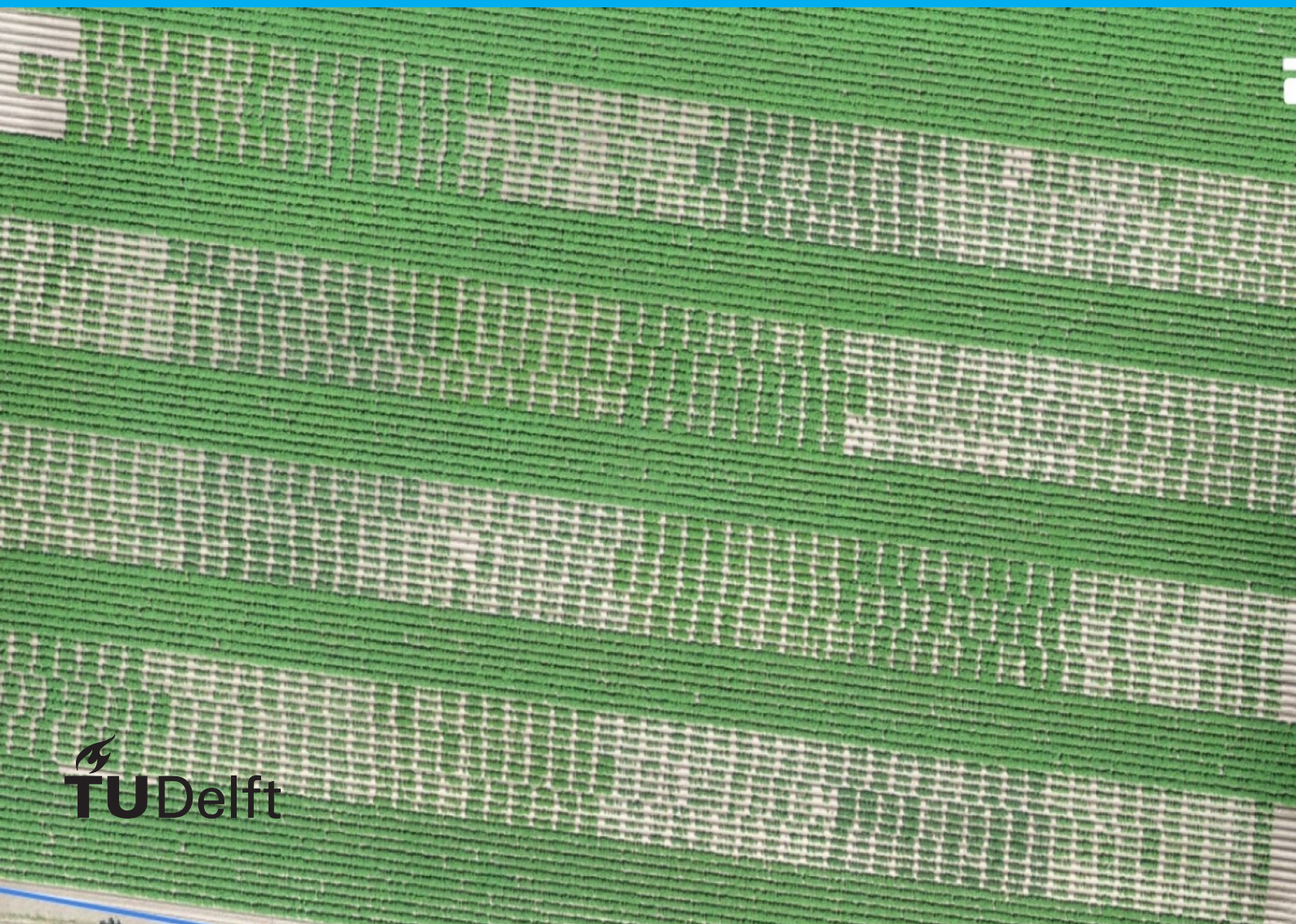
Partial Least Squares on Hyperspectral Data

A data-driven approach for predicting the growth of potato plants

Danish Rehman Khan



Partial Least Squares on Hyperspectral Data

A data-driven approach for predicting the growth of potato plants

by

Danish Rehman Khan

to obtain the degree of Bachelor of Science
at the Delft University of Technology,

Student number:	4715462	
Project duration:	April 19, 2021 – July 19, 2021	
Thesis committee:	Dr. N. Budko,	Associate Professor, supervisor
	E. Atza,	PhD student, co-supervisor
	Dr. N. Parolya,	Assistant Professor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Partial Least Squares op Hyperspectrale beelden

Een datagedreven aanpak voor het voorspellen van de groei van aardappelplanten

door

Danish Rehman Khan

ter verkrijging van de graad van Bachelor of Science
aan de Technische Universiteit Delft,

Student number:	4715462	
Project duration:	April 19, 2021 – July 19, 2021	
Thesis committee:	Dr. N. Budko,	Associate Professor, supervisor
	E. Atza,	PhD student, co-supervisor
	Dr. N. Parolya	Assistant Professor

TU Delft

Abstract

The potato is a plant that can grow from another potato tuber. Farmers and producers have to contend with a declining quality of soil and are producing less vital potato tubers. The differences of emergence speed between batches of the same variety of potato tubers cannot be explained. This is why Project 'Flight to Vitality' (FtV) is developing a diagnostic test to make predictions about the vitality of different varieties of potato tubers, with various measurements. One of these measurements consists of hyperspectral images taken from potato tubers, prior to planting. Another measurement is the actual growth of the potato plants, which was recorded after emerging.

This thesis attempts to relate both measurements by means of regression. After grouping both data sets on each batch of tubers, we are left with an explanatory matrix that has more columns than rows. This causes an ordinary least squares model to overfit on new data. Therefore an alternative regression model is proposed, which is Partial Least Squares (PLS). The goal of PLS is to find latent variables, that maximizes the covariance between both sets, to explain the variance in the explanatory space as good as possible and to provide good predictions on the response space.

The predictive performance of PLS, which is for the most part measured with the Mean Squared Error, relies on the number of latent variables used. Cross-validation is performed for finding this number. By splitting the data into training and testing sets, we evaluate PLS and find that some of the variance is explained well with PLS. This model is extended by including variable selection on the explanatory variables, which are the frequency bands taken from hyperspectral imaging. With this extension, the same experiment were rerun and we saw that the predictive performance can increase. However, the number of variables omitted seem to be inconsistent across different experiments. We have also found that the performance can slightly decrease, but the number of omitted variables remain much more consistent (and were overlapping across different experiments).

With this established baseline model, we have also looked at the predictive information on different tuber parts. By conducting a similar experiment, it appears that the model is capable of explaining almost all variance and performs outstanding. We have also looked at leaving one tuber variety out before training the model. This variety is then used for evaluation. From this experiment it seems that there is a strong variety-dependent component found in the FtV data, which makes it impossible to predict the relative vitality of the tubers. Finally we compare PLS with another model that belong to the same Krylov Subspace method, which is LSQR. The same experiment is conducted on both model and it seems like PLS and LSQR are not identical in terms of finding the number of latent variables used. Regardless, as an independent regression estimator, LSQR is much faster than PLS.

Contents

1

Introduction

The potato is a plant that grows from one potato tuber planted underground. As shown in Figure 1.1 multiple potatoes can grow from this tuber for consumption or for it to be reused for growing new potato plants. As this plant grows, the stems develop leaves. The more (and larger) leaves they have, the more potatoes it can harvest. The number of tubers that one plant can produce depends on numerous factors, for example the available moisture and soil nutrients [5], the potato variety and its genetics [6] and the volume of cortical and pith-storage cells [19].

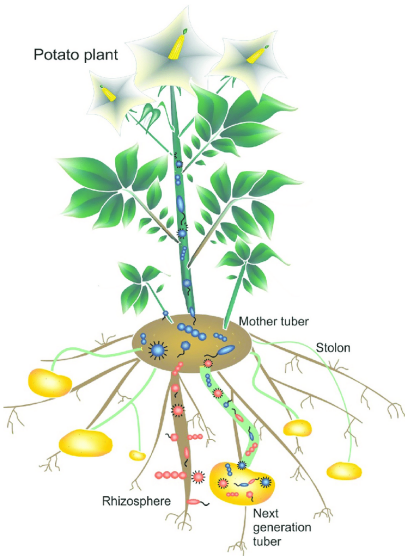


Figure 1.1: An illustration of the morphology of one potato tuber, made by Buchholz et al. [3]

The potato growers' sector have to contend with a declining quality of the soil structure and as a result are producing less vital potato tubers [11]. During the progeny of potato tubers, differences of emergence speed regularly occur between batches of tubers having the same variety. These difference often cannot be explained.

1.1. Project 'Flight to Vitality' and their goal

Prior to planting, the farmers and the producers are interested in the so called vitality of the potato tuber, which is defined as the number of tubers emerging from the soil and the rate of growth [6]. The current approach is to place a number of potatoes in a bucket for a week, and see how many became rotten. Furthermore, the potatoes need to be dug out for inspection, which is a wasteful and time-consuming process.

Project 'Flight to Vitality' is developing a diagnostic test that provides insights into factors that have impact on the vitality of multiples batches of potato tubers. With such a test, it is possible to make predictions about the vitality of these batches, given a number of measurements that they deem important. The growth of the plant itself, one of these measurements, is recorded with drones equipped with cameras. This measurement is taken after emergence. Another camera was used for the potato tubers, prior to planting. However, these cameras are capable of capturing images at wavelengths beyond the visible light. Images derived from such wavelengths are known as hyperspectral images and are a valuable source of information [21]. A pattern, or rather a curve, arises from the absorbed or reflected radiation at each band of wavelengths. Both the hyperspectral curves and the growth of the plant were recorded for six different potato varieties, in varying circumstances, such as location (and thus the soil) and climate. Each variety consists of thirty batches of multiple tubers, that have grown into plants. This gives 180 different batches of potatoes.

1.2. Regression on Hyperspectral images

Most images are captured in three colors: red, green and blue (rgb) and thus only captures information from three wavelengths. Hyperspectral images (HSI) on the other hand can be broken down into more colors by splitting the three colors into more bands. This extends to wavelength that are outside the visible spectrum, such as ultraviolet and infrared and thus reveals information that invisible to the human eye, as illustrated in Figure 1.2 [21]. These types of images cause little to no damage to the object and as such do not lose any quality, while capturing information beyond the naked eye. The images on the left side of Figure 1.2 are stacked on top of each other for each wavelength, which creates so called 'hypercubes', where two dimensions represent the spatial information and the third represent the spectral curve [14].

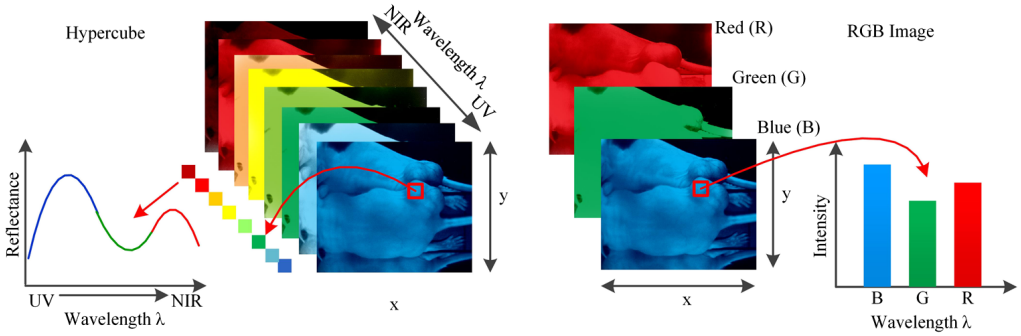


Figure 1.2: Spectrum generated from hyperspectral image (left) vs spectrum generated from RGB image (right). Each pixel of the image has a one-dimensional vector that comprises the spectral curve formed by the number of bands.

