2^k$ factorial design is a design that has $k$ factors with two levels each. For preforming a statistical analysis for a $2^k$ factorial design one should start by estimating the factor effects and then state the model. In a full statistical model for $2^k$ factorial design there will be $k$ main effects, $\binom{k}{2}$ two-factor interactions, $\binom{k}{3}$ three-factor interactions, ..., and one $k$-factor interaction. If $k$ is large this can lead to complicated model and interaction levels that are not significant. To make the model simpler one can reduce the model by performing a set of statistical tests to the model. One way to do this is to use backward selection method. The user sets a significance level ($\alpha$), which often is set to $\alpha = 0.05$. Then the method will look through the model for the highest p-value, if the p-value is over $\alpha = 0.05$ the corresponding term will be removed from the model and the model is refitted without the term. This is done until every effect is significant. There are some rules that has to be followed, the hierarchy in the model must be maintained. This means that if an effect on a lower level is not significant, but is part of a

higher order interaction that is significant, then the effect on the lower level must be retained even though it is not significant. This also means that main effects can not be rejected if it is a part of a significant order higher interaction effect.

# Chapter 2

# Methods

## 2.1 Methods

### 2.1.1 Hot PLS

The Hot PLS [Liland et al. 2014] will classify objects by following a known hierarchical structure for the classes and using PLS at each hierarchical split. The Hot PLS structure is similar to a classification tree à la [Breiman et al. 1984], but the classification in each branch is replaced by PLS discriminant analyse (PLS- DA). Figure 2.1 shows how the data with a hierarchical structure is organized.

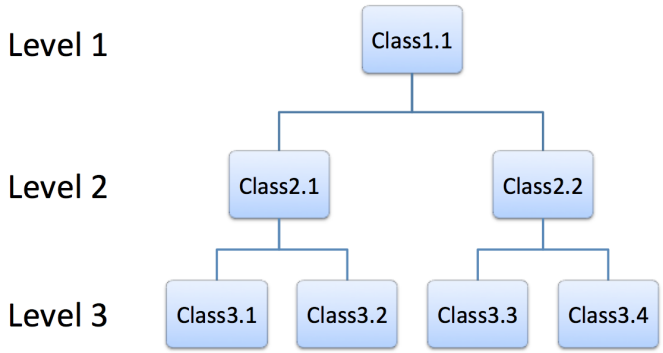



Figure 2.1: *The figure show a hierarchy system with branches and nodes, where there are three levels. The horizontal lines are the different nodes and the the vertical lines are the branches.*

In the article of Liland et al. 2014 they explain that the Hot PLS will calculate a PLS model based on the observations on the exactly branch for each node. In the model for the nodes there can be one or more groups per node. Figure 2.1 explains what a branch and a node are. The number components will be determined locally by cross-validation. In the article they use CPPLS [Indahl et al. 2009] for making the PLS model and the classifier is LDA which uses the PLS scores as predictors, but in this thesis we will be using the normal PLS with LDA. A problem with using methods like PLS and LDA is that an observation always needs to be assigned to a group. This can lead to that new groups will not be discovered. In Liland et al. 2014 they deal with this problem by adding an estimation of how similar an observation has to be to an existing group for belonging to that group.

Observations which do not seem to belong to any existing group will get labeled "low confidence", because it could simply be an outlier of a another group or belong to a new group. The algorithm for the training data are as follows:
Constructing the tree

1. Make a hierarchical tree for the classes based on background knowledge.
2. Remove any obviously non-informative levels in the tree.
   Training the nodes recursively:
3. Estimate the number PLS components for the node.
4. Calculate the PLS model.
5. Repeat from 3. for the next node(s).

And the algorithm for classfication:

1. Calculate the prediction scores.
2. Classify by LDA.
3. Identify the "low confidence" observations.
4. Repeat from 1. for the next node(s).

## 2.2 Data

The data in this thesis are simulated by the R-package Simrel [Sæbø 2015]. The Simrel package provides data with specific properties which are decided

by the users. The properties that are specified are:

- $n$ : the number of observations
- $p$ : the number of predictors
- $q$ : the number of relevant predictors
- $m$ : the number of relevant components
- Relpos : the set of relevant components
- $R^2$ : the population coefficient of determination
- $\gamma$ : a parameter defining the of collinearity in $x$

The Simrel package is based on the general random regression model

$$\boldsymbol{y} = \mu_y + \boldsymbol{\beta}^t (\boldsymbol{x} - \boldsymbol{\mu}_x) + \epsilon \tag{2.1}$$

where the $\boldsymbol{y}$ is the response variable, $\mu_y$ is the expected value for the respsonse, $\boldsymbol{\beta}$ is a vector of regression coefficients, $\boldsymbol{x}$ is the vector with $p$ predictor variables and is assumed to be random and $\epsilon$ is the error term. The error term is assumed to be normally distributed as $N(0, \sigma^2)$. Also the general linear model can be written as

$$\begin{bmatrix} y \\ \boldsymbol{x} \end{bmatrix} \sim N \left( \begin{bmatrix} \mu_y \\ \boldsymbol{\mu_x} \end{bmatrix}, \begin{bmatrix} \sigma_y^2 & \boldsymbol{\sigma}_{xy}^t \\ \boldsymbol{\sigma}_{xy} & \boldsymbol{\Sigma}_{xx} \end{bmatrix} \right)$$

where the $\boldsymbol{\sigma}_{xy}$ is the vector of covariances between the response and the predictor variables, $\boldsymbol{\Sigma}_{xx}$ is the covariance matrix of $\boldsymbol{x}$.

It is known that any set of variables spanning the same p-dimensional predictor space as $\boldsymbol{x}$ gives the same prediction of $\boldsymbol{y}$ and keep the same noise variance and coefficient of determination. To be able to simulate the $\boldsymbol{x}$ and the $\boldsymbol{y}$ from the model in equation 2.1 this knowledge is used. Therefore we let $\boldsymbol{R}$ be a $(p \times p)$ matrix with rank $p$, then use this matrix to define the random variable vector $\boldsymbol{z} = \boldsymbol{R}\boldsymbol{x}$. Then we have:

$$\begin{bmatrix} y \\ \mathbf{z} \end{bmatrix} \sim N\left( \begin{bmatrix} \mu_y \\ \boldsymbol{\mu}_z \end{bmatrix}, \begin{bmatrix} \sigma_y^2 & \boldsymbol{\sigma}_{zy}^t \\ \boldsymbol{\sigma}_{zy} & \boldsymbol{\Sigma}_{zz} \end{bmatrix} \right) = N\left( \begin{bmatrix} \mu_y \\ \boldsymbol{R}\boldsymbol{\mu}_x \end{bmatrix}, \begin{bmatrix} \sigma_y^2 & \boldsymbol{\sigma}_{xy}^t \boldsymbol{R}^t \\ \boldsymbol{R}\boldsymbol{\sigma}_{xy} & \boldsymbol{R}\boldsymbol{\Sigma}_{xx}\boldsymbol{R}^t \end{bmatrix} \right)$$

The $\boldsymbol{R}$ matrix is also chosen to be an orthonormal matrix such that $\boldsymbol{R}^t\boldsymbol{R} = \boldsymbol{I}_p$ then we will also have that $\boldsymbol{\Sigma}_{xx} = \boldsymbol{R}^t\boldsymbol{\Sigma}_{zz}\boldsymbol{R}$ and $\boldsymbol{\sigma}_{xy} = \boldsymbol{R}^t\boldsymbol{\sigma}_{zy}$ and the linear model:

$$\boldsymbol{y} = \mu_y + \boldsymbol{\alpha}^t (\boldsymbol{z} - \boldsymbol{\mu}_z) + \boldsymbol{\epsilon} \tag{2.2}$$

with $\boldsymbol{\epsilon} \sim N(0, \boldsymbol{\tau}^2)$. Further, the simrel will choose $\sigma_y^2 = 1$ and

$$\boldsymbol{\Sigma}_{zz} = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_p \end{bmatrix}$$

where the $\lambda$s are the eigenvalues. In relsim the $\lambda_1 = 1$ and they are in descending order, so $\lambda_1 > \lambda_2 > \ldots > \lambda_p$. The value of the other $\lambda$s are decided by the equation

$$\lambda_j = \frac{e^{-\gamma j}}{e^{-\gamma}} \quad j = 1, \ldots, p$$

where the $\gamma > 0$, $\gamma$ is the factor in Simrel that will decide how fast the eigenvalues will decrease. When $\gamma$ is large the eigenvalues will decrease fast,

and if the $\gamma$ is small the eigenvalues will decrease slow. The $\boldsymbol{\sigma}_{yz}$ vector is chosen by the user who decide the number of the $\boldsymbol{z}$s which are relevant, $m$. The user also decides which of the $\boldsymbol{z}$s which are relevant. The set of relevant components is called relpos. Let say that $m = 1$ and relpos= [1] then the $\boldsymbol{\sigma}_{yz} = \begin{bmatrix} \alpha_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ or if the $m = 3$ and the relpos= [1 3 4] then the $\boldsymbol{\sigma}_{yz} = \begin{bmatrix} \alpha_1 \\ 0 \\ \alpha_3 \\ \alpha_4 \\ \vdots \\ 0 \end{bmatrix}$

In the article about Simrel [Sæbø et al. 2015] they show that there is a direct connection between the regression coefficients of $\boldsymbol{z}$ and $\boldsymbol{x}$ by $\boldsymbol{\alpha} = \boldsymbol{R\beta}$ where the $\boldsymbol{\alpha}$ is from equation (2.2) and the noise variance are similar $\boldsymbol{\tau}^2 = \boldsymbol{\sigma}^2$. They also show that the population coefficient of determination for $\boldsymbol{R}_z^2$ and $\boldsymbol{R}_x^2$ are similar: $\boldsymbol{R}_z^2 = \boldsymbol{R}_x^2$. The next step in the simulation is to simulate $\boldsymbol{y}$ and $\boldsymbol{z}$, for this part the we generate $\boldsymbol{w} \sim N(0, \boldsymbol{I})$. Further let $\boldsymbol{\Sigma}_{yz}^{\frac{1}{2}}$ be so that $\left(\boldsymbol{\Sigma}_{yz}^{\frac{1}{2}}\right)^t \cdot \boldsymbol{\Sigma}_{yz}^{\frac{1}{2}} = \boldsymbol{\Sigma}_{yz}$. This is done by Simrel using the Cholesky decompostion. Then, the procedure is

- draw a random number from $w_1$
- calculate $\left(\boldsymbol{\Sigma}_{\boldsymbol{yz}}^{\boldsymbol{\frac{1}{2}}}\right)^t \cdot \boldsymbol{w} = \boldsymbol{v}$, so that the $cov(\boldsymbol{v}) = \left(\boldsymbol{\Sigma}_{yz}^{\frac{1}{2}}\right)^t \cdot \boldsymbol{I} \cdot \boldsymbol{\Sigma}_{\boldsymbol{yz}}^{\boldsymbol{\frac{1}{2}}} = \boldsymbol{\Sigma}_{\boldsymbol{yz}}$
- let $\boldsymbol{v_1} = \begin{bmatrix} y_1 \\ \boldsymbol{z_1} \end{bmatrix}$
- repeat these operations n times to get n observations.

Now the simrel package will use a random orthonormal rotation matrix $\underset{p\times p}{\boldsymbol{R}}$ which by rotation of $\boldsymbol{z}$ yields predictors:

$$\underset{p\times 1}{X} = \boldsymbol{R}^t z$$

Further, the covariances between $\boldsymbol{y}$ and $\boldsymbol{x}$ is

$$\boldsymbol{\sigma}_{yx} = \boldsymbol{R}^t \boldsymbol{\sigma}_{yz}$$

and the covariance matrix of

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

Now Simrel has given us everything the model in equation (2.1) needs, where $\mu_x = \mu_y = 0$ and $\boldsymbol{\beta} = \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{\sigma}_{yx}$, this defines the model:

$$\boldsymbol{y} = \boldsymbol{\beta^t x} + \boldsymbol{\epsilon}$$

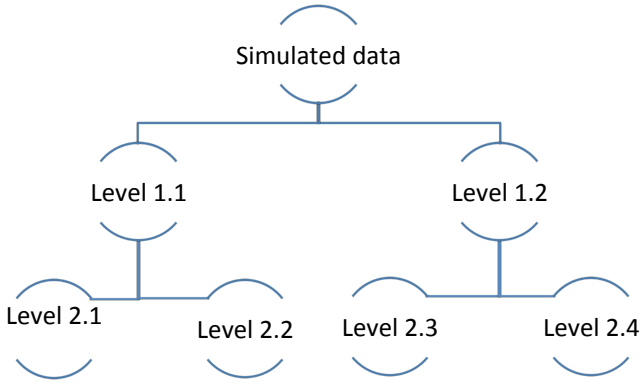



*Figure 2.2: An overview the levels in the simulated data. Simrel provides a y that is continuous. For making the y categorical it is dichotomized by $y < 0$ and $y > 0$. For the y to belong to level 1.1 it has to be $y < 0$ and for belong to level 1.2 it has to be $y > 0$. For level 2 the y was dichotomized in the same way, but being done in two stages with half n that was in level 1. So in the first stage will the $y < 0$ will belong in level 2.1 and $y > 0$ will belong in level 2.2. In the second stage will the $y < 0$ will belong in level 2.3 and $y > 0$ will belong in level 2.4.*

The simulated data are organized hierarchically with two levels, the first level has two groups and the second level has four groups, see Figure 2.2. Simrel will generate Xs for both first level and the second level in the hierarchy, these Xs is then added together, so there is only one X for the whole hierarchy.

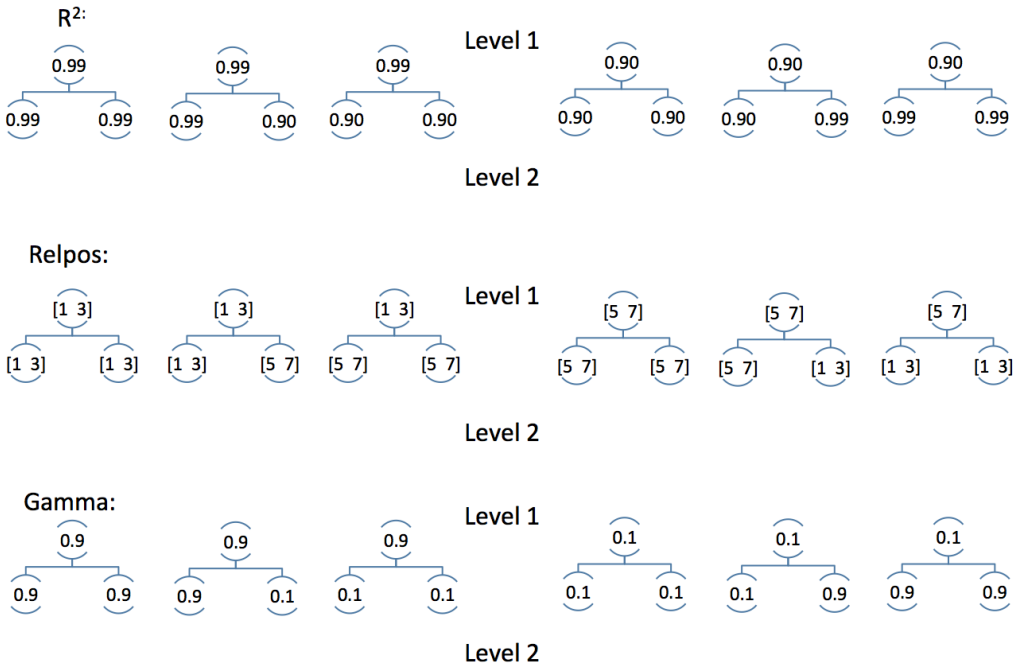



Figure 2.3: *Figure over the levels on the different factors in the simulated data*

Simrel was used to simulate the hierarchical data at each level of the hierarchy, Figure 2.3. The hierarchical data were simulated by setting five factors in Simrel: $R^2$, $\gamma$, relpos, number of observations $n$ and number of variables $p$. The factors $R^2$, $\gamma$ and relpos have two levels each, high and low. When this is combined with the hierarchical system for the simulated data we get six different combinations for each factor, see figure 2.3. That constitutes totally 216 different combinations of the three factors. The number of observations, $n$, and number of variables, $p$, do also have the high and low levels, but the number of observations and the number for variables will not change in the hierarchically system therefore the total number of different

combinations will be $216 \times 2 \times 2 = 864$.

A for-loop goes through the design matrix and run every design one by one. Each design will go through the the simulation process and the classification process (HOT PLS, PLS, LDA, QDA, 3NN) three times. The average APER from the HOT PLS and the LDA will be stored in a matrix and this will be used in a ANOVA.