A number of successful results have been obtained from adopting HSI on agricultural applications and food quality applications [16] and has become a well-known technique in the industry [7]. In many of these applications quantitative information from HSI is extracted for prediction purposes, where regression is used most often for estimating such predictive information [10]. In this case the band of the spectral curves are taken as variables, which can number in hundreds and are likely to be high collinear according to Tobias et al. [22, p. 1]. Therefore a dimensionality reduction is required, in which only a selection of variables will be used during regression. These selected variables, which are called latent variables, captures most of the variance.

1.3. Research Question and Thesis Outline

The goal of this thesis is to utilize these latent variables of the hyperspectral signatures to accurately predict the growth of the potato plants. Prediction is achieved by means of a regression model with a certain predictive performance. By omitting some of the variables of the explanatory matrix, which decreases the dimension of the explanatory space, the predictive performance changes. The research question of this thesis is stated as follows: does variable selection on the hyperspectral data increase the predictive performance on the growth of potato plants?

This thesis is divided into eight chapters. Chapter 2 describes the Flight to Vitality data and gives an introduction to Least Squares. This chapter also explains why standard least squares methods work poorly the Flight to Vitality data. In such situations, only a few variables account for most of the variation [22], which are the latent variables. The are multiple approaches for finding such latent variables, which are discussed in chapter 3. One of these approaches is to bias them towards variables that accurately predicts the growth, while seeking directions in the explanatory space that captures most of the variance [22]. This is done by Partial Least Squares. The workings of this regression method is described in chapter 4. Moreover, this method has connections with other numerical methods, which is also explored in the same chapter. Chapter 5 discusses how PLS does prediction on the growth of the plants using the hyperspectral images. A good prediction relies heavily on the number of latent variables used, which is optimized by means of cross-validation. Variable selection is also included in this model. After this baseline model is constructed, a number of experiments are conducted. The description and the corresponding results can be found in chapter 6. Chapter 7 gives an analysis of the performance of the baseline model. Chapter 8 concludes the thesis and gives recommendations for future work.

2

FtV Data and an introduction to regression analysis

In this chapter, an outline of the Flight to Vitality (FtV) data is given, which consist of the hyperspectral images and the vitality dataset, that has captured the growth of the plants. Both datasets are grouped on different batches and then the average is taken on each batch. This results in a matrix X and a vector $\mathbf{y}$, where the matrix has more columns than rows. Two transformations are applied on the first matrix and one on the vector. Then Ordinary Least Squares is introduced and an explanation is given why this method works poorly on such data.

2.1. Flight to Vitality data

The provided FtV data is made up of two separate datasets. The first dataset consists of hypercubes after performing hyperspectral imaging on the potato tubers and the second dataset is the vitality data that has captured the growth of the potato plants.

2.1.1. Near-Infrared (NIR) hypercubes

An experiment has been conducted where hyperspectral images have been taken on multiple potato tubers from varying varieties. The produced hypercubes have spectral curves corresponding to wavelengths in the Near-Infrared (NIR) spectrum. The wavelength ranges from 939.64 nm to 2546.24 nm and is divided equally into 288 frequency bands.

Extracting the measured spectral curves from each individual hypercube can be challenging, since the size can get quite large at a pixel scale. This obstacle can be resolved by taking the mean spectrum within an area of hypercubes, instead of each pixel [10]. For this experiment, the mean spectra is taken on two distinctive types of tuber parts: the cortex and the pith, as illustrated in Figure 2.1. As already mentioned in the introduction, both areas might have a significant effect on the growth of the plant. The model is going to apply regression on the cortex, on the pith and on the mean of them both. The results can be found in section 5.3.

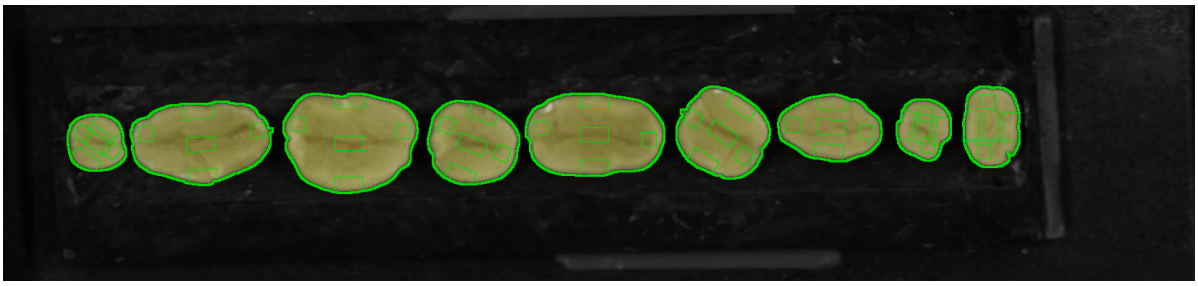


Figure 2.1: A group of potato tubers used for HSI, where both the pith and cortex are highlighted in green rectangles. For each tuber, the inner rectangle is the so-called pith and the average of the remaining four rectangles form the cortex. The mean tuber part is the average of all five rectangles.

The dataset itself has recorded multiple hypercubes for different batches of tubers. In total, there are 180 different batches, distributed evenly over six varieties of potatoes. By taking the average on each individual batch, the the newly acquired data has 180 rows, which comprises the mean spectral curve on each batch. Mathematically speaking, such tables can be interpreted as a 180×288 matrix X, with each row $i \in \{1, 2, \ldots, 180\}$ directly corresponding to the batch numbers. With this procedure, three matrices are obtained for each tuber part: $X_{\text{pith}}, X_{\text{cortex}}, X_{\text{mean}}$. Figure 2.2 displays the spectral curves of X_{mean}, where each variety is highlighted with a different colors.

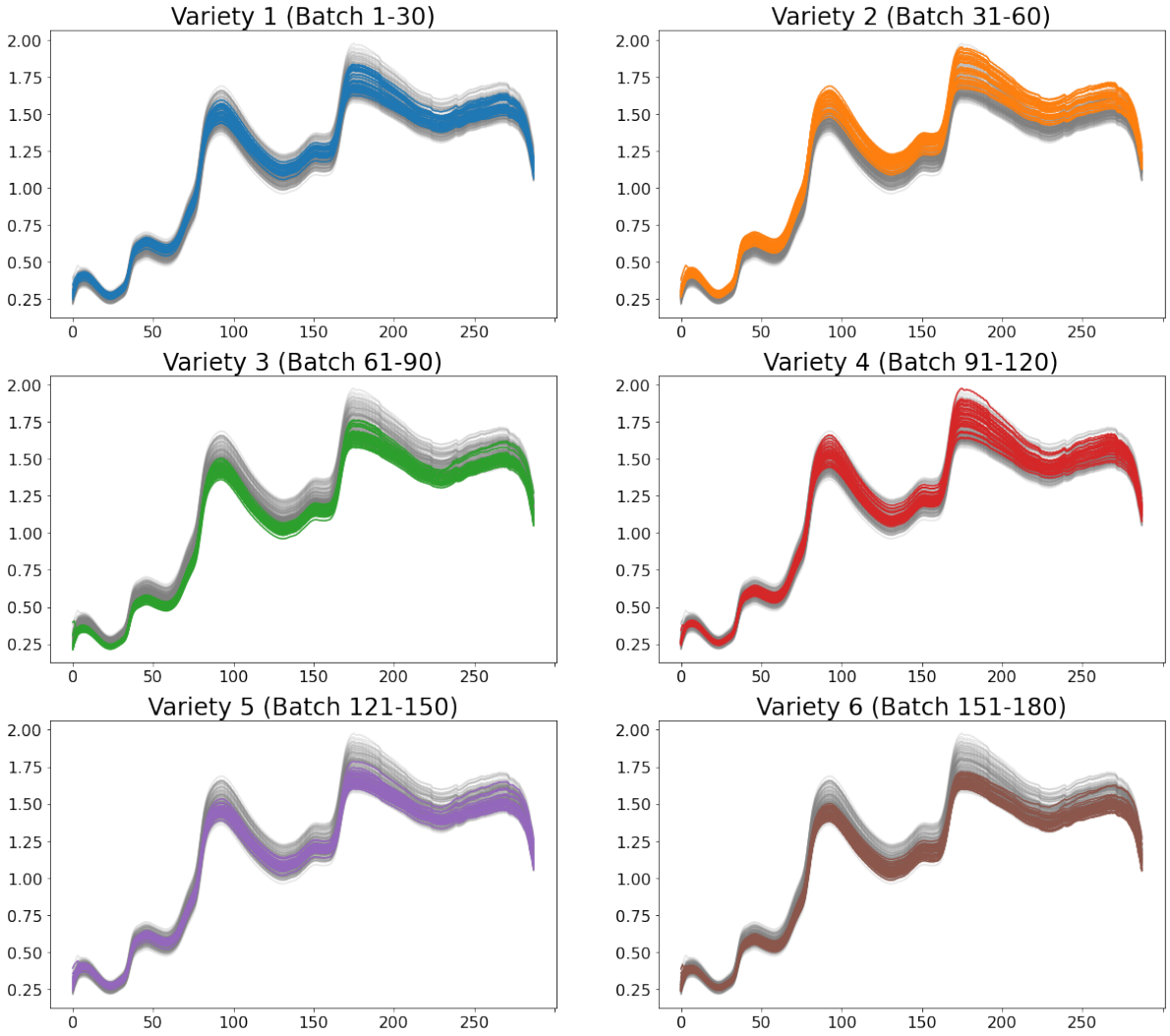


Figure 2.2: Spectral curve for varying varieties. Each curve corresponds to the mean spectral curve of one batch. All curves are taken from the mean tuber part. The horizontal axis shows all 288 frequency bands and the vertical axis shows absorbance/reflectance on each frequency band.

2.1.2. Vitality data

The Vitality data has captured the growth of multiple potato plants for different batches, which also ranges from 1 to 180. This experiment took place in three different locations, each differing in the type of soil and climate. Two of these potato fields are located in the Netherlands (SPNA and Veenklooster) and the third is located in France (Montfrin). The growth of each field has only been recorded for the first two weeks, with the starting day differing per field. Furthermore, the number of recorded days also differ in each field. To give an illustration of one growth, the growth of Montfrin is displayed in Figure 2.3.

By taking the average on the batches, just like with the NIR hypercubes, the growth can also be interpreted as a matrix Y, with 180 rows and q columns ($q = 5$ for SPNA and Veenklooster, $q = 6$ for Montfrin). Since q is not

fixed on all fields, comparing the growth per day with each other becomes quite challenging. Therefore, by taking the average over all recorded days, the overall growth per batch can be compared across the fields. This turns the matrix Y into a vector $\mathbf{y}$ for each field. That is, $\mathbf{y}_{\text{m}}, \mathbf{y}_{\text{s}}, \mathbf{y}_{\text{v}}$, for fields Montfrin, SPNA and Veenklooster respectively.

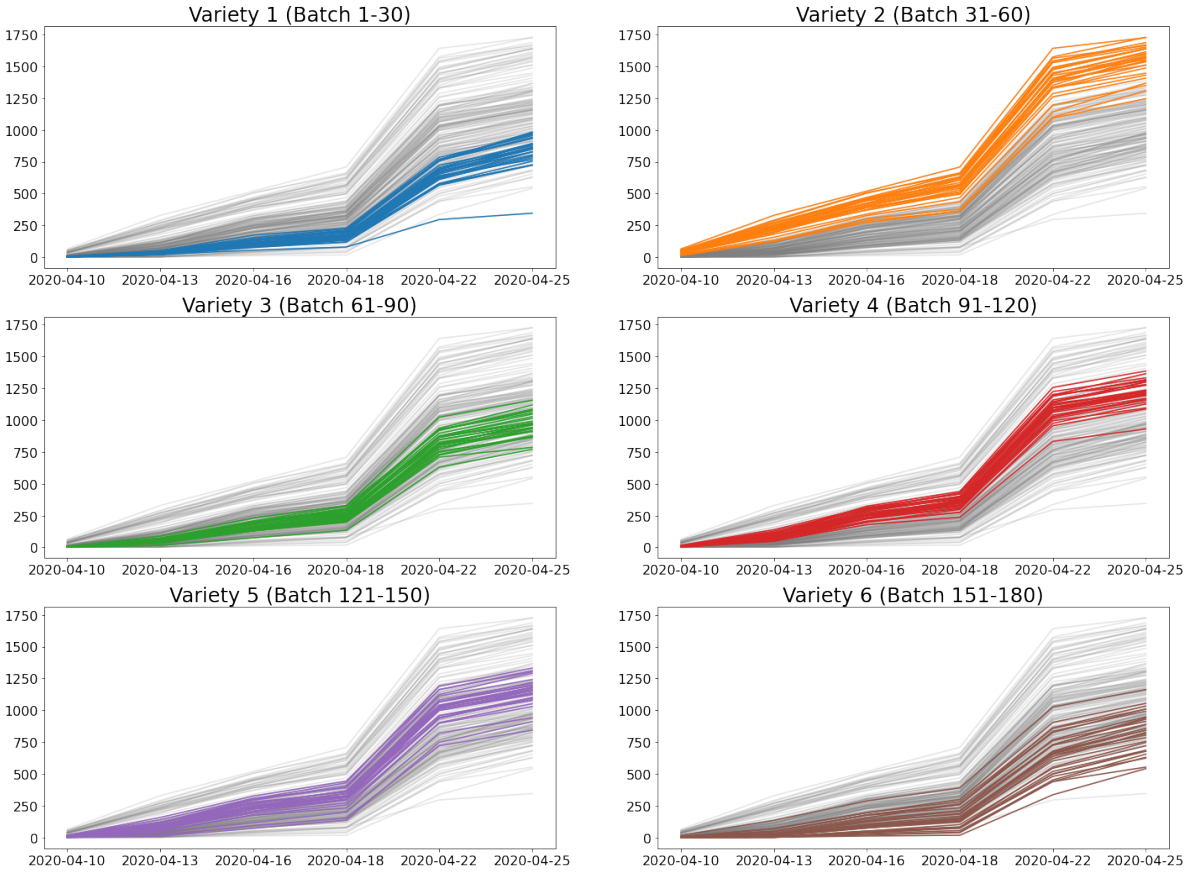


Figure 2.3: Growth of Montfrin for varying varieties. Each curve corresponds to the mean growth of each batch. The horizontal axis displays the first two weeks recorded in multiple days. The vertical axis shows the growth, measured in millimetres.

To summarize, the newly acquired matrix X and vector $\mathbf{y}$, along with their possible combinations are described in Table 2.1. With these three matrices and three vectors, the relation between them can be quantified already with regression. However, pre-processing both X and $\mathbf{y}$ are also taken into consideration.

Matrix $X \in \mathbb{R}^{n \times p}$	
X_{cortex}	Hyperspectral signatures taken from the cortex tuber part for each batch.
X_{pith}	Hyperspectral signatures taken from the pith tuber part for each batch.
X_{mean}	Hyperspectral signatures taken from the mean tuber part for each batch.
Vector $\mathbf{y} \in \mathbb{R}^{n}$	
$\mathbf{y}_m$	Average growth of Montfrin for each batch.
$\mathbf{y}_s$	Average growth of SPNA for each batch.
$\mathbf{y}_v$	Average growth of Veenklooster for each batch.

Table 2.1: Acquired data summarized. There are three different tuber parts and thus three different matrices X. There are three different fields and thus three different vectors $\mathbf{y}$. For each batch the mean spectral curve is taken.

2.2. Ordinary Least Squares and why it is not a viable method on FtV data

Conducting regression analysis on both the hyperspectral data and the vitality data helps with quantifying relations within both datasets. The objective of regression is to relate the response space with the explanatory space [4]. The most common form of regression, Ordinary Least Squares, is characterized with equation (2.1).

$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \cdots + \beta_p \mathbf{x}_p + \boldsymbol{\epsilon} = X\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.1}$$

$\mathbf{y}$ is a vector of length n, found from the observed vitality data on one field out of the three. $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ is the vector of regression coefficients, which assigns weights to the explanatory variables. X is an $n \times p$ matrix containing the values of these p explanatory variables $\mathbf{x}_j$, $j = 1, 2, \ldots, p$, which are the spectral curves taken from the NIR hypercubes. $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ is a vector of length n containing the errors. These errors are assumed to be uncorrelated, to be normal and to have the same variance, as mentioned by Wold et al. [24]. The regression coefficients can be calculated explicitly from the matrix $\mathbf{X}$ and vector $\mathbf{y}$. Furthermore, the estimated vector $\hat{\boldsymbol{y}}$ is given by (2.2), where the vector $\boldsymbol{\beta}_{\text{OLS}}$ minimizes the sum of squared errors $\left(\boldsymbol{y} - X\boldsymbol{\beta}_{\text{OLS}}\right)^2$. In addition, $(A)^+$ denotes the Moore-Penrose Pseudoinverse of matrix A.

$$\hat{\boldsymbol{y}} = X\boldsymbol{\beta}_{\text{OLS}}, \quad \boldsymbol{\beta}_{\text{OLS}} = \left(X^T X\right)^+ X^T \mathbf{y} \tag{2.2}$$

To describe how well a regression model fits the observed data, one can pick goodness-of-fit measureS, such as R^2 (2.3) and the Mean Squared Error (MSE) (2.4), where $\bar{\mathbf{y}}$ is the average of $\mathbf{y}$. With these measures the performance of a regression model can be quantified.

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{n} \left(\mathbf{y}_i - \hat{\mathbf{y}}_i\right)^2}{\sum_{i=1}^{n} \left(\mathbf{y}_i - \bar{\mathbf{y}}\right)^2} \tag{2.3}$$

$$\begin{aligned} \text{MSE}\left(\hat{\boldsymbol{\beta}}\right) &= E\left[\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)^T \left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right] \\ &= \left(E\left[\hat{\boldsymbol{\beta}}\right] - \boldsymbol{\beta}\right)^T \left(E\left[\hat{\boldsymbol{\beta}}\right] - \boldsymbol{\beta}\right) + E\left[\left(\hat{\boldsymbol{\beta}} - E\left[\hat{\boldsymbol{\beta}}\right]\right)^T \left(\hat{\boldsymbol{\beta}} - E\left[\hat{\boldsymbol{\beta}}\right]\right)\right] \end{aligned} \tag{2.4}$$

2.2.1. Data normalization on FtV data

There is no prior selection of important independent features available, thus scaling each feature to having zero mean and unit variance is a viable option before doing any prediction [24]. This preprocessing method is named standardization and can increase the performance of the model. By standardizing the spectral curves from Figure 2.2, Figure 2.4 is obtained. Just like with the NIR hypercubes, the vector $\mathbf{y}$, that describes the average growth, has to be standardized as well. This is because a variable with large variance can have more (negative) impact on the prediction than a variable that have a smaller variance [24].

Let $\tilde{\mathbf{y}} \in \mathbb{R}^n$ be the vector of raw vitality data on a given day and $\tilde{X} \in \mathbb{R}^{n \times p}$ – the matrix with the raw NIR spectral signatures of the batches as its rows. Mathematically, the zero-column-mean, unit–column-variance scaling of the regression problem

$$\tilde{\mathbf{y}} = \tilde{X}\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\epsilon}} \tag{2.5}$$

can be understood as the following algebraic operations:

$$\begin{aligned} \mathbf{y} &= X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \\ \mathbf{y} &= \left[I - n^{-1}\mathbf{1}\mathbf{1}^T\right]\tilde{\mathbf{y}}, \\ \boldsymbol{\epsilon} &= \left[I - n^{-1}\mathbf{1}\mathbf{1}^T\right]\tilde{\boldsymbol{\epsilon}}, \\ X &= \left[I - n^{-1}\mathbf{1}\mathbf{1}^T\right]\tilde{X}V^{-1}, \\ \boldsymbol{\beta} &= V\tilde{\boldsymbol{\beta}}, \end{aligned} \tag{2.6}$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector of all ones, and the diagonal matrix $V \in \mathbb{R}^{p \times p}$ contains the variances of each column of $\tilde{X}$. If, in addition, the vitality data are normalized to have the unit variance, then the whole problem is divided by the variance of the entries of $\mathbf{y}$ as well.