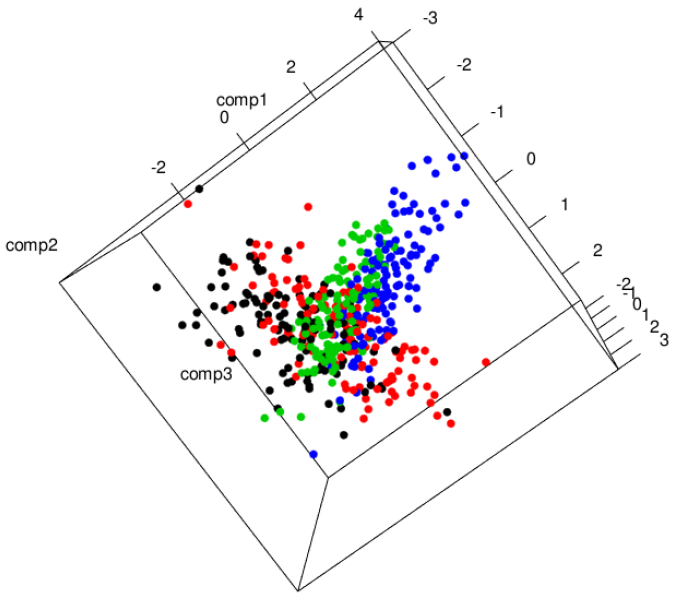



Figure 2.4: *Figure shows a 3D with an example when the simulation parameter is set to be easy to classify. The data that has been plotted has these setting;* $R^2 = 0.99$, *relpos = [1 2 3] and* $\gamma = 0.9$. *The color coding give the four different groups.*

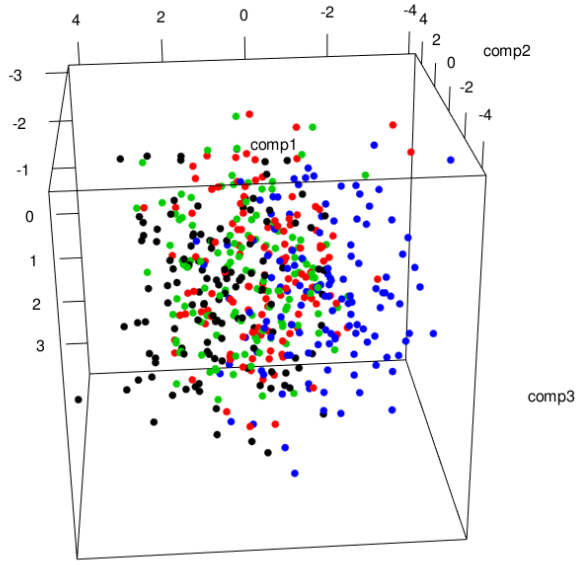



Figure 2.5: *Figure shows a 3D with an example when the simulation parameter is set to be hard to classify. The data that has been plotted has these setting;* $R^2 = 0.80$, *relpos = [1 2 3] and* $\gamma = 0.1$. *The color coding give the four different groups.*

Figure 2.4 one can see that the plot is easier to separated the groups from each other than the plot in Figure 2.5 where the observations are on top of each other.

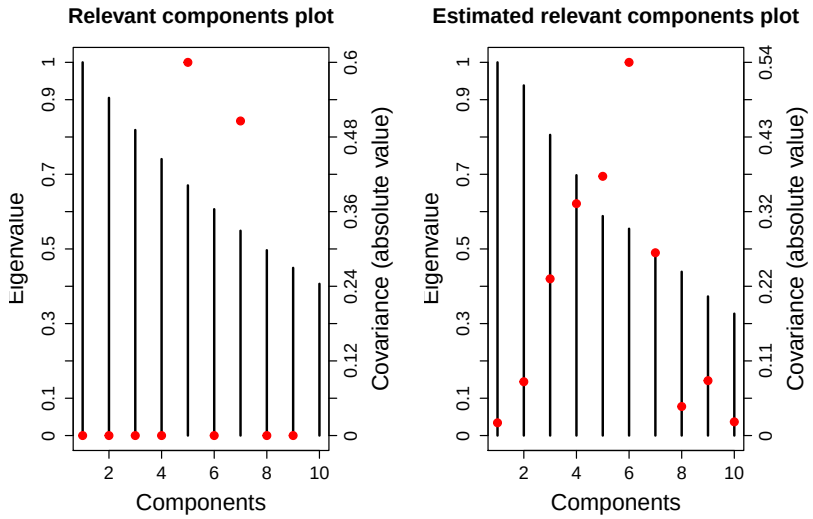



Figure 2.6: *The figure give the eigenvalues for components and the correlations between the components and the y. The relevant components is set to component 5 and 7, and* $\gamma = 0.1$. *The relevant components plot (on the left) shows how the relationship between the eigenvalues and correlations are in simrel and estimated relevant components plot (on the right) shows how the relationship between the eigenvalues and correlations are for real data. The correlations is given in absolute value.*

Figure 2.6 shows how the eigenvalues slowly drops with $\gamma = 0.1$ and most of the information are in component 5 and 7. It also shows how hard it is to find the information when Relpos has components with small eigenvalues.

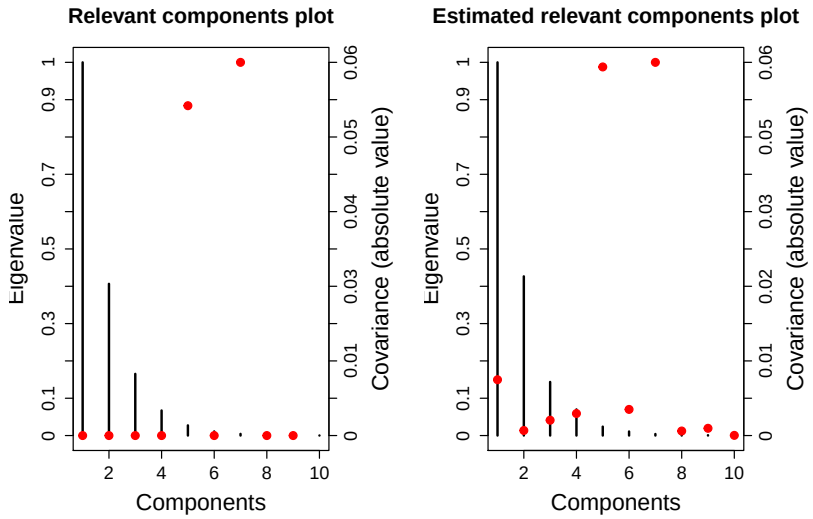



Figure 2.7: *The figure give the eigenvalues for components and the correlations between the components and the y. The relevant components is set to component 5 and 7, and* $\gamma = 0.9$. *The relevant components plot (on the left) shows how the relationship between the eigenvalues and correlations are in simrel and estimated relevant components plot (on the right) shows how the relationship between the eigenvalues and correlations are for real data. The correlations is given in absolute value.*

Figure 2.7 shows that it is even harder it is to find the information when Relpos has components with small eigenvalues and $\gamma$ is set so the eigenvalues drops fast.

### 2.2.1 Analyzing the data

For analyzing the classification error the data was fitted by an ANOVA model which included up to the three-factor interactions and the the classification

error was the response. For finding the significant factors and interactions in the model it was conducted a backward/forward selection where the signification level, $\alpha = 0.05$, was selected. Effect plot was made so that the signification factors that it should be easier to interpret the results. The $\gamma$, $R^2$, and relpos has six different levels, however $n$, $p$, and methods have only two levels, see Table 2.2.1.

| Factor: | $\gamma$ | $R^2$ | Relpos | $n$ | $p$ | Methods |
|---|---|---|---|---|---|---|
| Level 1: | 0.1/0.1/0.1 | 0.9/0.9/0.9 | [1 3]/[1 3]/[1 3] | 100 | 10 | Hot PLS |
| Level 2: | 0.1/0.1/0.9 | 0.9/0.9/0.99 | [1 3]/[1 3]/[5 7] | 500 | 200 | PLS |
| Level 3: | 0.1/0.9/0.9 | 0.9/0.99/0.99 | [1 3]/[5 7]/[5 7] | | | |
| Level 4: | 0.9/0.1/0.1 | 0.99/0.9/0.9 | [5 7]/[1 3]/[1 3] | | | |
| Level 5: | 0.9/0.1/0.9 | 0.99/0.9/0.99 | [5 7]/[1 3]/[5 7] | | | |
| Level 6: | 0.9/0.9/0.9 | 0.99/0.99/0.99 | [5 7]/[5 7]/[5 7] | | | |

Table 2.1: *Overview of the two values/settings for each factor in model*

The ANOVA-model is as follows:

$$\begin{aligned} y_{ijklmn} = \mu + \alpha_i + \beta_j + \xi_k + \theta_l + \phi_m + \omega_n + (\alpha\beta)_{ij} + (\alpha\xi)_{ik} + (\alpha\theta)_{il} + \\ (\alpha\phi)_{im} + (\alpha\omega)_{in} + (\beta\xi)_{jk} + (\beta\theta)_{jl} + (\beta\phi)_{jm} + (\beta\omega)_{jn} + (\xi\theta)_{kl} + (\xi\phi)_{km} + \\ (\xi\omega)_{kn} + (\phi\omega)_{mn} + (\alpha\beta\xi)_{ijk} + (\alpha\beta\theta)_{ijl} + (\alpha\beta\omega)_{ijn} + (\alpha\xi\theta)_{ikl} + (\alpha\xi\phi)_{ikm} + \\ (\alpha\theta\phi)_{ilm} + (\alpha\xi\omega)_{ikn} + (\alpha\phi\omega)_{imn} + (\xi\theta\phi)_{klm} + (\beta\phi\omega)_{jmn} + (\xi\phi\omega)_{ikl} + \epsilon_{ijklmn} \end{aligned} \tag{2.3}$$

where $\epsilon_{ijklmn} \sim N(0, \sigma^2)$. And the $y_{ijklmn}$ is the average classification error of three replicates with the level; $i$ of $\gamma$ with effect $\alpha_i\,(i = 1, \ldots, 6)$, $j$ of $R^2$ with effect $\beta_j\,(j = 1, \ldots, 6)$, $k$ of relpos with effect $\xi_k\,(k = 1 \ldots 6)$, $l$ of $n$

with effect $\theta_l\,(l = 1, 2)$, $m$ of $p$ with effect $\phi_m, (m = 1, 2)$, and $n$ of methods with effect $\omega_n, (n = 1, 2)$. This is the ANOVA-model used to compare Hot PLS and PLS. Later the model will be extended for also compare with other classifiers, LDA, QDA, and KNN.

# Chapter 3

# Results

## 3.1 Comparison of Hot PLS and regular PLS

### 3.1.1 Main effects of the design parameters

First part of the results will focus on the model from equation 2.3, where we are only comparing Hot PLS and PLS. And then will the focus be on comparing the five different classifiers, Hot PLS, PLS, LDA, QDA, and LDA.