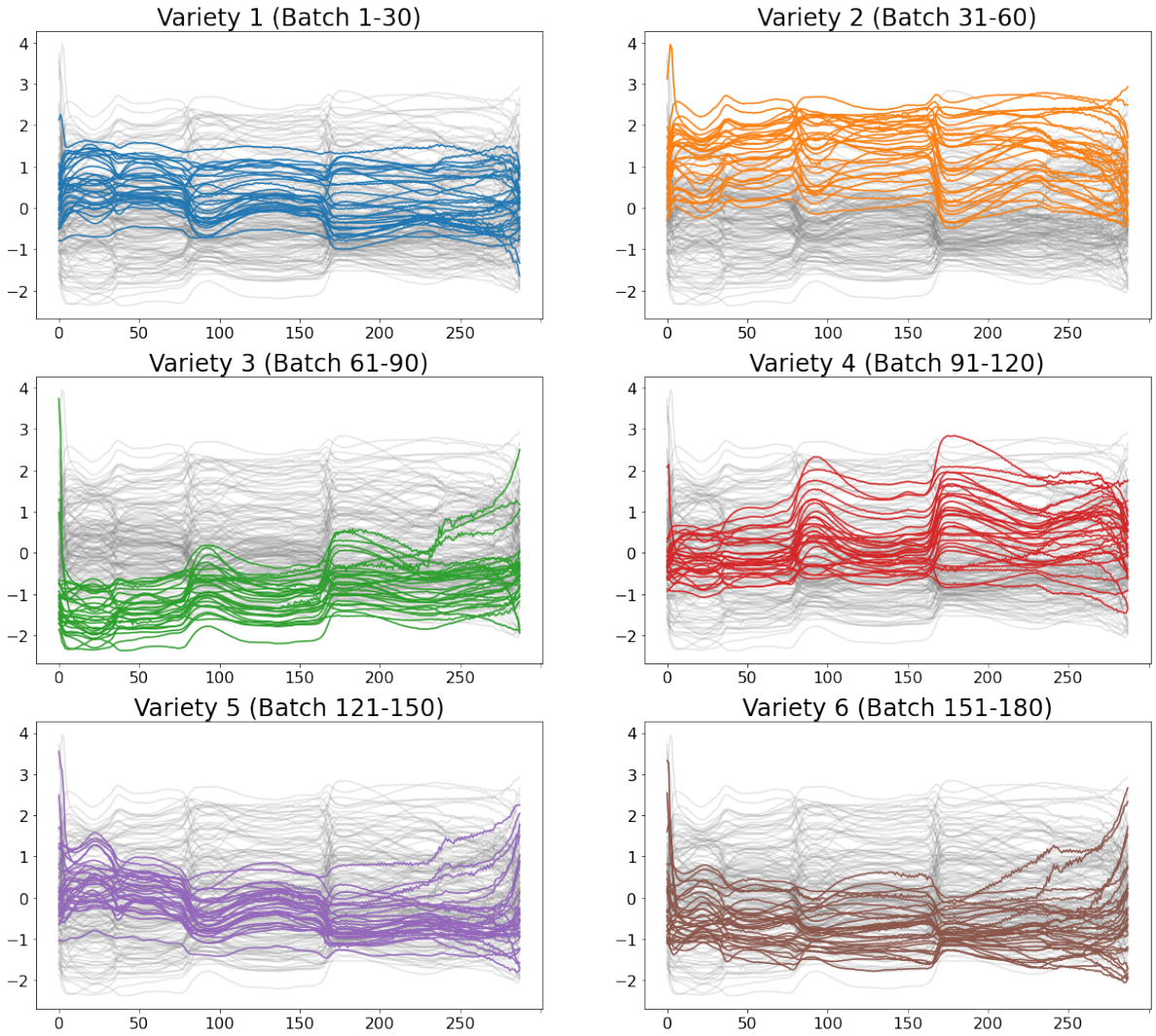


Figure 2.4: Spectral curve for varying varieties standardized. The curves are from the mean tuber part. All 288 variables have mean zero and are unit variant.

2.2.2. Savitzky-Golay filter on hyperspectral signatures

This pre-processing method is only adapted on the raw spectral signatures $\tilde{X}$. Savitzky-Golay filtering performs noise reduction on the matrix, while preserving the spectral curvature. In this filtering process a least-squares polynomial smoothing is performed on a subset of frequency bands centered about one frequency band. This results in a smoothing on every central frequency band, that moves point by point across the entire spectrum [12]. Typically, a low order polynomial is selected for approximation. In this case the order of this polynomial is set to two. The length of this subset, or rather the window, must be odd and is set to 19. This number is found after visually inspecting the curvature of different filtered signatures, with varying lengths of windows. After inspecting multiple window lengths, 19 seems to smooth the curves the best, while maintaining the curvature of the hyperspectral data.

Once the polynomial is constructed, the derivative of this polynomial can be calculated as well. This is because all (regression) coefficients (of the least-squares fit) are now known. By taking the first derivative of this polynomial, the linear slope is eliminated from the hyperspectral signatures. The filtered curve is displayed in Figure 2.5 for each variety.

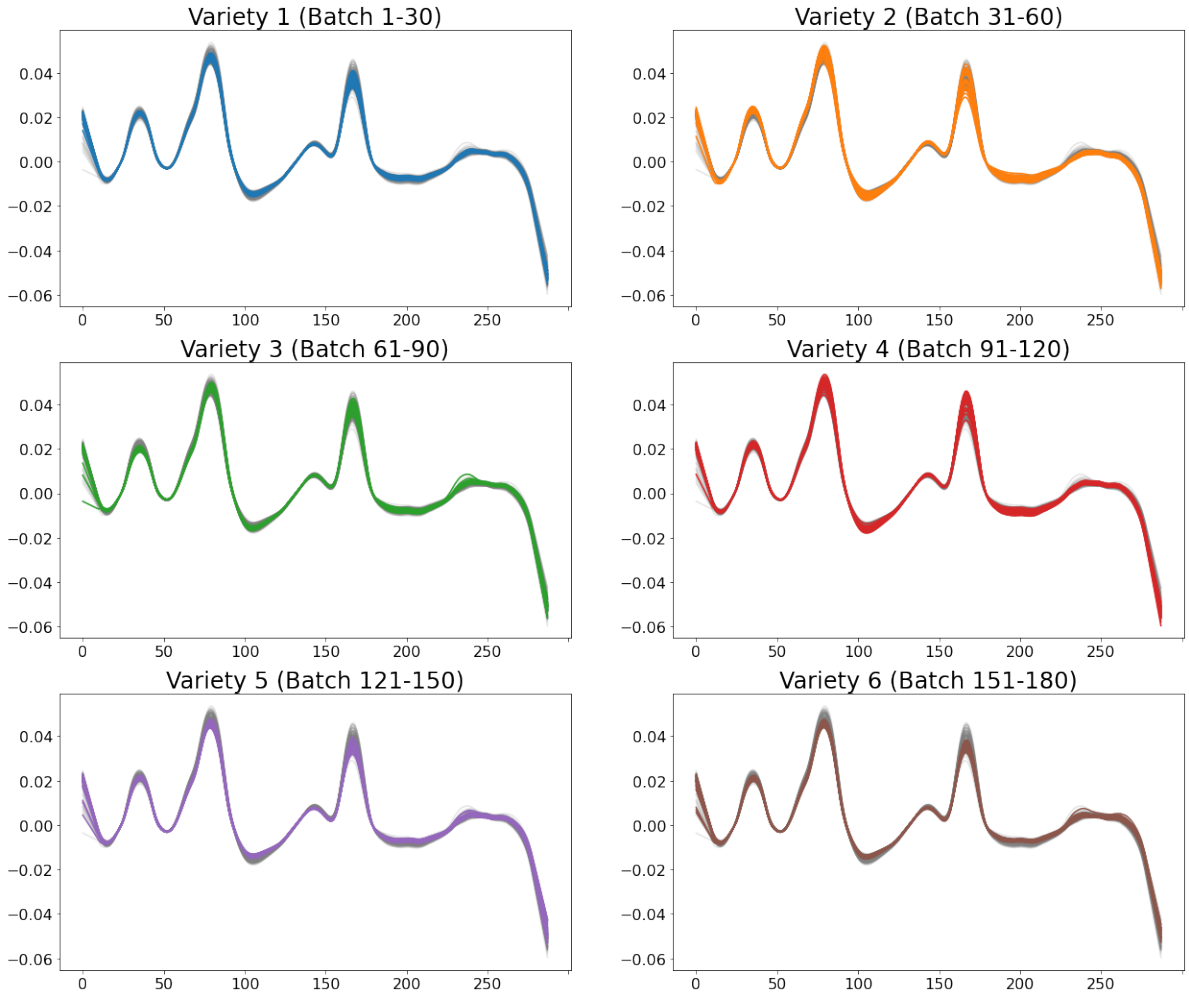


Figure 2.5: Spectral curves for varying varieties passed through the Savitzky-Golayfilter, with polynomial order 2 and window length 19. The curves are from the mean tuber part.

2.2.3. Overfitting FtV data

It is important to note that the number of variables in X are larger than the number of observations (288 and 180 respectively). Hence, it may, in principle, be possible to fit the vitality data $\mathbf{y}$ exactly, provided that $\text{rank}(X) = 180$. This 'overfitting', however, is undesirable, since the vitality data vector $\mathbf{y}$ contains significant statistical noise, as its entries are not the exact expectations, but rather the sample estimates of the mean batch vitality. Overfitting causes the regression vector $\hat{\boldsymbol{\beta}}$ estimated on some part of the data to be a very poor predictor of the remaining part of the data.

One way to resolve this issue of overfitting is to find a selection of variables $\mathbf{x}_j$ that "account for most variation in the response vector", as mentioned by Tobias et al. [22]. These selected variables, which are called latent variables, are not (always) directly observable from the spectral curves and must be extracted through some (mathematical) method. The main purpose with extracting latent variables is to increase the bias as little as possible, while capturing most variance. The newly selected latent variables, which span a new (reduced) predictor space, regresses on the response space. There are multiple methods for extracting and interpreting these latent variables, which are discussed in the next chapter.

3

Extracting Latent Variables from Hyperspectral Data

As shown in the previous chapter, collinearity makes the regression model perform bad and by extracting latent variables from the explanatory variables found in X, one can resolve the issue of overfitting. This chapter explains how latent variables can reduce the number of variables needed for regression and focuses on multiple methods of finding such variables. These methods are then compared with each other with evaluations metrics. The chapter finished by picking the most suitable method, which is Partial Least Squares (PLS).

3.1. Framework for finding latent variables in regression

For this chapter the singular value decomposition of X is extensively used, where $X = V\Sigma S^T$. Here V and S both are orthonormal matrices and Σ is a diagonal matrix containing the singular values of X in a decreasing order. Combining equation (2.2) and the singular value decomposition of $X^T X = S\Sigma^2 S^T$, the regression coefficients of this equation can be rewritten as

$$\boldsymbol{\beta}_{\text{OLS}} = \left(X^T X\right)^+ X^T \mathbf{y} = \left(S\Sigma^2 S^T\right)^+ \left(V\Sigma S^T\right)^T \mathbf{y} = S\left(\Sigma^2\right)^+ \Sigma V^T \mathbf{y} = \sum_{i=1}^{r} \frac{\boldsymbol{v}_i^T \boldsymbol{y}}{\sqrt{\lambda_i}} \boldsymbol{s}_i = \sum_{i=1}^{r} \hat{\boldsymbol{b}}_i, \tag{3.1}$$

where $r = \text{rank}(X^T X)$. With this equation, orthogonal vectors in the explanatory space X are selected. Figure 3.1 displays the singular values of $X^T X$ in logarithmic scale, where $X = X_{\text{mean}}$. In this figure only a small number of singular values appear to be large, compared to the rest, which indicates significant collinearity of the rows of X. The singular values starting from the 181^{th} component and are zero, since $r \leq 180$.

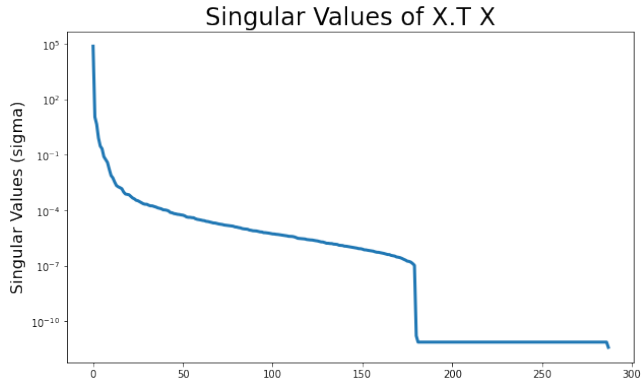


Figure 3.1: Singular values of $X^T X$ in decreasing order and the vertical axis is in logarithmic scale. Here the matrix X is taken from the mean part of the tubers.

Even though the OLS estimator can be unbiased under certain conditions, the variance depends on the singular values of $X^T X$, which gets high for small singular values. Furthermore, the corresponding component used in regression, that is associated with this singular value, has low sample spread. [20].

There are different methods for finding and interpreting the latent variables. In each method, different orthogonal vectors are computed that form a basis of the space spanned by the columns of X, according to Burnham et al. [4]. One way to approach this is by looking at the so called shrinking estimators, as described in equation (3.2). In general, a shrinkage estimator shrinks directions that yields high variance, where $f(\lambda_i)$ are the shrinkage factors. This shifts the regression coefficients away from the OLS coefficients and moves them towards directions that have larger sample spread. The variance of the i^{th} variable $\mathbf{x}_i$ decreases for $|f(\lambda_i)| < 1$ and increases for $|f(\lambda_i)| > 1$. However, the bias increases whenever a component is added to the shrinkage estimator.

$$\boldsymbol{\beta}_{\text{shr}} = \sum_{i=1}^{r} f(\lambda_i)\, \hat{\boldsymbol{b}}_i \tag{3.2}$$

Different shrinking factors $f(\lambda_i)$ give rise to different regression methods. Three of which are described in the upcoming sections, along with their performance on the observations of the FtV data.

3.2. Principal Component Regression

Principal Component Regression (PCR) looks at the first $k \leq r$ elements of both singular vectors $v_1, \ldots, v_k$ and $s_1, \ldots, s_k$, which are the first k principal components. The variable k indicates how much bias is added to the model. Since the singular values are sorted in a decreasing order, increasing r reduces the bias[8]. If $k = r$, then the resulting regression vector $\boldsymbol{\beta}_{\text{PCR}}$ has no bias and therefore $\boldsymbol{\beta}_{\text{PCR}} = \boldsymbol{\beta}_{\text{OLS}}$. The regression coefficients of PCR, which is a shrinkage estimator of OLS, is described with equation (3.3).

$$\boldsymbol{\beta}_{\text{PCR}} = \sum_{i=1}^{r} f(\lambda_i)\, \hat{\boldsymbol{b}}_i = \sum_{i=1}^{k} \hat{\boldsymbol{b}}_i, \quad f(\lambda_i) = \begin{cases} 1 & i \leq k \\ 0 & i > k \end{cases} \tag{3.3}$$

The objective is find the value of the parameter k for which the mean squared error MSE is minimized. Some method of model selection is required for finding this k and for reporting its performance with this parameter. The full description of this method is explained in chapter 5, but the performance of PCR is already shown in Table 3.1. Equations (2.3), (2.4) and (3.3) were used for deriving the estimated MSE and R^2 values, where the matrix $X = X_{\text{mean}}$ was separated into two sets.

Field name	MSE	R^2
Montfrin	0.274221	0.684000
SPNA	0.223153	0.730065
Veenklooster	0.355838	0.561824
Average	**0.284404**	**0.658630**

Table 3.1: Performance of PCR for each field on the mean tuber part. The matrix is standardized before doing regression. In this table $k = 39$.

3.3. Ridge Regression

Ridge Regression (RR) slightly adjusts the regression coefficients of (3.1) by adding a small number $\alpha \in (0, 1)$ to the diagonal elements of the matrix $X^T X$. This ridge parameter α controls of the degree of stabilization. Setting $\alpha = 0$ results in an unbiased estimator, that is $\boldsymbol{\beta}_{\text{RR}} = \boldsymbol{\beta}_{\text{OLS}}$, and $\alpha > 0$ increases the bias [8]. Unlike in PCR, every singular value is taken into consideration in RR. Finding the ridge parameter for which the MSE value is minimized is a much harder task, since the range $(0, 1)$ is continuous. Therefore a grid is taken on this range, consisting of a finite number of evenly spaced numbers. The same method of model selection is used and the performance of RR is put in Table 3.2.

$$\boldsymbol{\beta}_{\text{RR}} = \left(X^T X + \alpha \cdot I\right)^{+} X^T \mathbf{y} = \sum_{i=1}^{r} f(\lambda_i)\, \hat{\boldsymbol{b}}_i = \sum_{i=1}^{r} \frac{\lambda_i}{\lambda_i + \alpha} \hat{\boldsymbol{b}}_i \tag{3.4}$$

Field name	MSE	R^2
Montfrin	0.295579	0.659389
SPNA	0.229668	0.722185
Veenklooster	0.375760	0.537293
Average	**0.300336**	**0.639622**

Table 3.2: Performance of PLS for each field on the mean tuber part. The matrix is standardized before doing regression. In this table $\alpha = 0.02$.

3.4. Partial Least Squares

Partial Least Squares (PLS) has similarities to PCR, but the principal components in PCR are determined only from the explanatory matrix X, whereas with PLS, the components are taken from both X and $\mathbf{y}$. The components derived from PLS seeks directions in the explanatory space that has low sample spread, and biases them towards directions that provides accurate predictions [22]. PLS obtains approximated eigenvalues μ_j and eigenvectors $\mathbf{r}_j$ from the Krylov subspace of $X^T X$ and $X^T \mathbf{y}$, which are used for computing the regression estimate (3.5).

$$\boldsymbol{\beta}_{\text{PLS}} = \sum_{i=1}^{r} f(\lambda_i)\, \hat{\boldsymbol{b}}_i = \sum_{i=1}^{k} \left[1 - \prod_{j=1}^{k} \left(1 - \frac{\lambda_i}{\mu_j} \right) \right] \hat{\boldsymbol{b}}_i \tag{3.5}$$

The procedure for finding the optimal number of components is similar as the procedure in PCR. If $k = r$, then $\boldsymbol{\beta}_{\text{PLS}} = \boldsymbol{\beta}_{\text{OLS}}$. The results are found in Table 3.3.

Field name	MSE	R^2
Montfrin	0.261075	0.699149
SPNA	0.209283	0.746843
Veenklooster	0.383465	0.527804
Average	**0.284608**	**0.657932**

Table 3.3: Performance of RR for each field on the mean tuber part. The matrix is standardized before doing regression. In this table $k = 15$.

4

Partial Least Squares and its connection Krylov Subspace methods

This chapter describes the workings of Partial Least Squares. The general idea of PLS is that latent vectors are obtained by maximising the covariance between two sets, by decomposing them into scores and loadings matrices [20]. The computation of these matrices is done with the algorithm **N**onlinear **I**terative **Pa**rtial **L**east **S**quares (NIPALS). Moreover, Partial Least Squares is identical with the Conjugate Gradient method for solving normal equations [18]. This equivalence is explored in this chapter, and finally an algorithm for solving normal equations efficiently is presented.

4.1. Partial Least Squares Regression

The general description of PLS is given in this section, where Y is assumed to be the $(n \times q)$ response matrix. For the actual implementation of this method, which is discussed in chapter 5, q is set to one. In PLS the covariance between two sets, X and Y, are maximized by decomposing the matrices X, Y into scores and loadings matrices. The decomposition of both matrices X and Y is given by equation (4.1)

$$\begin{cases} X = TP^T + E, & T = (\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k) \\ Y = UQ^T + F, & U = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k) \end{cases} \tag{4.1}$$

Here T, U are $(n \times k)$ matrices containing k score vectors, which are in fact the desired latent vectors. Matrices P, Q, with dimensions $(p \times k)$ and $(q \times k)$ respectively, are the matrices of loadings. Lastly, the $(n \times p)$ matrix E and the $(n \times q)$ matrix F represent the residuals.

Nonlinear Iterative Partial Least Squares (NIPALS) is an algorithm that computes vectors $\mathbf{r}_i, \mathbf{s}_i, i \in \{1, 2, \ldots, k\}$ for which the covariance between vectors $\mathbf{t} = X\mathbf{r}, \mathbf{u} = Y\mathbf{s}$ are maximized. In each iteration i, the vector that attains this maximum covariance are the weight vectors $\mathbf{w}, \mathbf{c}$. The corresponding optimization task is described in equation (4.2).

$$\max_{|\mathbf{r}|=|\mathbf{s}|=1} [cov(X\mathbf{r}, Y\mathbf{s})]^2 = [cov(X\mathbf{w}, Y\mathbf{c})]^2 = [cov(\mathbf{t}, \mathbf{u})]^2 \tag{4.2}$$

According to Rosipal and Krämer, it can be shown that the weight vector $\mathbf{w}$ correspond to the first eigenvector of the following eigenvalue problem [20].

$$X^T Y Y^T X\mathbf{w} = \lambda \mathbf{w} \tag{4.3}$$

Once this vector $\mathbf{w}$ is found, the following steps are taken for computing $\mathbf{u}$:

$$\mathbf{c} = \frac{Y^T\mathbf{t}}{\mathbf{t}^T\mathbf{t}}, \quad ||\mathbf{c}|| \to 1, \quad \mathbf{u} = Y\mathbf{c}. \tag{4.4}$$

For $q = 1$, $\mathbf{u} = \mathbf{y}$ (univariate regression) [20]. After calculating the vectors $\mathbf{t}, \mathbf{u}$, both matrices X, Y are deflated using the vectors of loadings $\mathbf{p}, \mathbf{q}$. These vectors $\mathbf{p}, \mathbf{q}$ are obtained by regressing X on $\mathbf{t}$ and Y on $\mathbf{u}$ respectively,

$$\mathbf{p} = \frac{X^T\mathbf{t}}{\mathbf{t}^T\mathbf{t}}, \quad \mathbf{q} = \frac{Y^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}} \tag{4.5}$$

However equations (4.4) and (4.5) assume that the relation between X and Y is symmetric, which is rarely the case when doing regression. For this reason, two assumptions must be made for the asymmetric relation between X and Y [20]. The first assumption is that the score vectors $\{\mathbf{t}_i\}_i^k$ must provide good predictions on Y and are mutual orthogonal. The second assumption states that there exist a relation between the scores vectors themselves, that is $U = TD + H$, where D is a $(k \times k)$ diagonal matrix and H is the matrix of residuals. With these two assumptions the score vectors $\{\mathbf{t}_i\}_i^k$ are used for deflating both matrices X and Y.

$$\begin{cases} X' = X - \mathbf{t}\mathbf{p}^T \\ Y' = Y - \frac{\mathbf{t}\mathbf{t}^T Y}{\mathbf{t}^T\mathbf{t}} = Y - \mathbf{t}\mathbf{c}^T \end{cases} \tag{4.6}$$

After this deflation step, the NIPALS algorithm moves to the next iteration and the process of extracting scores vectors $\mathbf{t}$ is repeated with the new deflated matrices X' and Y', as described in equation (4.6).

By combining equation (4.1) and $U = TD + H$, the values of matrix Y can be expressed in terms of the values of X, or rather, in terms of the scores matrix T derived from X.

$$Y = UQ^T + F = (TD + H)\,Q^T + F = TDQ^T + (HQ^T + F) = TC^T + F^* \tag{4.7}$$

In equation (4.7), OLS is performed on Y with T, where C^T is the matrix of regression coefficients and F^* the residuals matrix. To express equation 4.7 in terms of the explanatory matrix X, the relation $T = XW\left(P^TW\right)^{-1}$ is used [20], where P is the matrix of loadings of X.

$$Y = XB + F^*, \quad B = W(P^TW)^{-1}C^T = X^TU(T^TXX^TU)^{-1}T^TY \tag{4.8}$$

Figure 4.1 summarizes the entire process in an image, in which equations (4.1), (4.7), (4.8) and $U = TD + H$ are used. The same procedure can be used for the univariate problem ($q = 1$).