Analysis of Variance Table

Response: Err

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) | |
|---|---|---|---|---|---|---|
| gamma | 5 | 0.9043 | 0.1809 | 110.7974 | < 2.2e-16 | *** |
| Method | 1 | 0.3036 | 0.3036 | 186.0053 | < 2.2e-16 | * |
| n | 1 | 3.9740 | 3.9740 | 2434.4478 | < 2.2e-16 | *** |
| p | 1 | 15.7690 | 15.7690 | 9659.9211 | < 2.2e-16 | *** |
| pos | 5 | 5.8832 | 1.1766 | 720.8038 | < 2.2e-16 | *** |
| R2 | 5 | 0.3931 | 0.0786 | 48.1592 | < 2.2e-16 | *** |
| gamma:Method | 5 | 0.7015 | 0.1403 | 85.9416 | < 2.2e-16 | *** |
| gamma:n | 5 | 0.0211 | 0.0042 | 2.5867 | 0.0244810 | * |
| gamma:p | 5 | 1.0188 | 0.2038 | 124.8156 | < 2.2e-16 | *** |
| gamma:pos | 25 | 3.1611 | 0.1264 | 77.4588 | < 2.2e-16 | *** |
| gamma:R2 | 25 | 0.0797 | 0.0032 | 1.9527 | 0.0033868 | ** |
| Method:p | 1 | 0.0013 | 0.0013 | 0.8212 | 0.3649920 | |
| Method:pos | 5 | 0.5559 | 0.1112 | 68.1102 | < 2.2e-16 | *** |
| Method:R2 | 5 | 0.0866 | 0.0173 | 10.6051 | 5.291e-10 | *** |
| n:p | 1 | 0.8041 | 0.8041 | 492.5897 | < 2.2e-16 | *** |
| n:pos | 5 | 0.0534 | 0.0107 | 6.5370 | 5.131e-06 | *** |
| n:R2 | 5 | 0.0117 | 0.0023 | 1.4289 | 0.2109251 | |
| p:pos | 5 | 0.0863 | 0.0173 | 10.5767 | 5.644e-10 | *** |
| p:R2 | 5 | 0.0394 | 0.0079 | 4.8251 | 0.0002236 | *** |
| pos:R2 | 25 | 0.1093 | 0.0044 | 2.6794 | 1.535e-05 | *** |

```
gamma:Method:p      5  0.2423  0.0485   29.6809 < 2.2e-16 ***
gamma:Method:pos   25  0.3104  0.0124    7.6056 < 2.2e-16 ***
gamma:Method:R2    25  0.0639  0.0026    1.5649 0.0378015 *
gamma:n:p           5  0.0434  0.0087    5.3196 7.591e-05 ***
gamma:n:pos        25  0.0749  0.0030    1.8359 0.0073175 **
gamma:n:R2         25  0.0809  0.0032    1.9834 0.0027507 **
gamma:p:pos        25  0.5260  0.0210   12.8894 < 2.2e-16 ***
gamma:pos:R2      125  0.2785  0.0022    1.3651 0.0064028 **
Method:p:pos        5  0.0326  0.0065    3.9890 0.0013523 **
Method:p:R2         5  0.0211  0.0042    2.5846 0.0245818 *
n:p:pos             5  0.0735  0.0147    9.0031 2.003e-08 ***
Residuals        1312  2.1417  0.0016
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 3.1: *Anova table of the significant factors up to third interaction after model simplifications by backwards/forward elimination of non-significant effects. In this table the methods were only Hot PLS and PLS.*

Table 3.1 shows the Anova table with the significant factors of the model from equation 2.3. This table only show the significant factors up to third interaction. The backward/forward elimination will eliminated the non-significant factors by testing the p-values with $\alpha$ for the different treatments. The table is used to find which interaction plots that is interesting to examine the effect of the factors on APER values.

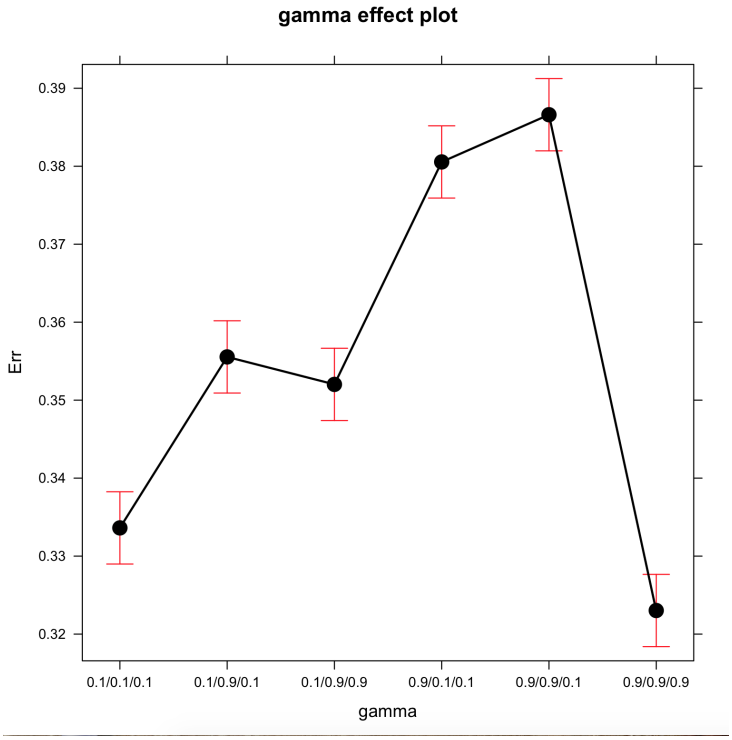



Figure 3.1: *The figure shows the main effect of $\gamma$, the x-axis the different $\gamma$ values, and the y-axis give the APER for the average of Hot PLS and PLS. The $\gamma$ values are given in six sets with three values each,* 0.1/0.1/0.1*, the first value is the $\gamma$-value for level 1 in the hierarchy, the next value is the $\gamma$-value for level 2.1 in the hierarchy and the last $\gamma$-value is for level 2.2 in the hierarchy.*

The Figure 3.1 shows the main effect of $\gamma$ on the APER. $\gamma$ has six different setting, where each setting has three values, these values are either low $\gamma$ or high $\gamma$. One can see that it is easy to classify when the $\gamma = 0.9$ on level 2.1 and level 2.2 and when $\gamma = 0.9$ on level 1. $\gamma = 0.9$ means that the eigenvalues are decreasing fast, and $\gamma = 0.1$ means that the eigenvalues are decreasing slowly. In general it is easier to classify when $\gamma = 0.1$ on level 1. And when

$\gamma = 0.9$ is on both level 2.1 and level 2.2. Easy to classify means that the APER is low, few classification errors. Eigenvalues only show that there are variation in the x-space, $\lambda_j = var(\mathbf{z_j})$, where $\lambda_j$ is a eigenvalue and $var(\mathbf{z_j})$ is the variation of scoresvector, $\mathbf{z_j}$. If the $cov(y, z_j) \neq 0$ it means that the the $\mathbf{z_j}$ contains information which is relevant.

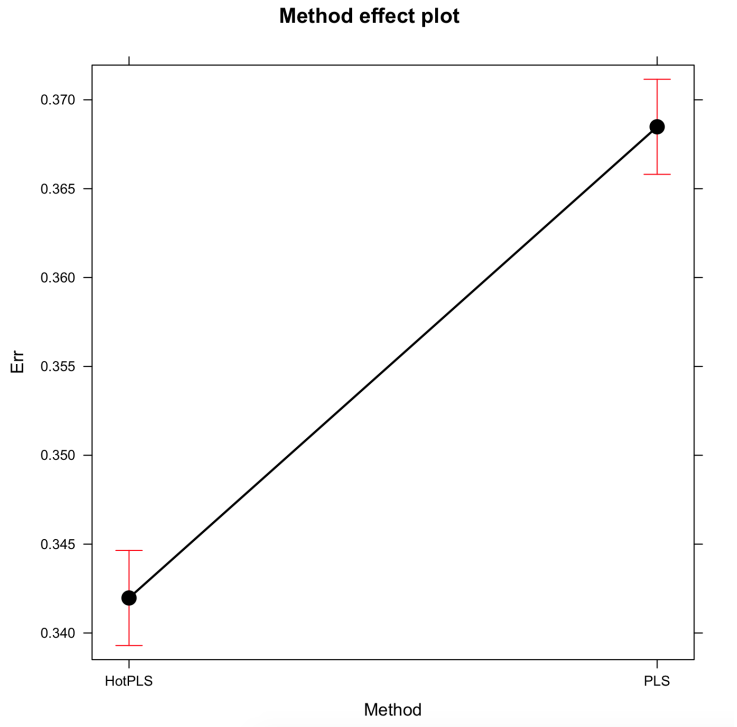



Figure 3.2: *The figure shows the main effect of the method, where the two methods are Hot PLS and PLS.*

Figure 3.2 shows that the lowest APER is on average when the method is Hot PLS.

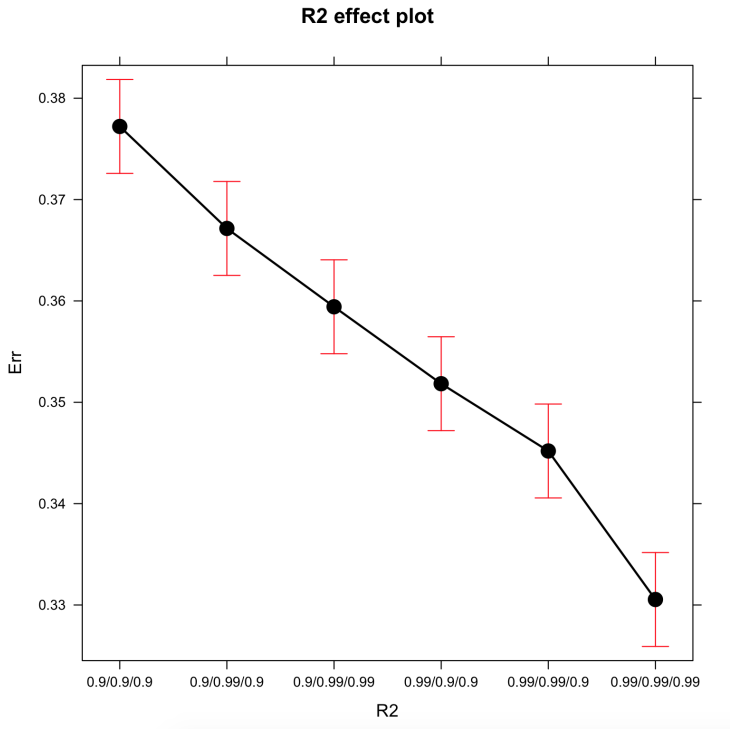



Figure 3.3: *The figure shows the main effect of $R^2$, the x-axis the different $R^2$ values, and the y-axis give the APER. The $R^2$ values are given in six sets with three values each, 0.9/0.9/0.9, the first value is the $R^2$-value for level 1 in the hierarchy, the next value is the $R^2$-value for level 2.1 in the hierarchy and the last $R^2$-value is for level 2.2 in the hierarchy*

In Figure 3.3 one can see the APER decreasing when $R^2$ is increasing. The best combination is when the $R^2$ is high $R^2 = 0.99$ at all the nodes in the hierarchy. $R^2$ stands for how much of the data which is explained by the model. An $R^2$ value of 0.99 indicates that the 99 % of the data are explained by the model. The results is intuitive, when more of the data is explained by the model the easier it gets to do the classification.

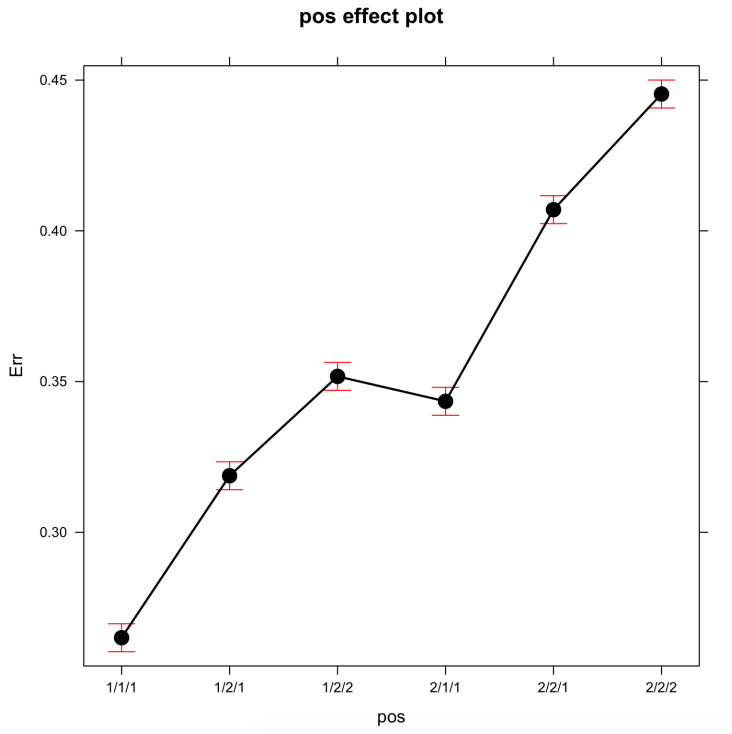



Figure 3.4: *The figure shows the main effect of relpos, the x-axis the different relpos values, and the y-axis gives the APER for Hot PLS and PLS. The relpos values are given in six sets with three values each,* 1/1/1*, the first value is the relpos vector 1 for level 1 in the hierarchy, the next value is the relpos vector 1 for level 2.1 in the hierarchy and the last relpos vector 1 is for level 2.2 in the hierarchy. Relpos has two vector which are, vector 1* [1 3] *and vector 2* [5 7]

The relpos parameter gives the components which contain the information. Table 3.4 shows that the first vector [1 3] in every level of the hierarchy gives the lowest APER value and then gives easiest classification. One interprets this as the information is in the first components and this makes it easier to find the information. More information give better classification. If

the second vector, [5  7], is used, the information will be stored in directions with less variation smaller eigenvalues which leads to harder classifications.

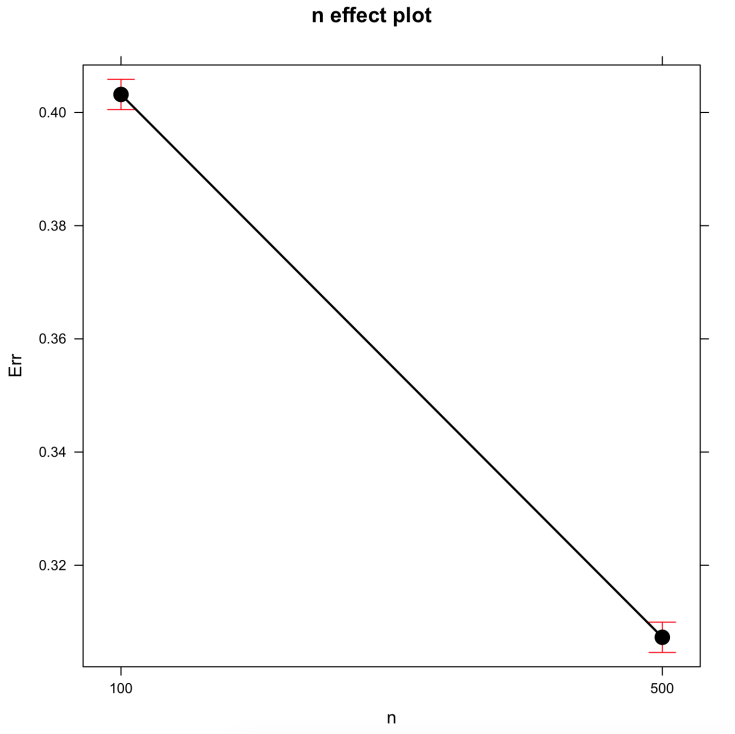



Figure 3.5: *This figure shows the main effect of n. The x axis is the number of n and the y axis the APER.*

Table 3.5 shows that higher number of observations will give a lower APER, and a lower number of observations will give a higher APER.

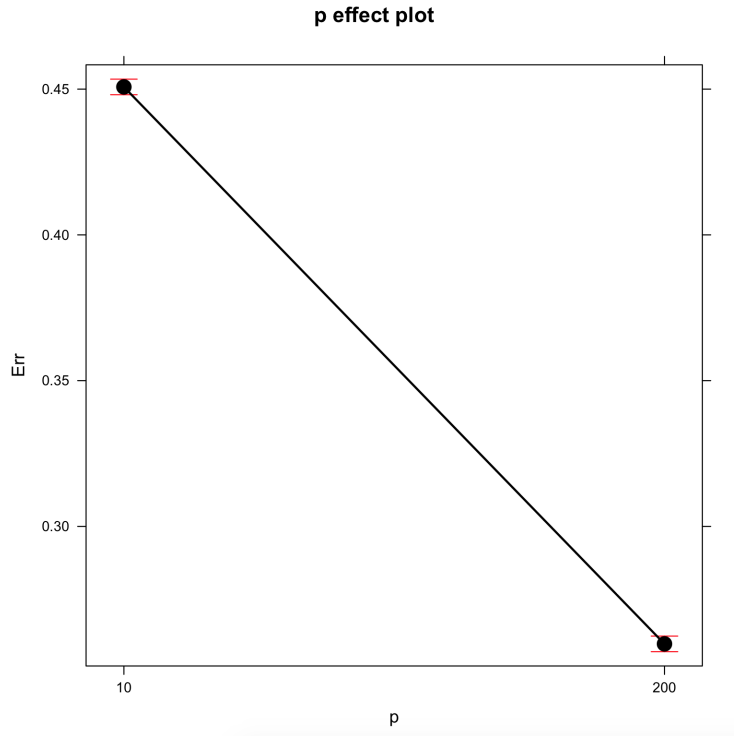



Figure 3.6: *Main effect plot of the number of variables, p, where p has two levels,* $p = 10$ *and* $p = 200$*. The y-axis is the APER value and the x-axis gives the p.*

Figure 3.6 shows that $p = 200$ will give the best APER value which means smallest classification error.

### 3.1.2 Second order interactions between design parameters

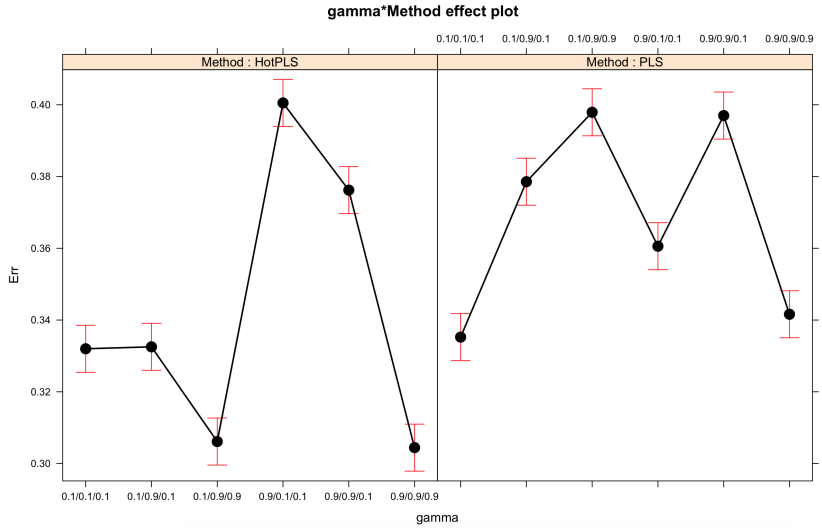



Figure 3.7: *Effect plot of the interaction between the six settings of $\gamma$ and the two methods. The plot to the left is when the method is Hot PLS and the plot on the right is when the method is PLS. The x-axis is the different levels of $\gamma$.*

From Figure 3.7 it seems as the Hot PLS performs better than PLS on average. Except when $\gamma = 0.9$ in the first level and $\gamma = 0.1$ on the second level, then the PLS is performing the best. Hot PLS performs best when level 1 has $\gamma = 0.1$, the eigenvalues decreases fast. It will also perform good if level 1 has $\gamma = 0.9$, but then both levels on level 2 must have $\gamma = 0.9$. PLS performs better than Hot PLS when $\gamma$ has the setting 0.9/0.1/0.1.

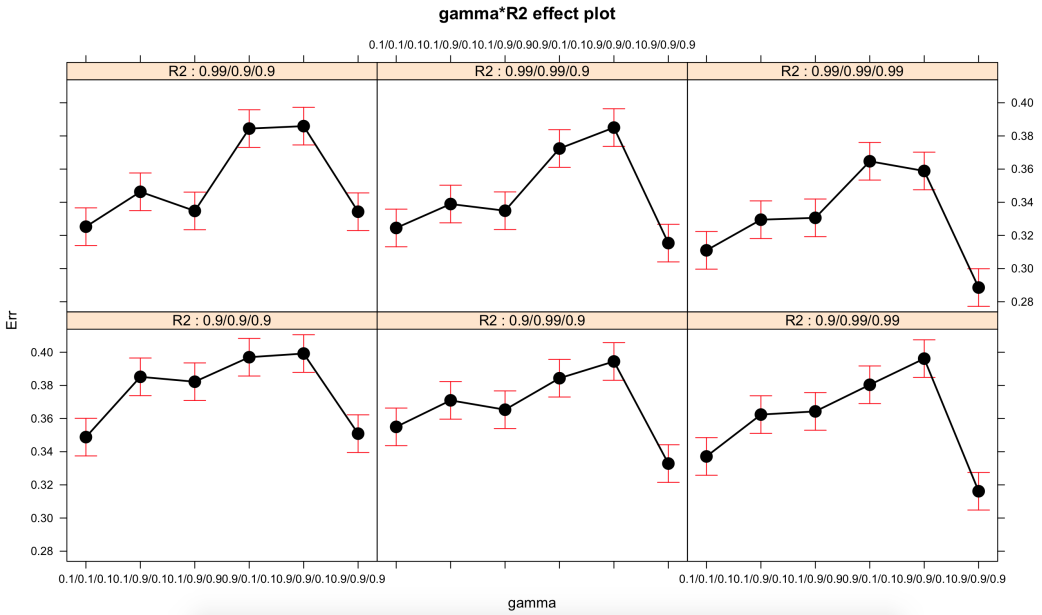



Figure 3.8: *Effect plot of the interaction between the six settings of $\gamma$ and the six settings of $R^2$. This gives in total 36 combinations. The y-axis gives the APER value, the x-axis gives the six settings of $\gamma$ and each square is one setting of $R^2$.*

Figure 3.8 shows that the more information ($R^2 = 0.99$) there is in the data, the easier it gets to classify the data upper right plot. The APER will also in general be lower when level of $\gamma = 0.1$ on level 1 and both levels on level 2 have equal value of $\gamma = 0.1$ or $\gamma = 0.9$.

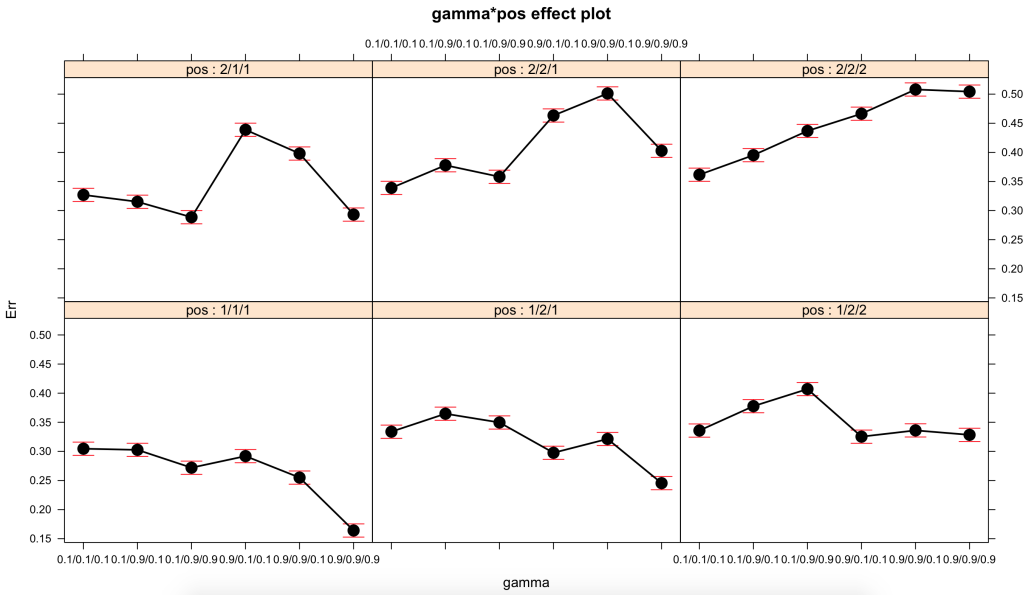



Figure 3.9: *Effect plot of the interaction between the six setting of $\gamma$ and the six settings of relpos. The y-axis give the APER, x-axis is the six settings of $\gamma$ and the six squares is one of the six settings of relpos.*

Figure 3.9 show that in general a relpos vector equal to [1 3] on level 1 will have a lower APER value than the relpos vector equal to [5 7] on the first level. The Figure also shows that the lowest APER are when the $\gamma = 0.1$ on all levels, and the relpos is set to have most of the information on the first and third components at all levels. When $\gamma = 0.9$ is in the first level and $\gamma = 0.1$ in both level2.1 and level2.2 the APER value get large. When relpos is set to [5 7] level 1 on the APER value will get a lager value than if the relpos was set to [1 3]. It is like this at level 2 as well, relpos set to [1 3] for both levels on level 2 will get a lower APER value.

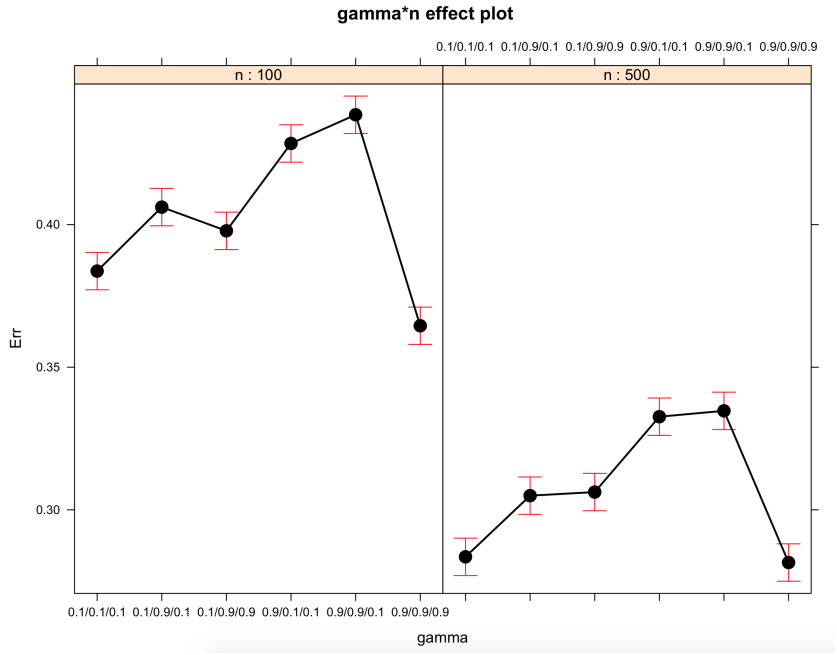



Figure 3.10: *Effect plot of the interaction between the six setting of* $\gamma$ *and the two levels of observation number* $(n = 100, 500)$*, in total 12 combinations. The y-axis gives the APER, x-axis is the six settings of* $\gamma$ *and the two sections give the number,* $n$*, of observations.*

Figure 3.10 shows a clear line between $n = 100$ and $n = 500$, where $n = 500$ gives the lowest APER. And again the APER is lowest when the first level of $\gamma = 0.1$ and both levels on level 2 have equal value of $\gamma = 0.1$ or $\gamma = 0.9$.

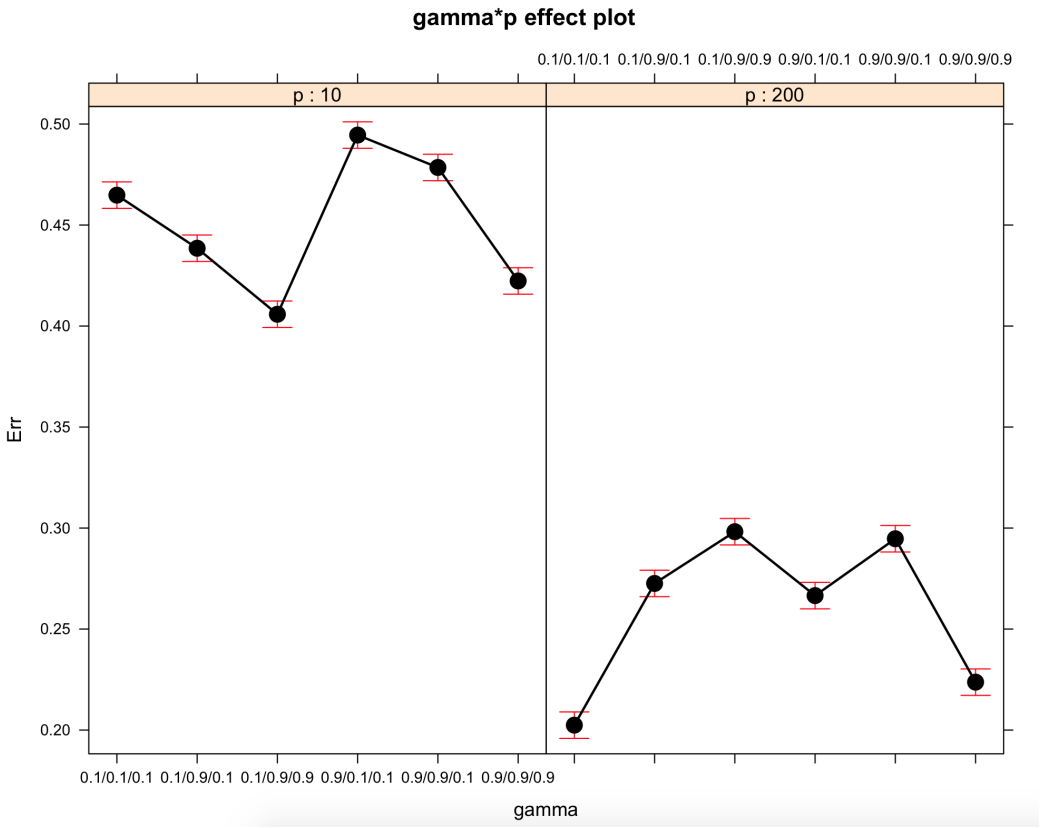



Figure 3.11: *The interaction plot between six setting of $\gamma$ and the two setting of number of explanatory variables.p. The y-axis gives the APER, x-axis is the six settings of $\gamma$ and the two sections give the number of variables.*

In Figure 3.11 one can see clearly the defference between $p = 10$ and the $p = 200$, where the $p = 200$ gives lower APER value than $p = 10$. For $p = 200$ it is easiest to classify when $\gamma$ has the same values in both levels, either $\gamma = 0.1$ or $\gamma = 0.9$. When $p = 10$ the level 1 for gamma should be $\gamma = 0.1$ and both levels in level 2 should have $\gamma = 0.9$. This result is immediately counter intuitive with regard to the effect of $p$, but will be

discussed later.

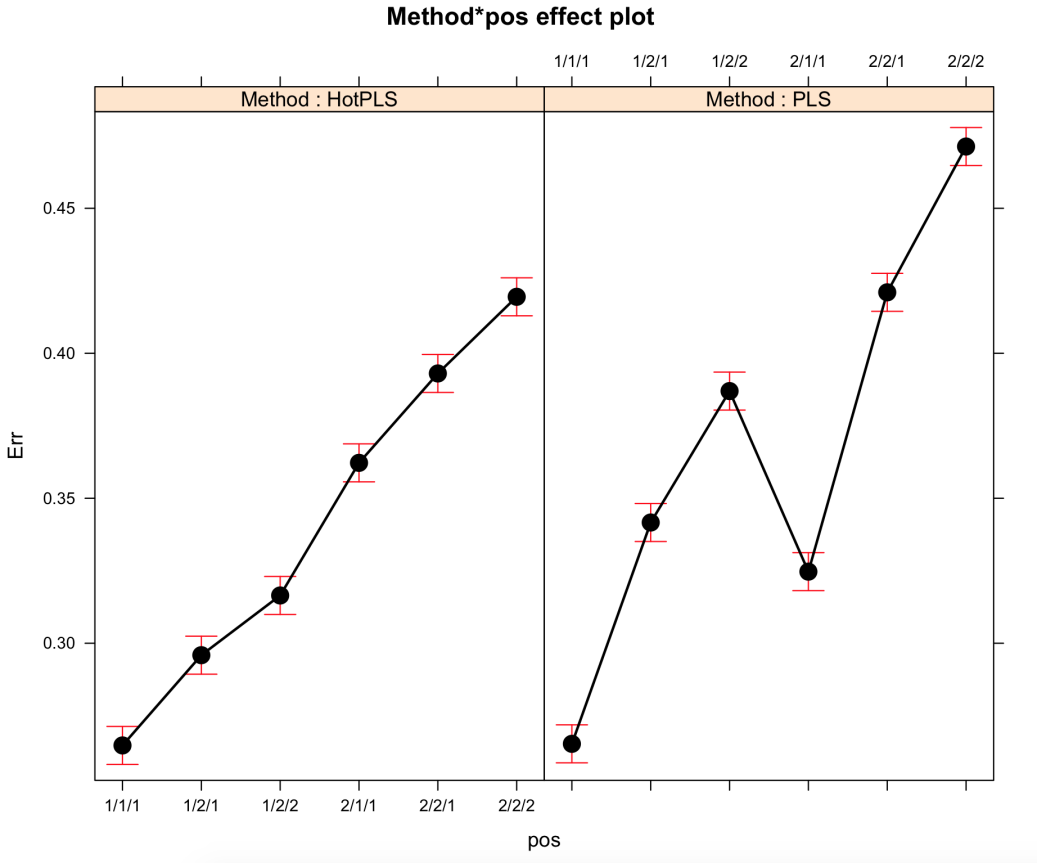



Figure 3.12: *Effect plot of the interaction between the six settings of Relpos and the methods. The y-axis gives the APER, x-axis is the six settings of relpos and the two sections are Hot PLS and PLS.*

Figure 3.12 shows that the Hot PLS will in general have the lowest APER values with one exception, which is when the relpos is components 5 and 7 in the first level and in level 2.1 and level 2.2, the relevant components is 1 and 3. In this case the PLS performs better. The best combination is when all levels are set to have the relevant components is 1 and 3. In this case

there is no difference in methods.

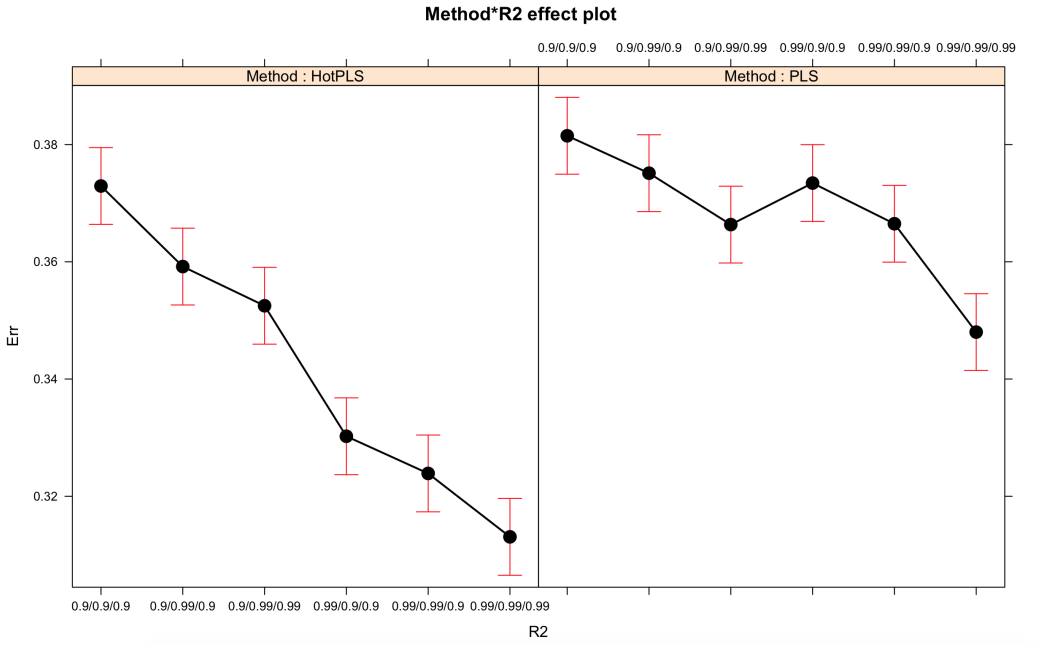



Figure 3.13: *Effect plot of the interaction between the six setting of $R^2$ and the methods. The y-axis gives the APER, x-axis is the six settings of $R^2$ and the two sections are Hot PLS and PLS.*

Figure 3.13 shows once more that the Hot PLS does it clearly better when level 1 is set to $R^2 = 0.99$ than $R^2 = 0.9$. The best combination is with Hot PLS and $R^2$ set to 0.99 on every level. $R^2 = 0.99$ is when there are a lot of information in the data. One can also notice that the PLS does a jump in the APER value when the $R^2 = 0.99$ in the first level and $R^2 = 0.9$ in the second levels.

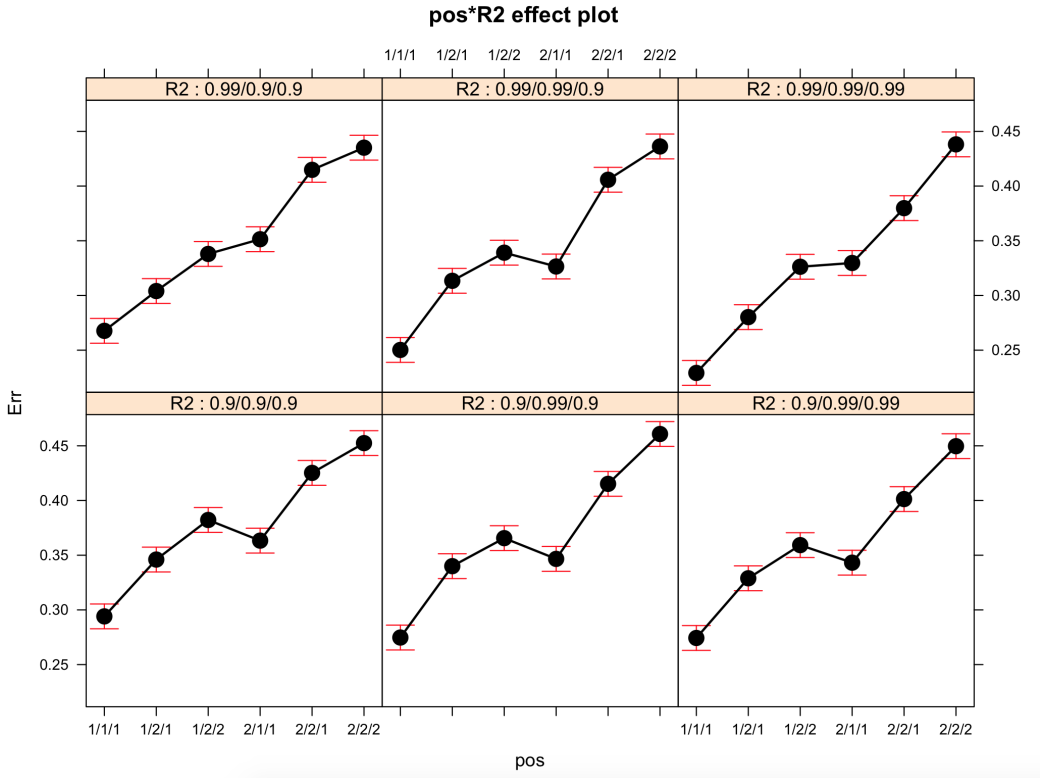



Figure 3.14: *Effect plot of the interaction between the six setting of $R^2$ and the six settings of relpos. The y-axis gives the APER, x-axis is the six settings of relpos and the six squares is one of the six settings of $R^2$*

Figure 3.14 shows that the more information ($R^2 = 0.99$) there is, the easier it is to classify correctly and especially when the information are in first components (1 and 3).

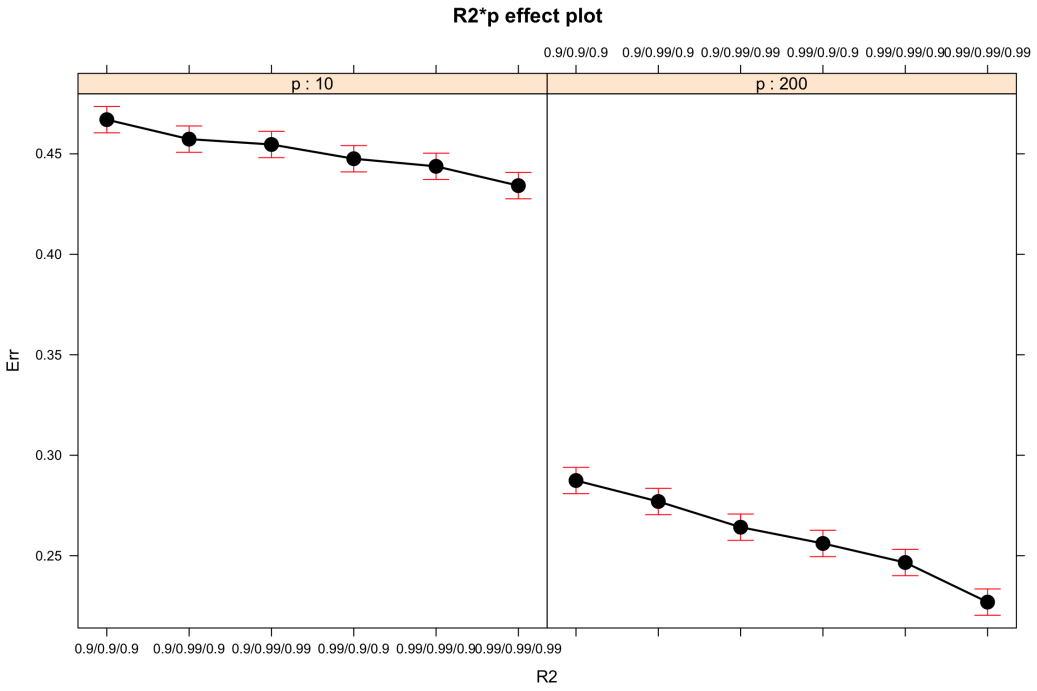



Figure 3.15: *Effect plot of the interaction between the six settings of $R^2$ and the two setting variable number, p. The y-axis gives the APER, x-axis is the six settings of $R^2$ and the two sections are Hot PLS and PLS.*

Figure 3.15 shows as before that $p = 200$ will give lower APER values than $p = 10$. The APER value also drops when there is more information in the data, this is when $R^2 = 0.99$, the best combination will be to have $p = 200$ and $R^2 = 0.99$ in each level of the the hierarchy. One can also see that the effect of $R^2$ is lager when the $p = 200$.

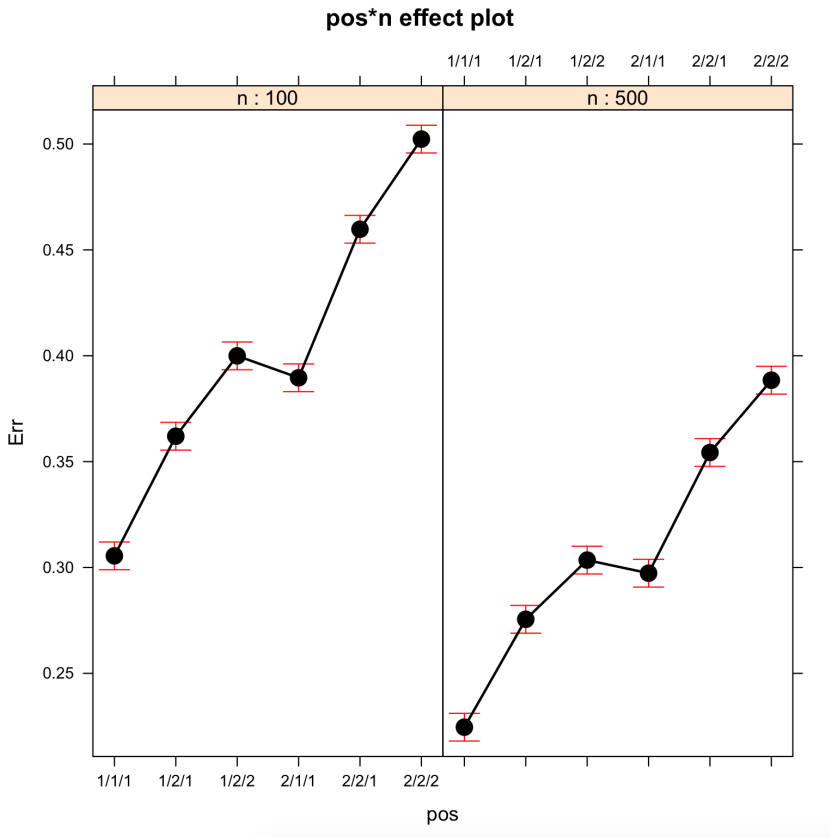



Figure 3.16: *Effect plot of the interaction between the six settings of relpos and the two levels of observation number* ($n = 100, 500$). *In total 12 combinations. The y-axis gives the APER, x-axis is the six settings of relpos and the two sections give the number of observations.*

Figure 3.16 shows that the lowest APER value is reached with $n = 500$ and the relpos in all the levels is 1 and 3. One notice that the APER value increasing when the amount of relpos vector 2 increases one can also see a small jump in the APER value when the relpos settings in level goes from having relpos vector 1 to have relpos vector 2.

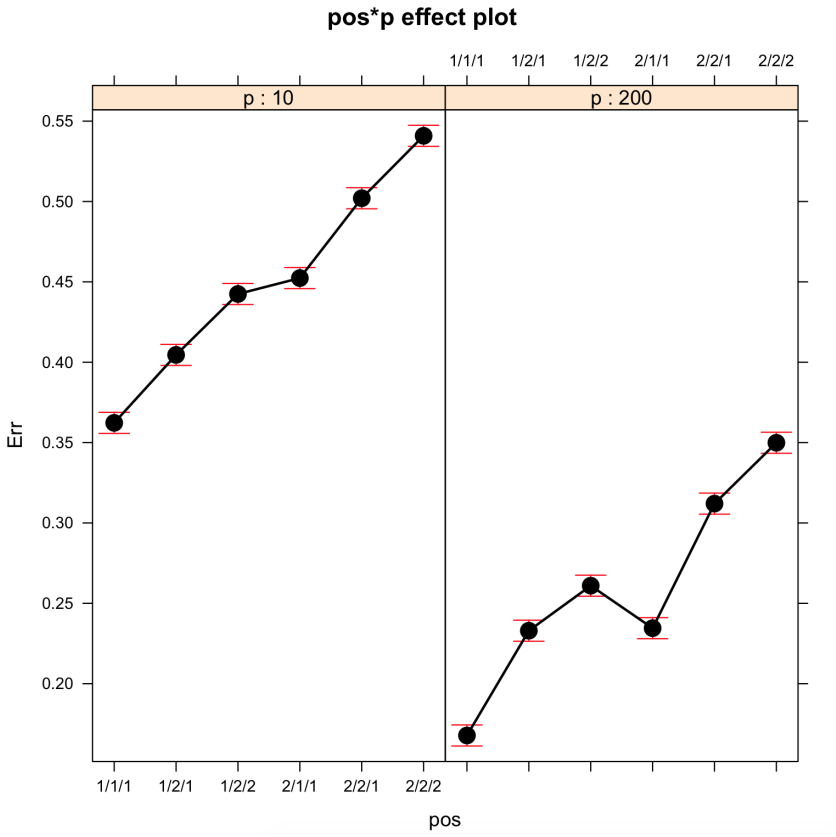



Figure 3.17: *Effect plot of the interaction between the six settings of relpos and the number of variables. The y-axis gives the APER, x-axis is the six settings of* $R^2$ *and the two sections are for* $p = 10$ *and* $p = 200$.

Figure 3.15 shows as before that $p = 200$ will give lower APER values than $p = 10$. The APER value also drops when the information in the data are in relpos vector 1 changes to relpos vector 2. The best combination will be to have $p = 200$ and relpos vector 1 in each level of the hierarchy.

### 3.1.3 Third order interactions between design parameters

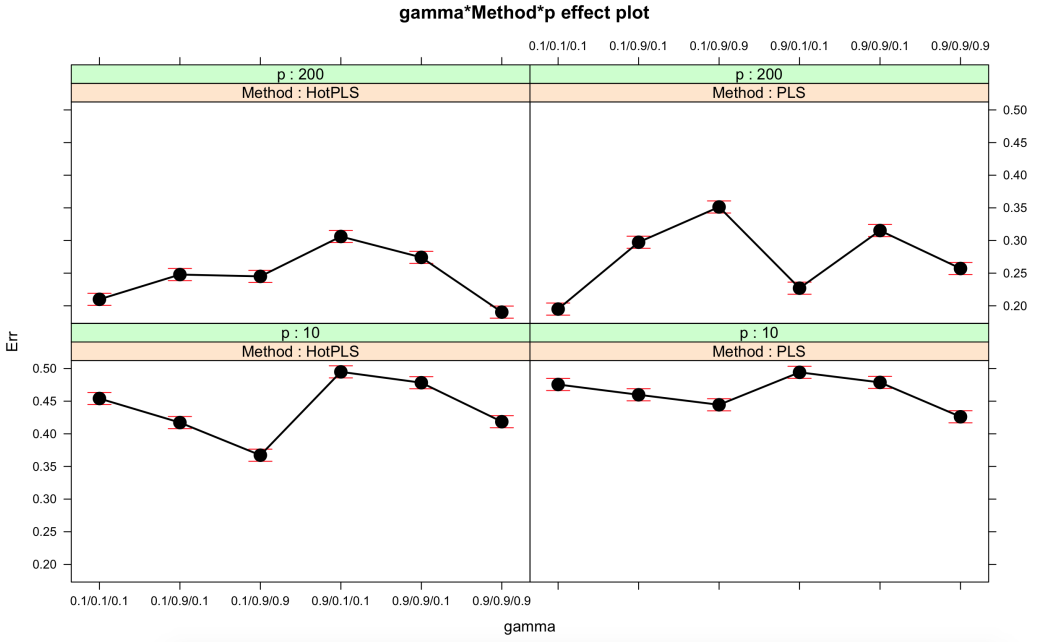



Figure 3.18: *The interaction plot of the six settings of $\gamma$, the two settings of $p$ and the two methods. The y-axis gives the APER values, the x-axis give the $\gamma$ settings and the sections have the different combinations of method and $p$.*