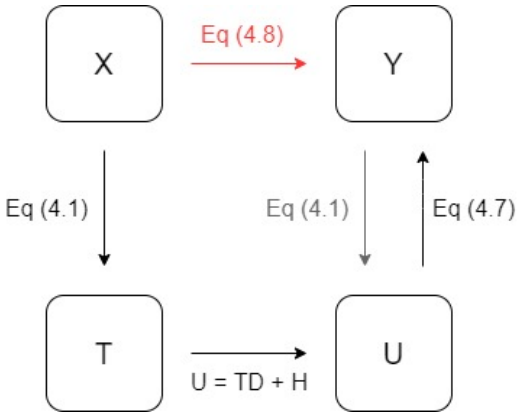


Figure 4.1: Illustration of Partial Least Squares Regression on explanatory matrix X and response matrix Y. Matrices T, U are the scores matrices of X, Y respectively.

4.2. Connections with Krylov Subspace methods

By using the same settings for OLS as in chapter 3, where the Singular Value Decomposition of X^TX was extensively used, then solving normal equations $\boldsymbol{\beta}$ in $\operatorname{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - X\boldsymbol{\beta}\|^2$ is equivalent with solving $\boldsymbol{\beta}$ for which $\left(X^TX\right)\boldsymbol{\beta} = X^T\mathbf{y}$. When writing a similar formulation for PLS, one can show that the PLS estimator at step k minimizes the least squares over a particular subspace that has dimension k [2]. The mathematical formulation of this notion is given by (4.9)

$$\boldsymbol{\beta}_{\text{PLS}} = \underset{\boldsymbol{\beta}\in\mathcal{K}^k\left(X^TX, X^T\mathbf{y}\right)}{\operatorname{argmin}} \|\mathbf{y} - X\boldsymbol{\beta}\|^2, \quad \mathcal{K}^k\left(X^TX, X^T\mathbf{y}\right) = \left\{X^T\mathbf{y}, \left(X^TX\right)X^T\mathbf{y}, \ldots, \left(X^TX\right)^{k-1}X^T\mathbf{y}\right\} \tag{4.9}$$

The subspace $\mathcal{K}^k$ is known as the k^{th} Krylov subspace. With equation (4.9) Phatak and de Hoog have shown that multiple Krylov Subspace methods yield iterates that are identical to the PLS estimator of the corresponding dimensionality [18]. These methods are explored in this section.

4.2.1. Conjugate Gradient method

Conjugate Gradient (CG) method solves a system of equations $A\mathbf{x} = \mathbf{b}$ iteratively by minimizing the quadratic equation, where A is positive semidefinite [18] [20].

$$\frac{1}{2}\mathbf{x}^TA\mathbf{x} - \mathbf{b}^T\mathbf{x} \tag{4.10}$$

Definition 4.2.1. Let A be a symmetric matrix. Vectors $\mathbf{d}_i \in \mathbb{R}^n (\mathbf{d}_i \neq 0)$ $i \in \{1, 2, \ldots, k\}$ are conjugate with respect to A, if $\mathbf{d}_i^TA\mathbf{d}_j = 0, \forall i \neq j$. These vectors $\{\mathbf{d}_i\}_{i=1}^k$ are called conjugate directions.

Lemma 4.2.1. Let A be positive definite. If vectors $\mathbf{d}_i, i \in \{1, 2, \ldots, k\}$ are conjugate with respect to A, then these vectors are linearly independent

For PLS regression set $A = X^TX$ and $\mathbf{b} = X^T\mathbf{y}$, then $A \in \mathbb{R}^{p\times p}$ is symmetric and positive definite. Let $\mathbf{x}^* = \sum_{i=0}^{p-1}\alpha_i\mathbf{d}_i$ be the exact solution of (4.10), where vectors $\left\{\mathbf{d}_0.\mathbf{d}_1, \ldots, \mathbf{d}_{p-1}\right\}$ are conjugate with respect to A. Using Lemma 4.2.1, vectors $\{\mathbf{d}_i\}_{i=0}^{p-1}$ are independent and thus

$$\mathbf{d}_i^TA\mathbf{x}^* = \alpha_i\mathbf{d}_i^TA\mathbf{d}_i \quad \Longleftrightarrow \quad \alpha_i = \frac{\mathbf{d}_i^TA\mathbf{x}^*}{\mathbf{d}_i^TA\mathbf{d}_i} = \frac{\mathbf{d}_i^T\mathbf{b}}{\mathbf{d}_i^TA\mathbf{d}_i} \tag{4.11}$$

With equation (4.11) the solution of equation (4.10) is found by only calculating inner products. Furthermore, the scalars α_i are derived explicitly from A and $\mathbf{b}$. Thus, the solution $\mathbf{x}^*$ can be computed as follows

$$\mathbf{x}^* = \sum_{i=0}^{p-1}\alpha_i\mathbf{d}_i = \sum_{i=0}^{p-1}\frac{\mathbf{d}_i^T\mathbf{b}}{\mathbf{d}_i^TA\mathbf{d}_i}\mathbf{d}_i. \tag{4.12}$$

The iterative process of constructing this vector $\mathbf{x}^*$ is achieved by initialing an arbitrary vector $\mathbf{x}_0 \in \mathbb{R}^n$ and then taking the following steps for $\mathbf{x}_i, i = 1, 2, \ldots, p-1$ described in (4.13). Then after p steps, $\mathbf{x}_i = \mathbf{x}^*$.

$$\begin{aligned}
\mathbf{d}_0 &= -\mathbf{g}_0 = \mathbf{b} - A\mathbf{x}_0 \\
\alpha_i &= -\frac{\mathbf{g}_i^T\mathbf{d}_i}{\mathbf{d}_i^TA\mathbf{d}_k} \\
\mathbf{g}_i &= A\mathbf{x}_i - \mathbf{b} \\
\gamma_i &= \frac{\mathbf{g}_{i+1}^TA\mathbf{d}_i}{\mathbf{d}_i^TA\mathbf{d}_i} \\
\mathbf{d}_{i+1} &= -\mathbf{g}_{i+1} + \gamma_i\mathbf{d}_i \\
\mathbf{x}_{i+1} &= \mathbf{x}_i + \alpha_i\mathbf{d}_i
\end{aligned} \tag{4.13}$$

The Conjugate Gradient algorithm is given by (4.13). The equivalence between CG and PLS is attained in each iteration i of both methods, since the solution $\mathbf{x}_i$ and the regression vector $\boldsymbol{\beta}_{\text{PLS}}$ at iteration i are identical [18]. Furthermore, the conjugate directions span the same Krylov subspace as in PLS, that is, for $j < p$, $\operatorname{span}\left(\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{j-1}\right) = \mathcal{K}^j\left(A, \mathbf{b}\right) = \mathcal{K}^j\left(X^TX, X^T\mathbf{y}\right)$.

4.2.2. Golub-Kahan Lanczos bidiagonalization

The nonsymmetric matrix $X \in \mathbb{R}^{n\times p}$, consisting of 180 hypercubes, can be reduced to a bidiagonal matrix by following the Golub-Kahan bidiagonalization process [9]. This bidiagonalization method is needed for the LSQR algorithm that solves the least-squares problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\| X\boldsymbol{\beta} - \mathbf{y} \right\|. \tag{4.14}$$

Given initial vector $\mathbf{y}$, the bidiagonalization of X is described in equation (4.15), where $\alpha_i, \gamma i \geq 0$ are picked such that $\|\mathbf{u}_i\| = \|\mathbf{v}_i\| = 1$.

$$\begin{gathered} \gamma_1 \mathbf{u}_1 = \mathbf{y}, \quad \alpha_1 \mathbf{v}_1 = X^T \mathbf{u}_1 \\ \left.\begin{aligned} \gamma_{i+1}\mathbf{u}_{i+1} &= X\mathbf{v}_i - \alpha_i \mathbf{u}_i \\ \alpha_{i+1}\mathbf{v}_{i+1} &= X^T\mathbf{u}_{i+1} - \gamma_{i+1}\mathbf{v}_i \end{aligned}\right\}, \quad i = 1,2,\ldots \end{gathered} \tag{4.15}$$

(4.15) can be rewritten as

$$\begin{aligned} U_{k+1}\left(\gamma_1 \mathbf{e}_1\right) &= \mathbf{y} \\ XV_k &= U_{k+1}C_k \\ X^{\mathrm{T}}U_{k+1} &= V_k C_k^{\mathrm{T}} + \alpha_{k+1}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^{\mathrm{T}} \end{aligned}, \tag{4.16}$$

where

$$U_k \equiv [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k], \quad V_k \equiv [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k], \quad C_k \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k} \tag{4.17}$$

The columns of V_k span the Krylov subspace $\mathscr{K}^k\left(X^TX, X^T\mathbf{y}\right)$ [9].

4.2.3. LSQR Algorithm

LSQR is based on the bidiagonalization process, where a sequence of approximations $\left\{\boldsymbol{\beta}_i : \boldsymbol{\beta}_i = V_i\mathbf{b}_i\right\}_{i=1}^k$ are generated in iteration i such that its residual norm $\|\mathbf{r}_i\|^2, \mathbf{r}_i = \mathbf{y} - X\boldsymbol{\beta}_i$, decreases monotonically in the next iteration $i+1$. Furthermore, this sequence of approximations are identical to the approximations found in (4.13). Nonetheless, LSQR is shown to be more reliable, according to Paige and Saunders [17].

$$\min_{\boldsymbol{\beta}\in\mathscr{K}^k(X^TX, X^T\mathbf{y})} \left\| X\boldsymbol{\beta} - \mathbf{y} \right\| = \min_{\mathbf{b}\in\mathbb{R}^k} \|C_k\mathbf{b} - \mathbf{e}_1\|\mathbf{b}\|\| \tag{4.18}$$

The goal of LSQR is to minimize equation (4.18), which is rewritten on the right-hand side. Denote the i^{th} solution of the right-hand side as $\mathbf{b}_i^*$, then the corresponding solution of (4.14) at iteration i is $\boldsymbol{\beta}_i = V_i\mathbf{b}_i$ [17].

5

PLS Regression on FtV Data

In this chapter Partial Least Squares (NIPALS) is used as a regressor for predicting the growth of the potato plants. The performance of this model relies heavily on the number of latent variables used, which is why the optimal number of latent variables needs to be found. A good way to approach this task is to split the data into training and testing sets and by performing cross-validation on the training set, where we minimize on the mean squared error of the prediction. Then with this optimal number, PLS produces regression coefficients that assigns a weight on each frequency band. Variable selection is then considered by omitting explanatory variables $\mathbf{x}_j$ in two different ways. One is done by iteratively omitting variables and the other omits variables if it lies outside a certain threshold. The performance of every steps described, in terms of MSE value and R2 value, is put in multiple tables and is found in the third section of this chapter. Finally, this chapter explores whether normality is present in the residuals.