Figure 3.18 shows that $p = 200$ will give the lowest APER values. Also the Hot PLS seems to be better than the PLS, with the exception when $\gamma = 0.9$ in level 1 and $\gamma = 0.1$ in both levels on level 2. The best combination is when the method is Hot PLS, $p = 200$ and $\gamma = 0.9$ in all the levels of the hierarchy or when the method is PLS, $p = 200$ and $\gamma = 0.1$ in all the levels of the hierarchy. These two combinations has the lowest and almost equal APER value.

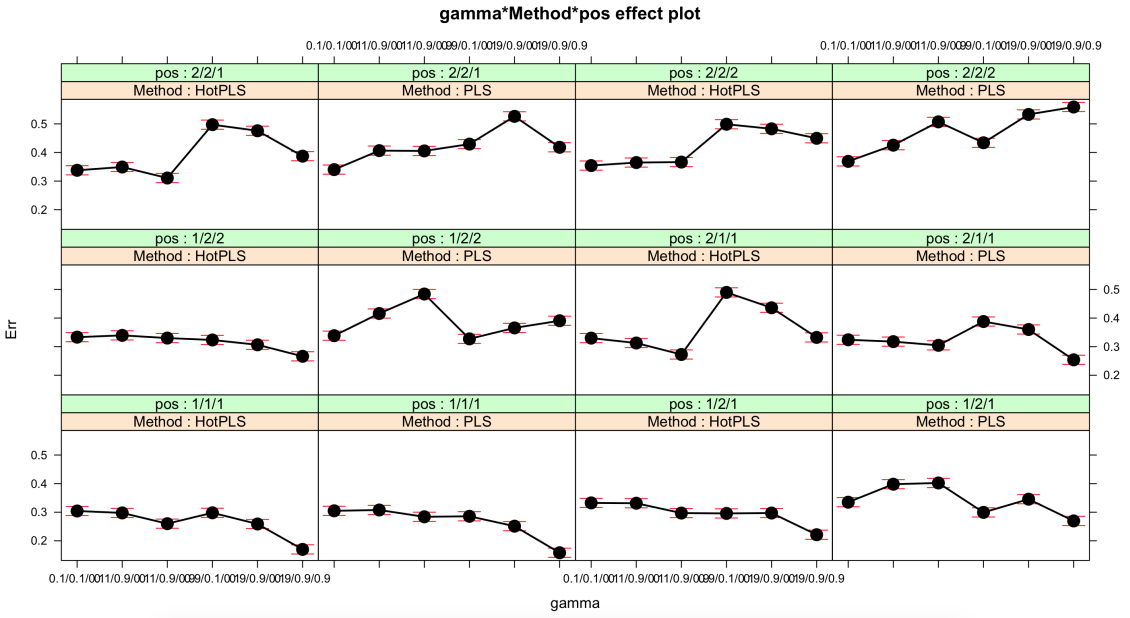



Figure 3.19: *The interaction plot of the six settings of $\gamma$, the six settings of relpos and the two methods. The y-axis gives the APER values, the x-axis give the $\gamma$ settings and the sections have the different combinations of method and relpos.*

Figure 3.19 shows mainly that Hot PLS performs better than compared with PLS, but PLS is in some cases better than Hot PLS, this is when $\gamma = 0.9$ on level 1 and on level 2 $\gamma = 0.1$. The APER value is also lower when the the information in the data are contained in the first and third variable (relpos vector 1) and the APER value is also lower when the eigenvalues decreases quickly, $\gamma = 0.9$. The best results with these factors is reached when the method is PLS, relpos vector 1 for all the levels in the hierarchy and $\gamma = 0.9$ for all the levels in the hierarchy.

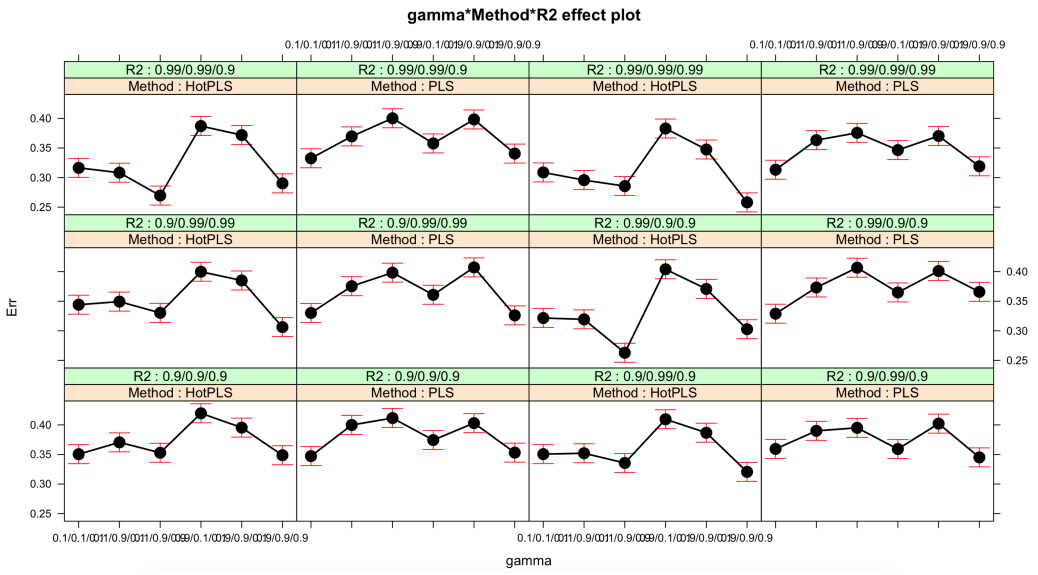



Figure 3.20: *The interaction plot of the six settings of $\gamma$, the six settings of $R^2$ and the two methods. The y-axis gives the APER values, the x-axis gives the $\gamma$ settings and the sections have the different combinations of method and $R^2$.*

The Figure 3.20 shows in general that Hot PLS preforms better than PLS with exception when the $\gamma = 0.9$ on level 1 and $\gamma = 0.1$ on both levels in level 2, and when $R^2 = 0.9$ on level 1 and $R^2 = 0.99$ on both levels in level 2. The APER value is smaller when there are more information in the data, $R^2 = 0.99$ and the eigenvalue drops quickly. The best combinations of these three factors are when using the Hot PLS, $R^2 = 0.99$ for all levels in the hierarchy and $\gamma = 0.9$ for all levels in the hierarchy.

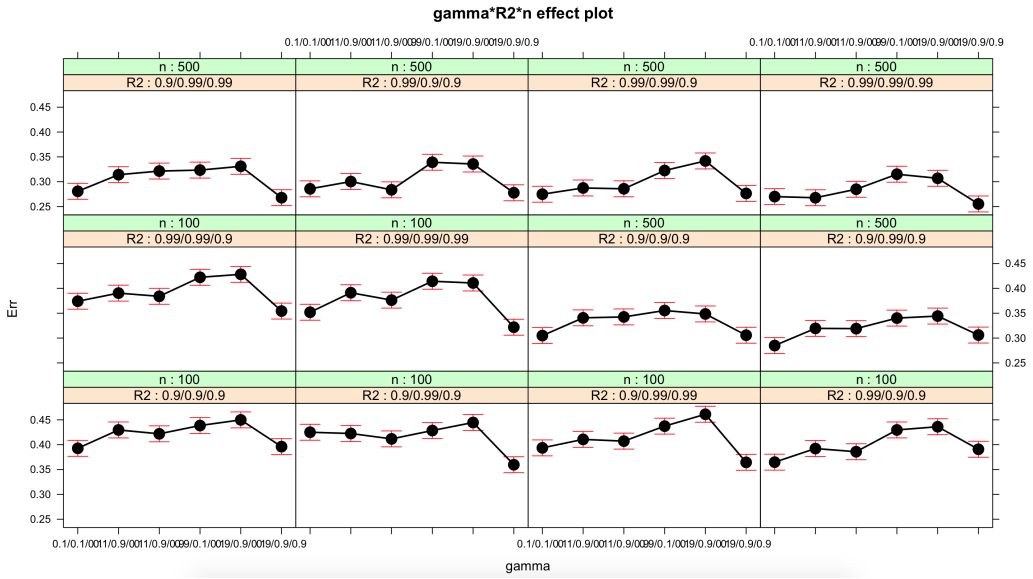



Figure 3.21: *The interaction plot of the six settings of $\gamma$, the six settings of $R^2$ and the two settings of $n$. The y-axis gives the APER values, the x-axis gives the $\gamma$ settings and the sections have the different combinations of $n$ and $R^2$.*

The Figure 3.21 shows that $n = 500$ tends to have lower APER values than $n = 100$. Also when there are much information in the data ($R^2 = 0.99$) the APER value tends to be lower than with less information ($R^2 = 0.90$). The $\gamma$ seems to have the lowest APER values when level 1 has either $\gamma = 0.1$ or $\gamma = 0.9$. The best combination of these three factors will therefore be $n = 500$, $R^2 = 0.99$ on all levels in the hierarchy and $\gamma = 0.1$ or $\gamma = 0.9$ on level 1 and $\gamma = 0.9$ on both levels in level 2.

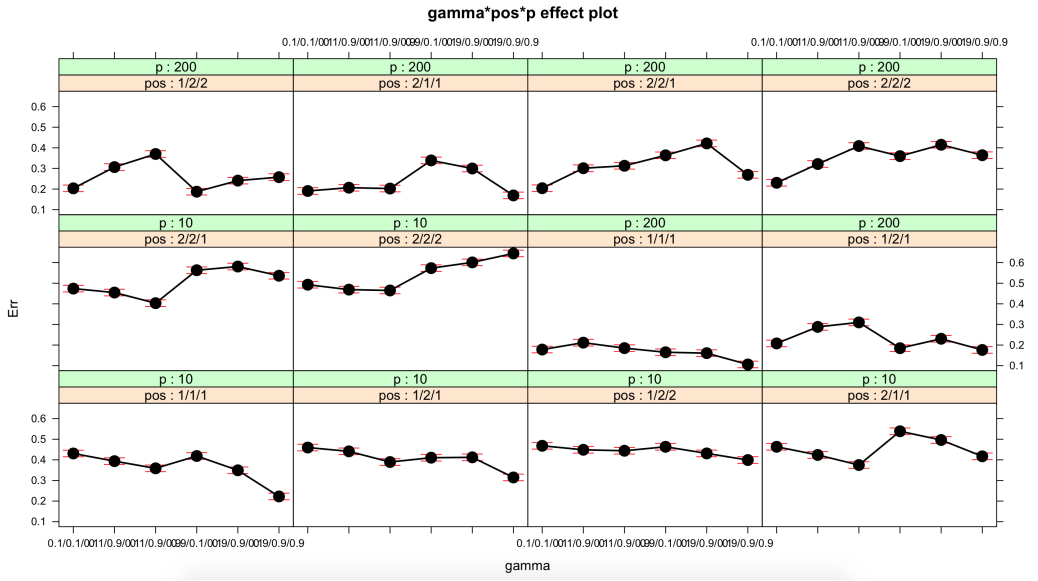



Figure 3.22: *The interaction plot of the six settings of $\gamma$, the six settings of relpos and the two settings of $p$. The y-axis gives the APER values, the x-axis gives the $\gamma$ settings and the sections have the different combinations of $p$ and relpos*

The first observation to make from Figure 3.22 is that $p = 200$ will give lower APER values than the $p = 10$. As seen earlier also that information in the first variables seems to have to have a positive effect on the APER value when the eigenvalues drops quickly, ($\gamma = 0.9$), for all levels of the hierarchy. When $\gamma$ goes from $\gamma = 0.1$ in level 1 to $\gamma = 0.9$ in level 1 the APER values gets higher. When this happen the $\gamma$-values on level 2 will be set to $\gamma = 0.1$. The best combination of these three factors will be to have $p = 200$, relpos vector 1 for all the levels of the hierarchy and $\gamma = 0.9$ for all the levels of the hierarchy.

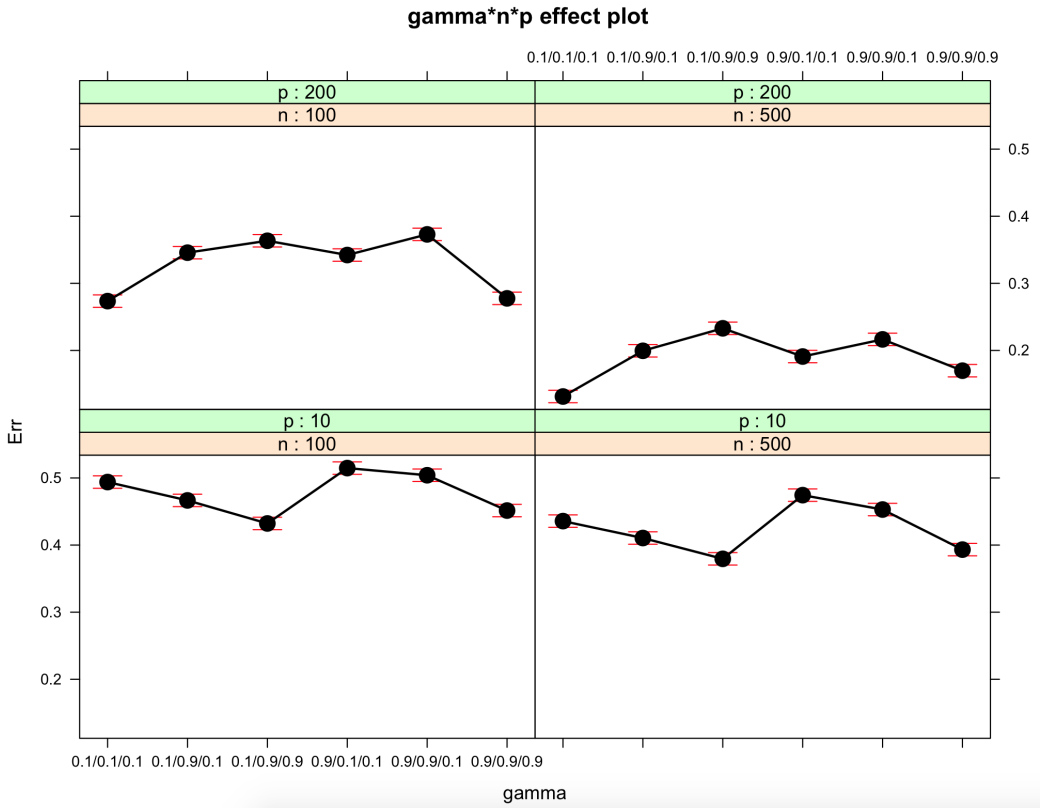



Figure 3.23: *The interaction plot between the six settings of $\gamma$, the number of observations and the number variables. The y-axis gives the APER value, the x-axis gives the $\gamma$ settings and each squares give one of the four combinations of number of observations and the number of variables.*

The Figure 3.23 shows that in general $p = 200$ will give good results, low APER values. Also $n = 500$ will give lower APER values than $n = 100$. The Figure 3.23 also show a jump between $\gamma = 0.1$ and $\gamma = 0.9$ on the first level, where $\gamma = 0.1$ in general give the best results. The best combination of these three factor is to have similar value of $\gamma$ either $\gamma = 0.1$ or$\gamma = 0.9$, $p = 200$ and a large $n, n = 500$.

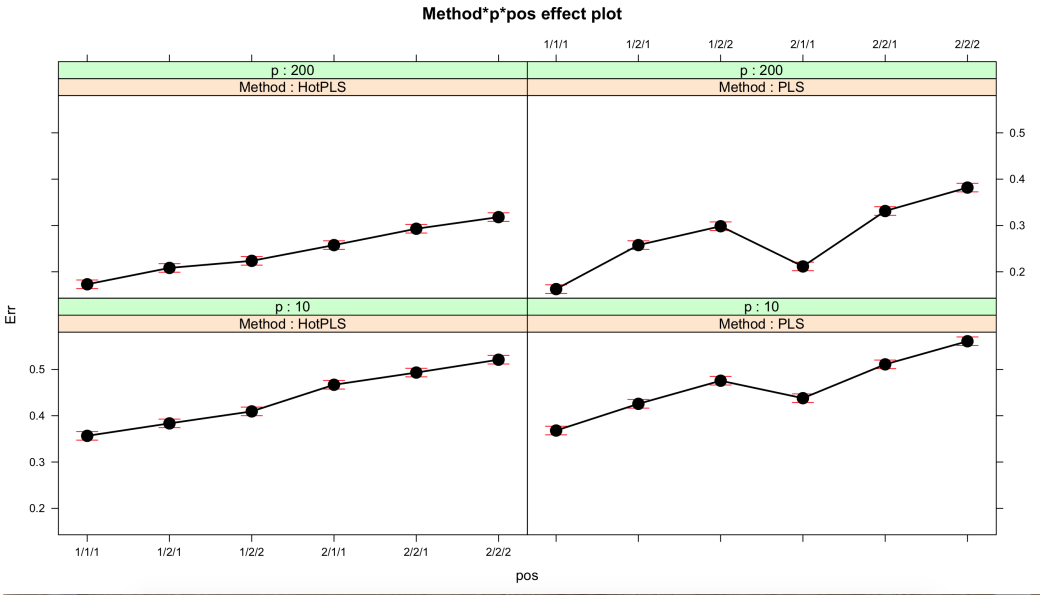



Figure 3.24: *The interaction plot between the six settings of relpos, the two methods and the number variables. The y-axis gives the APER value and each squares give one of the four combinations of the methods and the number of variables.*

The Figure 3.24 shows the clear difference between the $p$ levels, where the $p = 200$ gives better APER values than $p = 10$. The relpos give the best result when it is set to vector 1 for all levels. In general it seems that Hot PLS is slightly better then PLS. But the PLS has a jump where relpos change from vector 1 to vector 2 in the first level. The best combination is when $p = 200$, relpos is vector 1 for all levels and the method is PLS.

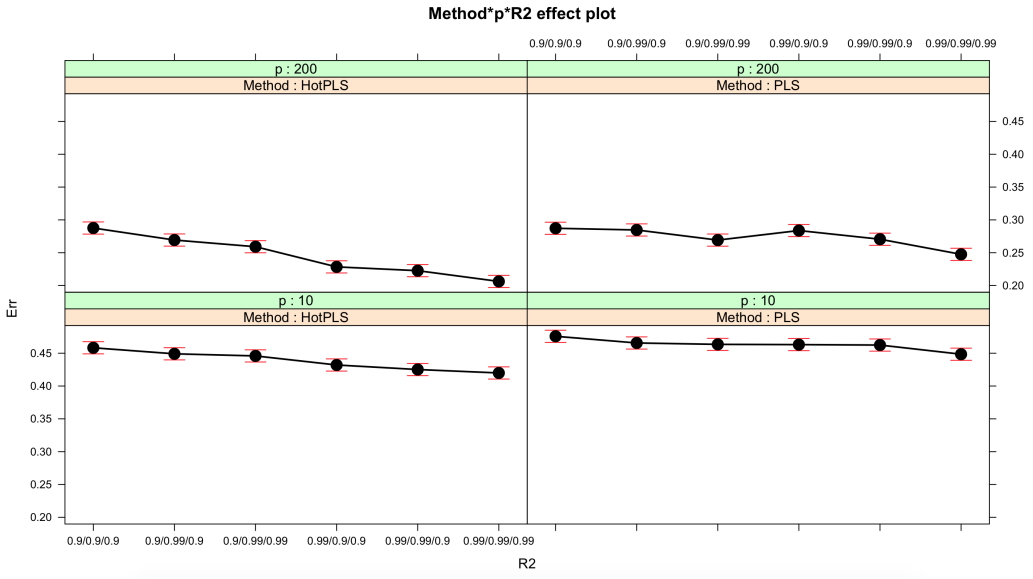



Figure 3.25: *The interaction plot between the six settings of $R^2$, the two methods and the number variables. The y-axis gives the APER value, x-axis give the $R^2$ setting and each squares give one of the four combinations of the methods and the number of variables.*

The Figure 3.25 shows a very clear difference between the $p$ levels. When $p = 200$ will the lowest APER values. Hot PLS is in general the better method. The lowest APER value will be achieved with $R^2 = 0.99$ in the levels, $p = 200$ and the method is Hot PLS.

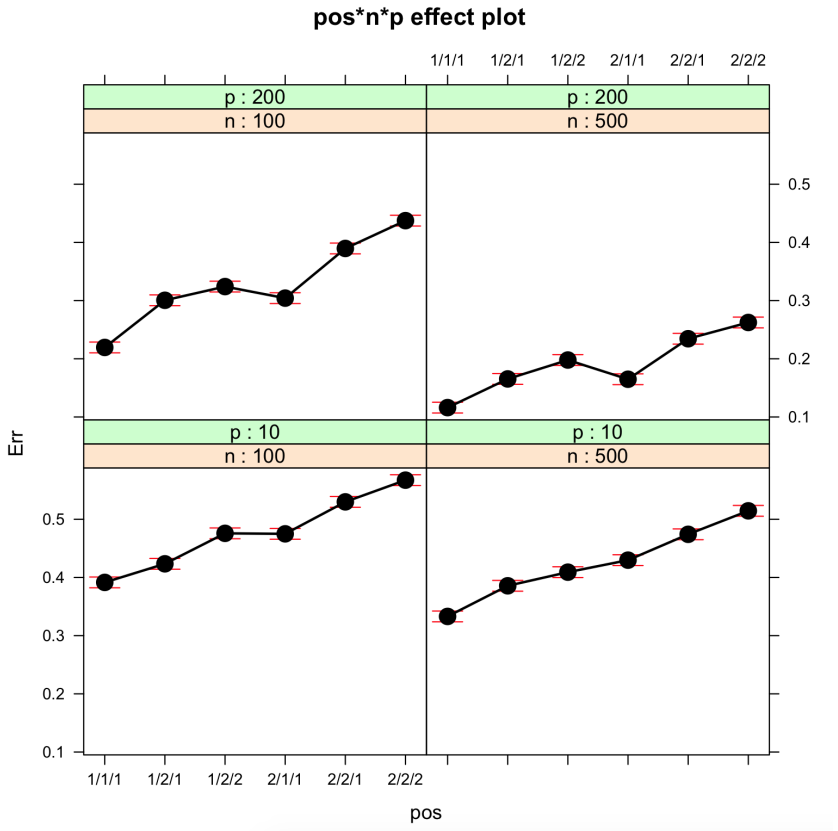



Figure 3.26: *The interaction plot between the six settings of relpos, the number of observations and the number variables. The y-axis gives the APER value, the x-axis gives the six settings of relpos and each squares give one of the four combinations of number of observations, $n$ and the number of explanatory variables, $p$.*

The Figure 3.26 shows a clear difference between $n = 100$ and $n = 500$, where $n = 500$ will give the lowest APER value. There is also a clear difference between $p = 10$ and $p = 200$, where the $p = 200$ will perform the best and get the lowest APER values. The Figure 3.26 shows also that the more information the are in the first variables (variables number 1 and 3).

The best combination for this interaction is when the $n = 500$, $p = 200$ and relpos has the variables [1  3] for all levels in the hierarchy.

## 3.2 Comparison of Hot PLS with an extended classifier set

There was also run an analysis where one had five different methods; Hot PLS, PLS, LDA, QDA, and 3NN, where all classifiers worked on the second level in the hierarchy with four classes. This means that the method effect has five levels in the ANOVA (see table A.1). The next part of the results will compare these five different classifiers.