5.1. Optimize number of latent variables

The predictive information of PLS, which is expressed with goodness-of-fit measures R^2 (2.3) and MSE (2.4), relies on the number of latent variables used. The procedure cross-validation is used, since it is a common technique for finding the optimal number of latent vectors [1] [24]. This optimization problem is also known as hyperparameter optimization or hyperparameter tuning. In this procedure, the 180 rows of matrix X are (randomly) divided into three sets: The training set, the testing set and the validation set. The exact split is also done for the response vector $\mathbf{y}$. An illustration of this procedure is given in Figure 5.1. In each split the model learns from the a portion of the training set (blue) and reports the estimated MSE value from the validation set (red), where the number of latent variables used varies from 1 to 40. The number of latent variables for which the MSE value is minimized is then the optimal number found. With this optimal number, the model now trains on the entire training set (purple) and assesses its performance on the testing set (green). The corresponding estimated MSE value and R2 value are reported in the final section of this report.

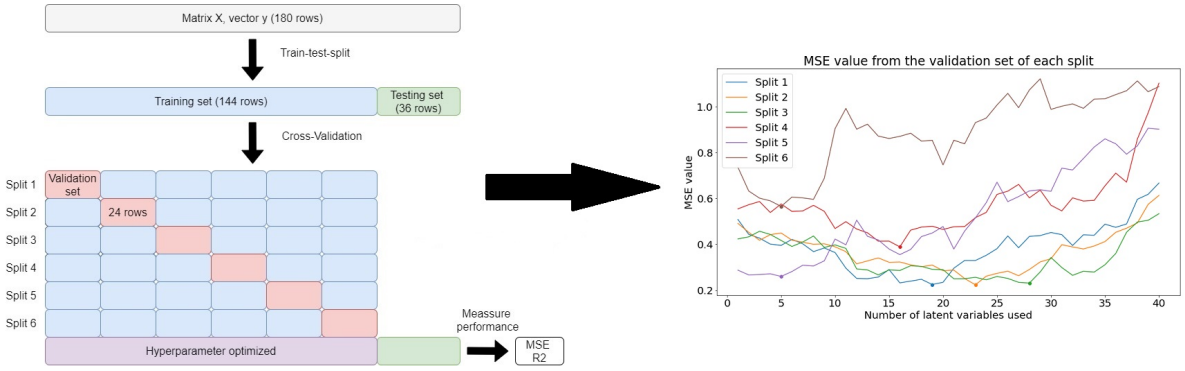


Figure 5.1: Splitting of matrix X and $\mathbf{y}$ visualized (left) and the estimated MSE value for multiple numbers of latent variables plotted for each split (right).

Figure 5.1 shows an example of k-Fold Cross Validation, with k (in case equal to 6) referring to the number of groups that a given data sample is to be split into. The value of k must be picked correctly. Otherwise it may result in a model where the performance has a high variance ($k = 1$) or where the performance has a high bias ($k = 180$). Furthermore for matrix X and vector $\mathbf{y}$, the variety of the tubers are not represented fairly in all three divided sets when applying 6-Fold CV. This issue is not resolved by changing the value of k, without introducing high variance or high bias. An alternative would be to divide the data into three sets, such that the ratio of each variety stays the approximately same. This is known as stratification and is illustrated in the second plot of Figure 5.2. If $k = 6, 12, 18, \ldots$, then the ratio stays exactly the same.

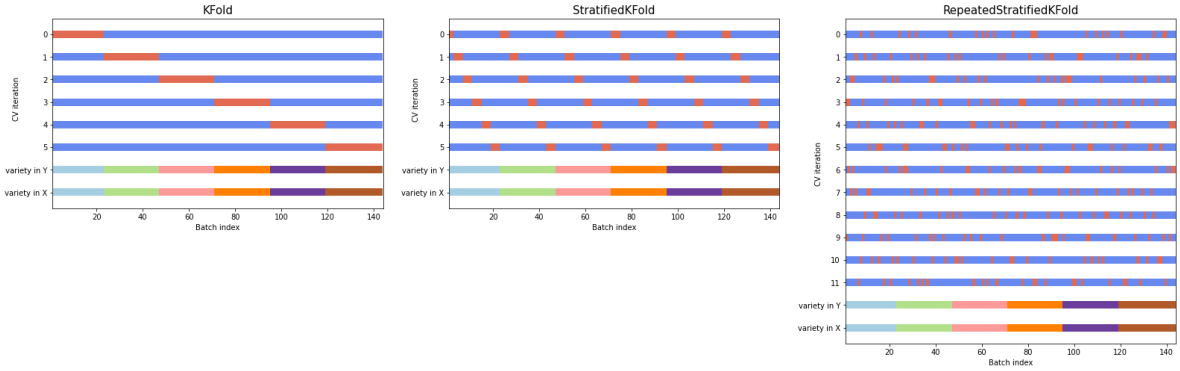


Figure 5.2: Different types of splitting on the training set and the validation set. On the left is 6-Fold CV. In the middle is also 6-Fold CV, but stratification is applied. On the right repeated stratified 6-Fold CV, where the procedure Stratified k-Fold CV is repeated twice.

Both k-Fold CV and Stratified k-Fold CV do not allow for repetitions in their splits. Repeating the splits in different ways (while stratified), improves the mean performance of the model. An example of this kind of cross-validation is shown in the third plot of Figure 5.2.

Following the same procedure as illustrated in the right plot of Figure 5.2, each split returns the number of latent variables used for which the MSE value is minimized, where the MSE value is estimated from the validation set. For each tuber part and each fieldname, the three most occurring optimal number of latent variables are listed in Tables 5.1 and 5.2, where repeated-stratified-12-Fold Cross-Validation is used on the normalized matrix and the matrix passed through the Savitzky-Golayfilter. The 12-Fold is repeated 8 times, which means that 96 splits are generated during hyperparameter optimization.

	Montfrin	SPNA	Veenklooster
Cortex	14 (10), 22 (9), 7 (6)	10 (11), 18 (9), 15 (8)	1 (11), 6 (10), 8 (10)
Pith	16 (9), 15 (8), 11 (6)	18 (13), 15 (11), 19 (11)	16 (11), 12 (10), 7 (10)
Mean	14 (9), 12 (8), 15 (7)	14 (11), 15 (10), 13 (9)	11 (14), 15 (11), 14 (8)

Table 5.1: Standardized matrix X. Each cell lists three most common opt. number of latent variables with the amount of it occurring in brackets.

	Montfrin	SPNA	Veenklooster
Cortex	1 (10), 28 (7), 17 (7)	16 (9), 9 (8), 26 (6)	13 (10), 1 (7), 12 (6)
Pith	22 (7), 19 (7), 17 (6)	18 (11), 23 (10), 17 (8)	13 (10), 8 (8), 16 (8)
Mean	1 (9), 23 (6), 21 (5)	16 (7), 15 (6), 29 (6)	12 (9), 30 (7), 8 (7)

Table 5.2: SG-filtered X. Each cell lists three most common opt. number of latent variables with the amount of it occurring in brackets.

When counting every number up, 15 seems to appear most often in Table 5.1 (55 times out of 96) and 1 appears the most in Table 5.2 (26 times out of 96). The second most occurring number in this table is 16 (24 out of 96). Even though 1 occurs more often than 16, it seems likely that 16 will produces more consistent results. Thus 16 is picked over 1. By taking these two numbers, 15 and 16, the hyperparameters are now optimized, which means that the performance can be evaluated with the testing set.

5.2. Variable selection

Each regression vector, from the nine possible PLS models, assigns weights to all 288 variables of the explanatory matrix X. Some of these variables can yield a large variation on the testing set and can thus worsen the performance in terms of predictive information [15]. Therefore two different approaches are described in this section, where the variables are filtered. The first method is an iterative process that filters variables based on the regression coefficients themselves and the second method filters variables based on a threshold of a measure, called variable importance in projection (VIP).

5.2.1. Regression coefficients

This method is one of the most straightforward methods, since the selection is based on the estimated coefficients itself. The variables are first sorted by the absolute value of their weights, as shown in Figure 5.3, and then iteratively filtered out.

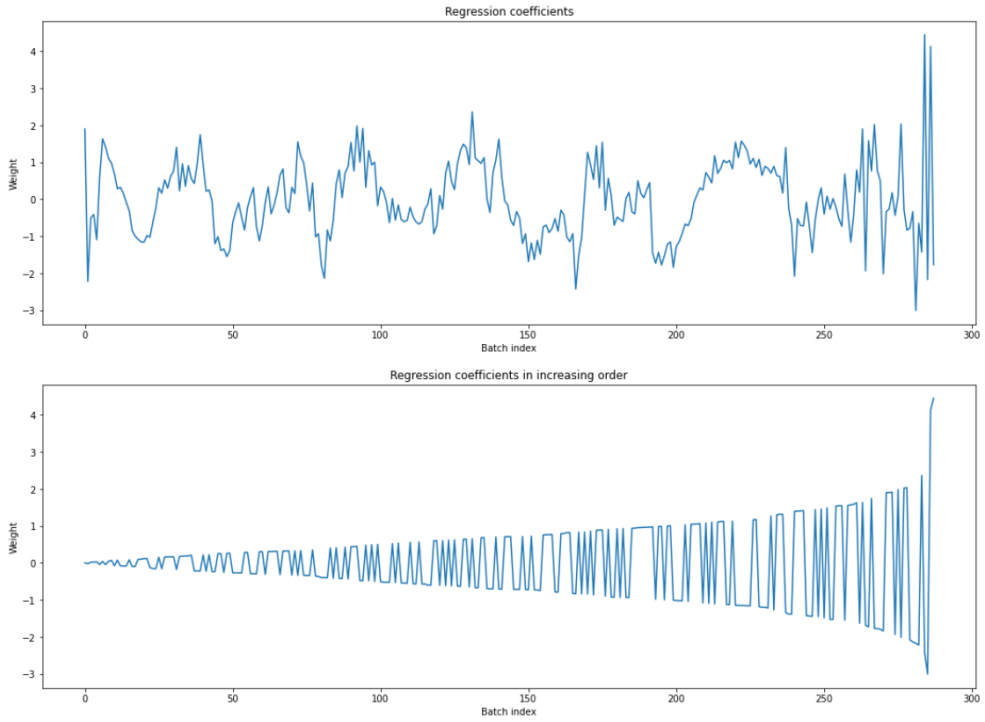


Figure 5.3: An example that illustrates the sorting procedure. The regression coefficients (top figure) is derived from regressing the mean tuber part (standardized) on the growth in Veenklooster (standardized), using 15 latent variables. The sorted coefficients (bottom figure) are sorted by the absolute value in an increasing order.

This results in a new (filtered) matrix X having one column less than before. Next, as done in the previous section, this matrix is divided into the same training and testing sets. The validation set is not needed anymore, since the hyperparameters are already optimized. Then PLS is trained on the entire training set. Finally the performance can be re-evaluated with the testing set. This procedure repeats until the matrix X has no more columns left. During this procedure, there exists a combination of filtered columns that has minimized the MSE value. This combination is then taken for comparison in section 5.3.