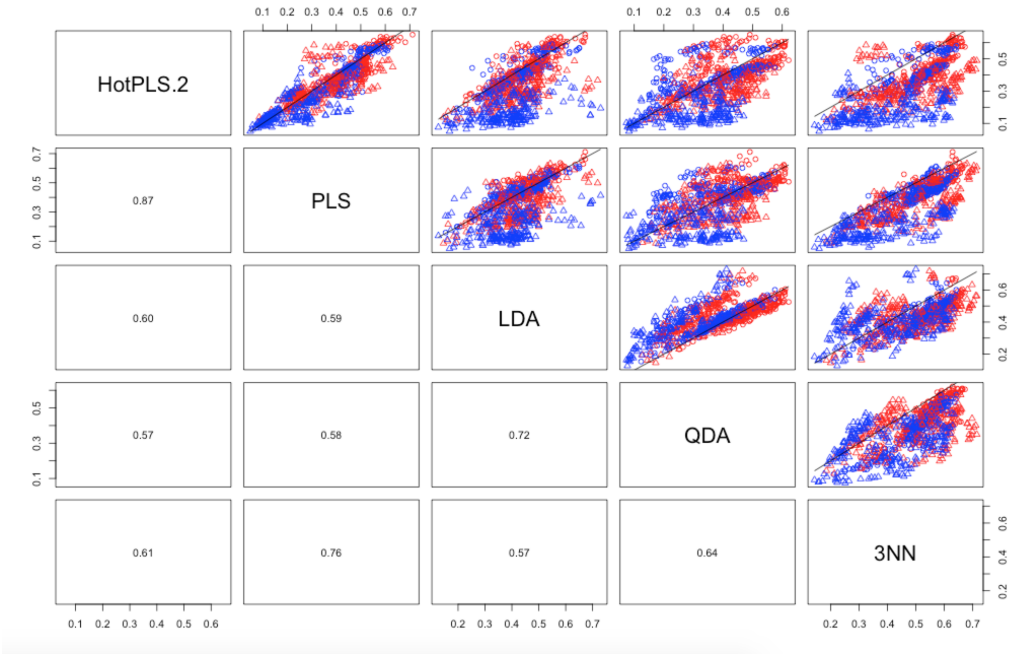



Figure 3.27: *This matrix plot show the methods on the diagonal and on the lower triangle give the correlation between the different methods for all the runs in the experiments design. The upper triangle has scatterplot between the APER values for the different methods. The different symbols shows how many explanatory variables the observations have, if it is a circle* $p = 10$ *and if it is a triangle* $p = 200$*. The color coding define the number of observations, blue is when* $n = 500$ *and red is when* $n = 100$*. When there are most observations under the line it means that the method that is on the y-axis will have the lowest APER values. And if there are most observation above the line it means that the method on the x-axis will have the lowest APER value.*

From Figure 3.27 one can read that the Hot PLS will do it clearly better than LDA, QDA and 3NN. It is also better than PLS, but it is not as clear

as with the others. The PLS is also better than LDA and 3NN, but it seems like PLS and QDA are close together. The LDA on the other hand seems to do it better than 3NN, but worse than QDA. The QDA performs much better than the 3NN. A short summary, 3NN is the method which has the highest APER value.

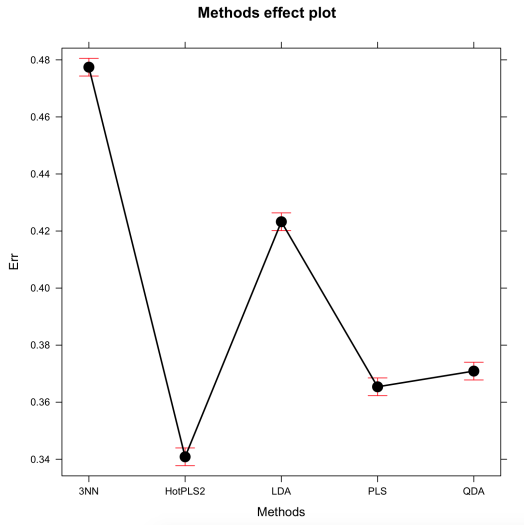



Figure 3.28: *The main effect plot of the methods. Where the y-axis gives the APER value and the x-axis give the method.*

The Figure 3.28 confirms what the Figure 3.27 showed. Hot PLS has lower APER value than the rest, and the 3NN has the highest APER value of all the methods. One can also see that the PLS and the QDA (run on PCA-scores) are close to have the same APER value.

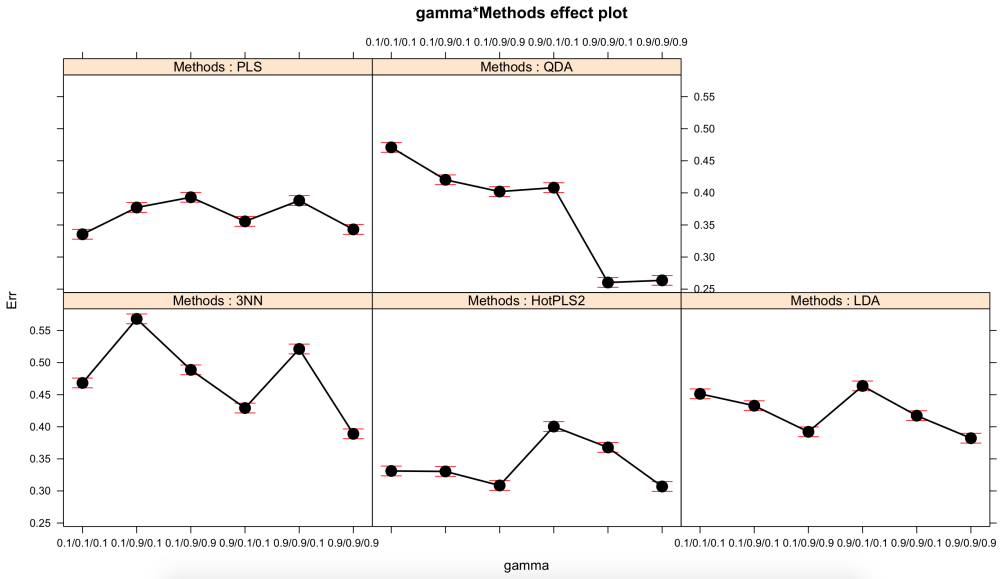



Figure 3.29: *Effect plot of the interaction between the six setting of $\gamma$ and the five methods. There are five squares, one for each method. The x-axis is the different levels of $\gamma$.*

The Figure 3.29 shows that it is the Hot PLS which performs best, hence the lowest APER values. The best $\gamma$ combination for Hot PLS is when $\gamma = 0.9$ in both levels on level 2. Also when $\gamma = 0.1$ in the first level will the Hot PLS perform better. For the other methods it also seems to be a good choice to have $\gamma = 0.9$ in both levels on second level. QDA stand out by having the two best APER values when $\gamma = 0.9$ on the first level and either $\gamma = 0.9$ on both levels in the second level or one of the second level has $\gamma = 0.9$ and the other has $\gamma = 0.1$.

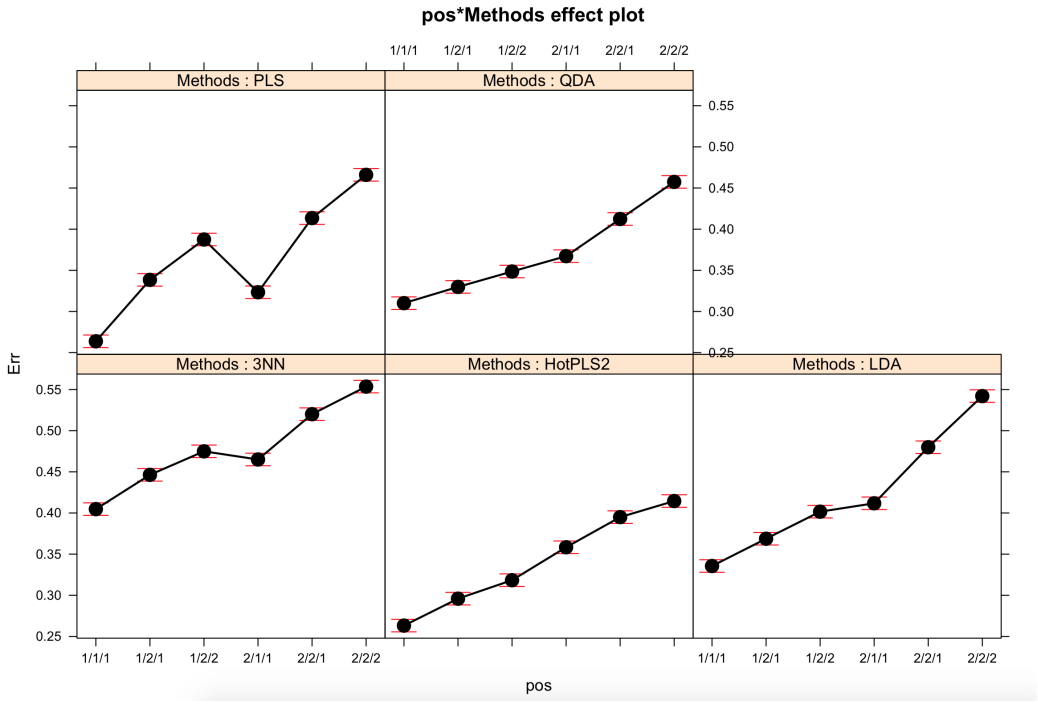



Figure 3.30: *Effect plot of the interaction between the six setting of relpos and the five methods. There are five squares, one for each method. The x-axis is the different levels of relpos.*

The Figure 3.30 supports what previous figures have showed, that relpos with vector 1 for all levels, gives the lowest APER value and that the APER value will increase when the amount of vector 2 increases. It is the Hot PLS which has the lowest APER values among the methods. From previous figures one has observed a drop in the APER value when the first level in hierarchy has vector 2 and in the second levels it has vector 1 in the PLS method. This is also observed in Figure 3.30, but here one can also observe this for 3NN, QDA and LDA where the slope is less steep than in the Hot PLS for this change in relpos values.

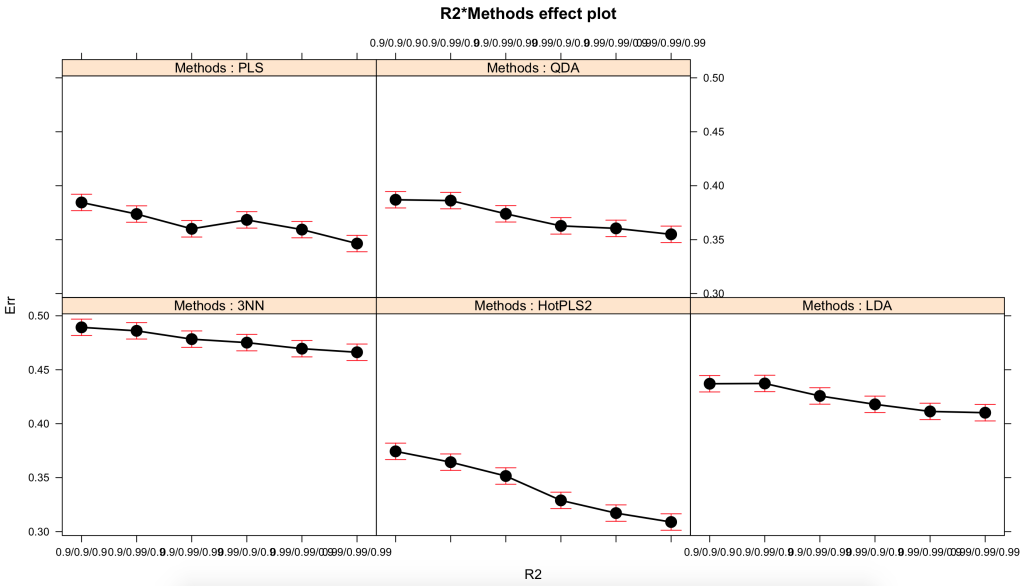



Figure 3.31: *Effect plot of the interaction between the six setting of* $R^2$ *and the five methods. There are five squares, one for each method. The x-axises is the different levels of* $R^2$.

The Figure 3.31 shows again that the higher $R^2$ will give more information in the data that will give lower APER values. Also in this figure the Hot PLS will have the lowest APER values and 3NN will have the highest APER values. One can also in this figure observe that the PLS does a jump in the APER value when the $R^2 = 0.99$ in the first level and $R^2 = 0.9$ in the both levels on the second level.

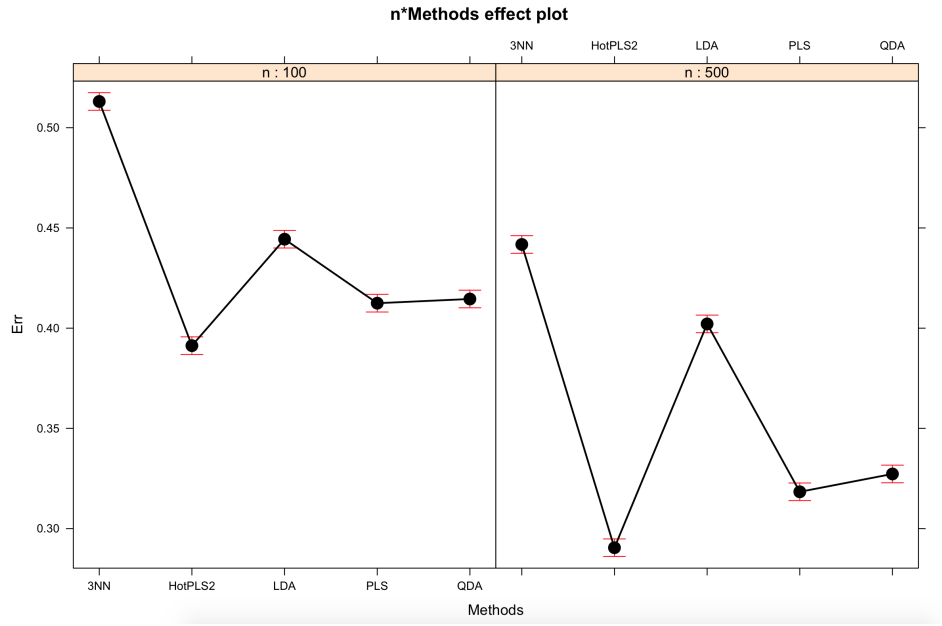



Figure 3.32: *Effect plot between the five methods and the two setting of $n$, where the y-axis is the APER value, x-axis give the method and the square give the number of observation, $n$.*

The Figure 3.32 one can recognize the same pattern as in the main effect plot in Figure 3.28. When $n = 500$ the APER value will be lowest.

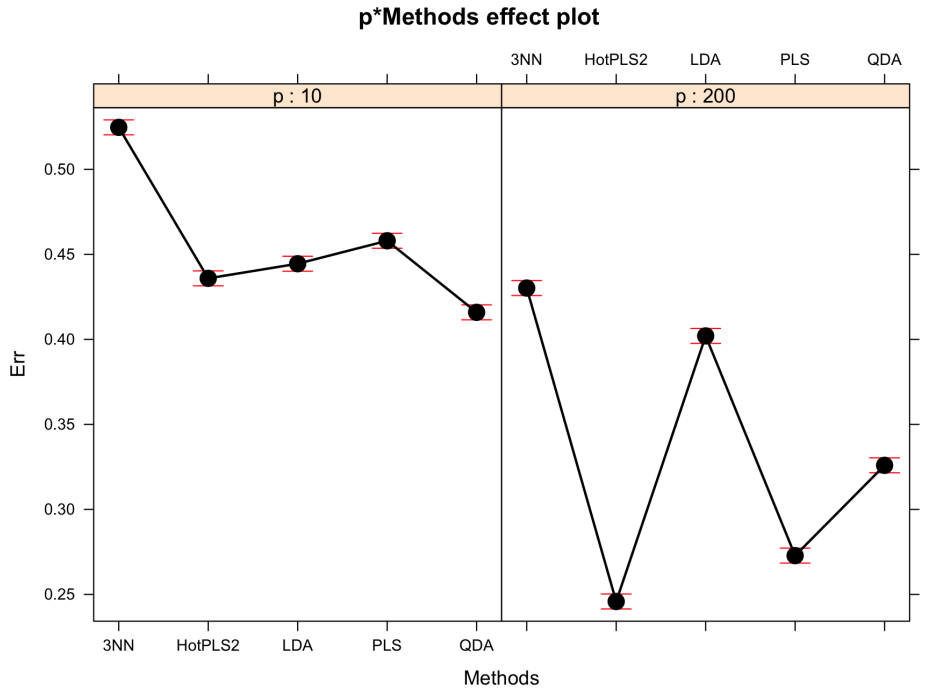



Figure 3.33: *Effect plot between the five methods and the two setting of p, where the y-axis is the APER value, x-axis gives the method and the two panels give the number of variables, p.*

The Figure 3.33 shows that all the methods performs better when $p = 200$, than $p = 10$. The effect of methods on APER increases when $p = 200$.

# Chapter 4

# Discussions

## 4.1 Summary of the results

In order to have a low classification error in a hierarchical case it is important to have as much information as possible in the data i.e $R^2 = 0.99$, see Figures: 3.3 and 3.13. In the simulation this can be decided, but in real data one cannot decides the simulation parameters.

Figure 3.4 shows that data with information stored in the first and third component (relpos vector 1) on all levels in the hierarchy will give the lowest APER values. If the information is stored in the first and third components a fast decreasing eigenvalue is desirable i.e. $\gamma = 0.9$ is preferable in every level, see Figure 3.1. The combination of these two setting, $\gamma = 0.9$ and relpos vector 1 is a dream pair (Figure 3.9), because a large $\gamma$ give a large drop in the eigenvalues which means that most of the variation is in the direction of the first component. The relpos vector 1 has the information in the first and third components. Thus such a combination makes it easy to classify.

Hot PLS is the preferable method when a clear hierarchical structure is given. This is the case in these data sets which can be seen in Figures 3.27 as well as 3.2.

The large number of observations will also lead to a better classification with less classification errors which is intuitive. Number of observations will also affect the other factors. The effect of $\gamma$ increases with smaller $n$. In other words, the impact of the different levels of $\gamma$ becomes larger see Figure 3.10. One can also see that the relpos is affected by the number of observations. The effect of the relpos is large if $n$ is small, see Figure 3.16. If the $n$ is small it is hard to find the information when it is stored in the 5th and 7th components. The five different methods are also affected by $n$. A large $n$ gives a big difference between the methods which can be found in Figure 3.32. The number of explanatory variables should also be large in order to give a low classification error. This results was unexpected. It was assumed that $p = 10$ would give the lowest classification error. The $n/p$ relation is often seen as a signal to noise relation. A high $n/p$ value is usually associated with much signal (information) and low noise, in this view the lowest APER was expected for $n = 500$ and $p = 10$. One can see this result as some kind of reverse of the Simpson's paradox. Simpson's paradox may occur if one explanatory variable with a significant effect on the response is left out. The effect of other variables will be reversed which makes it harder to explain. Because the data are simulated the $R^2$ is locked at one value. In real data the $R^2$ increases with the number of explanatory variables which leads to a limitation for the other parameters resulting in a counter intuitive result. If this is the case it is a result of how simrel is implemented. It is important to

notice that the results for $p$ is still unclear and need further investigation.

### 4.1.1 A closer look at Hot PLS and PLS

Looking at interaction plots including methods one finds that PLS performs better than Hot PLS when classification in level 1 is difficult and classification in level 2 is easy see Figure 3.12. This can be explained by the fact that Hot PLS starts with classifying level 1 and move on to the next level. If the classification is difficult the Hot PLS makes more wrong classifications because the mistakes follow to the next level. The PLS on the other hand classifies directly on level 2, thus it is not affected by the difficulties of level 1. The opposite holds too, i.e. if it is simple to classify on level 1 and challenging on level 2 the Hot PLS shows better performance since it has the opportunity to classify correctly on level 1 and the PLS can only classify on level 2 where the classification is hard.

Figure 3.13 shows that the difference between Hot PLS and PLS is not very big if $R^2 = 0.90$ on level 1. On the other hand if $R^2 = 0.99$ holds on level 1 there is a large gap between Hot PLS and PLS, suggesting Hot PLS has the most to gain when it is easy to classify on level 1.

When it comes to choose either Hot PLS or PLS one has to investigate whether the different levels in the hierarchy are informative or not. Non-informative levels should be left out. These levels are found by running a CV where the classification errors for different hierarchical structures are compared, for example with and without individual levels in the hierarchy.

### 4.1.2 A closer look at QDA of Figure 3.29

Figure 3.29 shows that there are two setting of $\gamma$ where QDA performs very well. In order to get a better understanding of these results an interaction plot between the five different methods, the six settings of $\gamma$ and the six settings of relpos is given in Figure 4.1

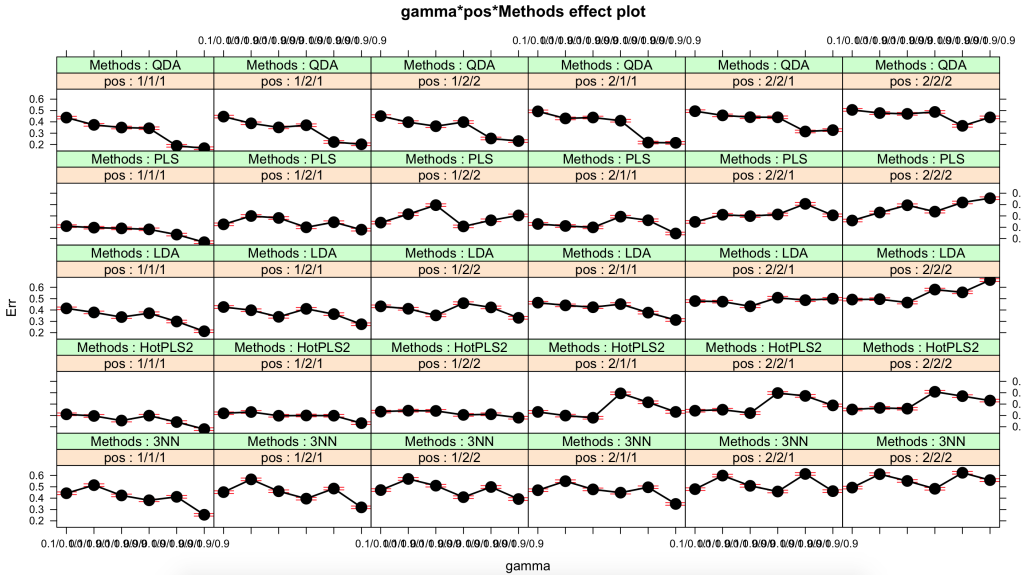



Figure 4.1: *Effect plot between the five methods, the six settings of $\gamma$and the six settings of relpos. The y-axis is the APER value and the x-axis gives the $\gamma$ setting. The squares are the relpos settings and the methods.*

In Figure 4.1 one can see that QDA has a lower classification error than the other methods if $\gamma = 0.9/0.9/0.9$ or $\gamma = 0.9/0.9/0.1$ on all the settings of relpos, especially when relpos has vector 2 in the first level. As shown earlier the vector 2 makes it harder for Hot PLS to classify correctly, and thus gives a higher classification error. Furthermore QDA is doing better than the other

methods when relpos and $\gamma$ have unequal values for the same of the levels in the hierarchy. In this case the four classes have different variance matrices, which is exactly the assumption made by the QDA model.

In the simulation the number of components was set permanently to $a = 8$, we also know that there is maximal two relevant components and that component number 7 is potentially is relevant with the lowest eigenvalue. When doing this we get the optimal value of $a$ for LDA and QDA. With real data would use CV to decide the amount of components to use as input for LDA and QDA, this will lead to bigger insecurity for both methods. LDA and QDA does it probably better than they should in this assembly. The PLS-methods and KNN knows in a way less about these data properties.

### 4.1.3 Other ways to do a hierarchy PLS

Hot PLS knows the hierarchically structure in advanced. This is not the only way of managing classification in a hierarchical structure. In [Tøndel et al. 2011] they explain how to perform a hierarchical cluster-based partial least squares regression (HC-PLSR) in the gene regulatory of mice. HC-PLSR uses a PLSR model to provide the PLS scores which is needed to divide the observation into different groups by a fuzzy C-means (FCM) clustering. This is a method to find natural groups in the data. After this the PLSR model goes through each discovered group and then runs FCM clustering on the PLS scores. This is done until no more natural group is available. Tøndel's HC-PLSR is a method like Hot PLS where the hierarchal structure is not known *a priori*, which uses Fuzzy clustering for establishing the structure, then uses this in a similar way as in Hot PLS.

## 4.2 Further research

For further studies it would be interesting test the methods on real data. This is something that had been interesting to explore more thoroughly. A interesting dataset to look in to would be bacteria data, classify bacteria based on the DNA. Other dataset where one can find hierarchal structure is the classification of moulds, like Liland used in [Liland et al. 2014].

The Simrel package in R simulated data in a way as the PLS wants the data to be. Simrel give data that are normally distributed and linear data. Checking how Hot PLS will perform on data that are not normally distributed and non-linear would be interesting.

Hot PLS assumes to know the hierarchal structure before the classification, it would be interesting to see what will happen if the assumed structure is wrong. One could compare the Hot PLS with Tøndels HC-PLSR.

The Hot PLS is using PLS to classify on each level in a hierarchal structure, one could also try to other classifiers (KNN, LDA or QDA) instead of PLS.

In this thesis it was used five different simulation parameters which gave 864 different combinations of experiments that have been run. It could be interesting is to expand the parameter space and then run more experiments. When this is said one should also take a look at the parameter $p$. To find out why the high $p$ gives the lowest APER values.

## 4.3 Conclusion

Hot PLS has the advantage of knowing the hierarchical structure. By starting at the top of the hierarchy, the method will carry more information that will help with the classifications in lower levels. This can also be a disadvantage if the method does a mistake in a higher level will the mistake follow down through the hierarchical structure. As long the data has hierarchical structure with levels which are easy to classify Hot PLS will perform well.

# Appendix A

# R commander tables

Analysis of Variance Table

Response: Err

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) | |
|---|---|---|---|---|---|---|
| gamma | 5 | 3.5088 | 0.7018 | 323.0914 | < 2.2e-16 | *** |
| pos | 5 | 13.5818 | 2.7164 | 1250.5969 | < 2.2e-16 | *** |
| R2 | 5 | 0.7616 | 0.1523 | 70.1240 | < 2.2e-16 | *** |
| n | 1 | 6.7691 | 6.7691 | 3116.4482 | < 2.2e-16 | *** |
| p | 1 | 15.6658 | 15.6658 | 7212.4681 | < 2.2e-16 | *** |
| Methods | 4 | 10.3504 | 2.5876 | 1191.3152 | < 2.2e-16 | *** |
| gamma:pos | 25 | 4.5214 | 0.1809 | 83.2647 | < 2.2e-16 | *** |
| gamma:R2 | 25 | 0.1974 | 0.0079 | 3.6347 | 2.722e-09 | *** |
| gamma:n | 5 | 0.0582 | 0.0116 | 5.3570 | 6.502e-05 | *** |
| gamma:p | 5 | 3.0411 | 0.6082 | 280.0205 | < 2.2e-16 | *** |
| gamma:Methods | 20 | 7.1226 | 0.3561 | 163.9596 | < 2.2e-16 | *** |

```
pos:R2          25  0.1588  0.0064    2.9245 1.531e-06 ***
pos:p            5  0.1096  0.0219   10.0953 1.268e-09 ***
pos:Methods     20  0.9192  0.0460   21.1593 < 2.2e-16 ***
R2:p             5  0.0468  0.0094    4.3136  0.000646 ***
R2:Methods      20  0.1686  0.0084    3.8806 1.232e-08 ***
n:p              1  0.4900  0.4900  225.5935 < 2.2e-16 ***
n:Methods        4  0.4709  0.1177   54.2007 < 2.2e-16 ***
p:Methods        4  3.6105  0.9026  415.5638 < 2.2e-16 ***
Residuals     4134  8.9792  0.0022
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table A.1: *Anova table of the significant factors up to third interaction after model simplifications by backwards/forward elimination of non-significant effects. The methods in this table is Hot PLS, PLS, LDA, QDA and KNN*

# Appendix B

# R-code

The programming for this thesis is done in R version 3.1.2 (2014-10-31) and is uploaded to https://bitbucket.org/hannebrit/master-thesis/overview. Some of these codes are written by Kristian Liland and Solve Sæbø which are described in the link.

# Bibliography


Breiman, L., J. Friedman, R. Olhsen, and C. Stone (1984). *Classification and Regression Trees*. Wadsworth, Inc.

Har-peled, S., D. Roth, and D. Zimak (2002). "Constraint Classification for Multiclass Classification and Ranking". In: *Advances in Neural Information Processing Systems* 14.

Indahl, U., K. Liland, and T. Næs (2009). "Canonical partial least squares - a unified PLS approach to classification and regression problems". In: *Journal of Chemometrics* 23.495–504.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013a). *An Introduction to Statistical Learning*. Springer, pp. 39–42.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013b). *An Introduction to Statistical Learning*. Springer, p. 179.

Johnson, R. A. and D. W. Wichern (2007a). *Applied Multivariate Statistical Analysis*. 6th. Person Education International, pp. 584–587.

— (2007b). *Applied Multivariate Statistical Analysis*. 6th. Person Education International, pp. 593–594.

Kendall, M. G. (1957). *A course in Multivariate Analysis*. Griffin, pp. 300–303.

Liland, K. H., A. Kohler, and V. Shapaval (2014). "Hot PLS - a framework for hierachically orded taxonomic classification by partial least squares". In: *Chemometrics and Intelligent Laboratory Systems* 138, pp. 41–47.

Martens, H. and T. Næs (1989). *Multivariate Calibration*. Wiley, pp. 116–165.

Montgomery, D. C. (2009a). *Design and Analysis of Experiments*. John Wiley Sons, Inc, pp. 63–75.

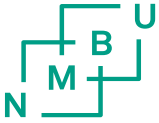
— (2009b). *Design and Analysis of Experiments*. John Wiley Sons, Inc, pp. 208–215.

Sæbø, S., T. Almøy, and I. Helland (2015). "A versatile tool for linear model data simulation based on the concept of a relevant subspace and relevant predictors." In: *Chemometrics and Intelligent Laboratory Systems*. (accepted for publication).

Sæbø, S. (2015). *simrel: Linear Model Data Simulation and Design of Computer Experiments*. R package version 1.1-0. URL: `http://CRAN.R-project.org/package=simrel`.

Tøndel, K., U. G. Indahl, A. B. Gjusland, V. J. O., P. Hunter, S. W. Omholt, and H. Martens (2011). "Hierarchical Cluster-based Partial Least Squares Regression (HC-PLSR) is an efficient tool for metamodelling of nonlinear dynamic models". In: *BMC Systems Biology* 5:90.

Wold, H. (1966). *Estimation of principal components and related model by iterative least squares*. Multivariate Analysis, Krishnaiah PR (ed.). Academic Press: New York, pp. 391–420.



Norwegian University
of Life Sciences

Postboks 5003
NO-1432 Ås, Norway
+47 67 23 00 00
www.nmbu.no