5.2.2. Variable Importance in Projection (VIP)

The second method requires only one extra computation for each variable $\mathbf{x}_j$, by calculating the VIP (Variable Importance in Projection) measure $v_j = v(\mathbf{x}_j)$. The VIP measure is defined in equation (5.1) and measures the importance of each variable j reflected by the loading weights $W = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k)$ from each component found in (4.3). Moreover, this scoring also utilizes the extracted scores and loadings vectors from both X and $\mathbf{y}$.

$$v_j = \sqrt{p \sum_{a=1}^{k} \left[\left(\mathbf{q}_a^2 \mathbf{t}_a' \mathbf{t}_a \right) \left(\mathbf{w}_{aj} / \|\mathbf{w}_a\| \right)^2 \right] / \sum_{a=1}^{k} \left(\mathbf{q}_a^2 \mathbf{t}_a' \mathbf{t}_a \right)}, \quad j = 1, 2, \ldots, p \tag{5.1}$$

With this measure, a variable $\mathbf{x}_j$ is omitted whenever $L < v_j < U$, for $L, U \in [0, \infty)$ and $L < U$. It is generally accepted that a variable should be selected if $v_j > 1$, but a proper threshold between 0.83 and 1.21 can yield more relevant variables, according to Mehmood et al. [15]. Once the VIP measure is calculated for all variables, the variables are filtered at once. This gives a new filtered matrix X. As an exmaple, the variable selection done for Montfrin on the standardized matrix and the SG-filtered matrix are shown in Figures 5.4 and 5.5 respectively. With this new matrix, we do the same process as described in the other method in which the data is split into the same training and testing sets.

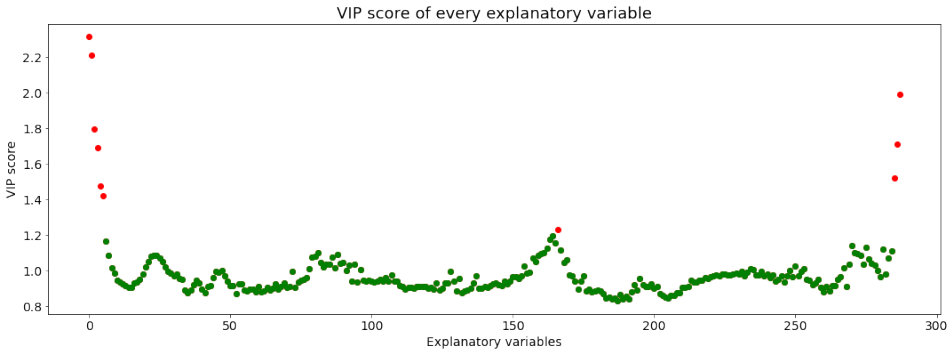


Figure 5.4: VIP score of every explanatory variable of matrix $X = X_{\text{mean}}$ standardized. The selection is done for the field located in Montfrin. Every variable omitted is colored in red.

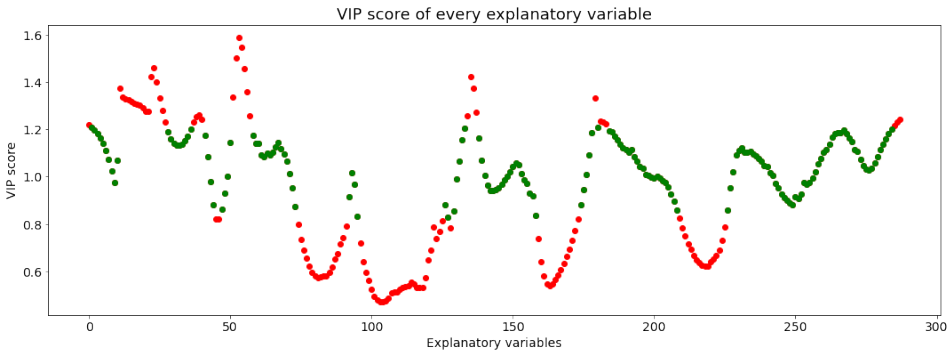


Figure 5.5: VIP score of every explanatory variable of matrix $X = X_{\text{mean}}$ passed through the SG-filter. The selection is done for the field located in Montfrin. Every variable omitted is colored in red.

5.3. Reported MSE and R^2 and overlapping features

Combining the steps described in the previous two sections yields the performance of the standardized matrix X on each tuber part and the standardized vector $\mathbf{y}$ on each field. Furthermore, this is done two more times in which matrix X is passed through the Savitzky-Golayfilter and a combination of the two normalization methods (first standardized, then SG-filtered). Tables 5.3 to 5.11 lists the MSE value and the R2 value of each possible combination, along with the performance of both variable selection methods. From these table it appears that the first method, which is sorting through the regression coefficients, returns the best performing model, in terms of MSE values and R2 values.

Since all combinations produce a different selection of variables, looking at the overlapping variables is also considered in this section. Figures 5.6, 5.7 and 5.8 shows the selected variables that appear in all fields for all possible explanatory matrix (standardized, SG-filtered and combined). In these figures the selected variables are highlighted in green and in purple. Green highlights the selection done by regression coefficients and purple highlights the selection done by VIP score.

5.3.1. Explanatory matrix standardized

Feature selection	none		Regression coefficients			VIP measure		
Montfrin	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.344249	0.603303	258	0.324372	0.626209	277	0.246804	0.715594
Pith	0.447973	0.483776	142	0.407226	0.530731	237	0.507711	0.414937
Mean	0.261075	0.699149	182	0.252997	0.708458	278	0.221015	0.745313
Average	**0.351099**	**0.595409**	**194**	**0.328198**	**0.621799**	**264**	**0.325177**	**0.625281**

Table 5.3: Predictive performance from Montfrin. The explanatory matrix X is standardized. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

Feature selection	none		Regression coefficients			VIP measure		
SPNA	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.323892	0.608208	277	0.300585	0.636400	136	0.251031	0.696343
Pith	0.321730	0.610823	50	0.277571	0.664240	121	0.343330	0.584695
Mean	0.209283	0.746843	246	0.194143	0.765157	151	0.216644	0.737939
Average	**0.284968**	**0.655291**	**191**	**0.257433**	**0.688599**	**136**	**0.270335**	**0.672992**

Table 5.4: Predictive performance from SPNA. The explanatory matrix X is standardized. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

Feature selection	none		Regression coefficients			VIP measure		
Veenklooster	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.374562	0.538768	216	0.303308	0.626509	238	0.261877	0.677526
Pith	0.548741	0.324286	136	0.531286	0.345780	181	0.617499	0.239617
Mean	0.383465	0.527804	66	0.321081	0.604624	228	0.343597	0.576898
Average	**0.435589**	**0.463619**	**139-140**	**0.385225**	**0.525638**	**215-216**	**0.407658**	**0.498014**

Table 5.5: Predictive performance from Veenklooster. The explanatory matrix X is standardized. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

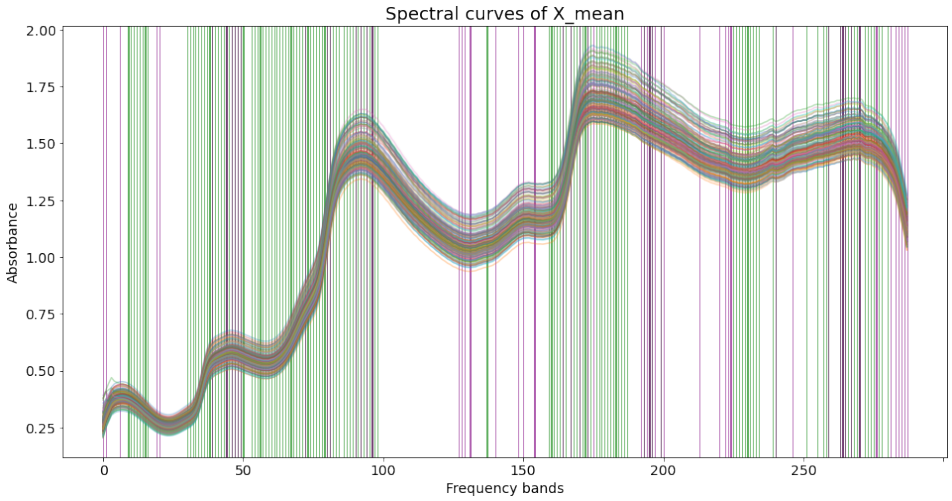


Figure 5.6: Spectral curves of the mean tuber part along with the overlapping variables in purple (regression coefficients) and green (VIP score). The spectral curves were standardized before regression and variable selection. There are 59 overlapping variables in purple and 138 overlapping variables in green.

5.3.2. Explanatory matrix SG-filtered

Feature selection	none		Regression coefficients			VIP measure		
Montfrin	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.224361	0.741457	34	0.203342	0.765678	96	0.273618	0.684695
Pith	0.352574	0.593709	40	0.303133	0.650684	96	0.515017	0.406518
Mean	0.249395	0.712608	173	0.220867	0.745483	96	0.333634	0.615535
Average	**0.275443**	**0.682591**	**82-83**	**0.242447**	**0.720615**	**96**	**0.374090**	**0.568916**

Table 5.6: Predictive performance from Montfrin. The explanatory matrix X is SG-filtered. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

Feature selection	none		Regression coefficients			VIP measure		
SPNA	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.315719	0.618094	31	0.208032	0.748356	177	0.666154	0.194194
Pith	0.313795	0.620421	68	0.233130	0.717997	160	0.304063	0.632194
Mean	0.312066	0.622513	82	0.245820	0.702646	193	0.380873	0.539281
Average	**0.313860**	**0.620343**	**60-61**	**0.228994**	**0.72300**	**176-177**	**0.450363**	**0.455223**

Table 5.7: Predictive performance from SPNA. The explanatory matrix X is SG-filtered. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

Feature selection	none		Regression coefficients			VIP measure		
Veenklooster	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.382365	0.529159	87	0.268405	0.669488	131	0.386730	0.523784
Pith	0.453221	0.441908	272	0.451057	0.444573	114	0.603283	0.257122
Mean	0.335356	0.587046	258	0.270978	0.666320	139	0.457113	0.437115
Average	**0.390314**	**0.519371**	**205-206**	**0.330147**	**0.593460**	**128**	**0.482375**	**0.406007**

Table 5.8: Predictive performance from Veenklooster. The explanatory matrix X is SG-filtered. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

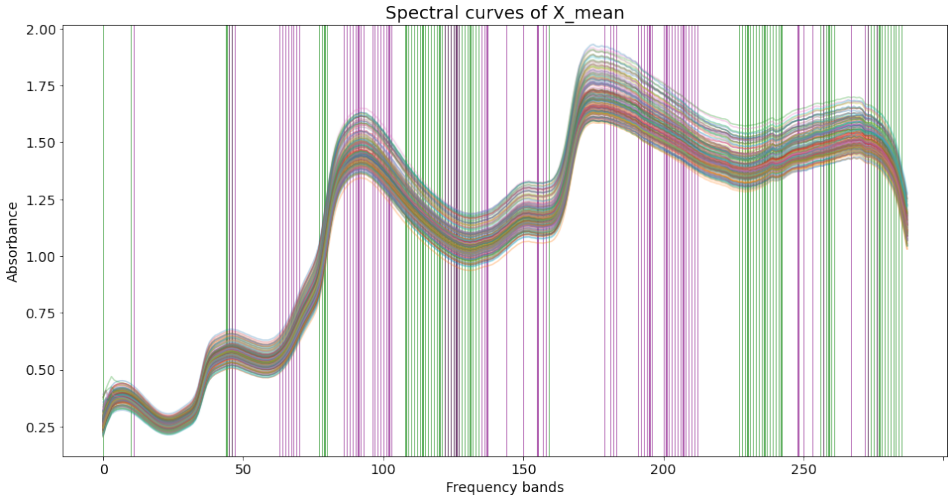


Figure 5.7: Spectral curves of the mean tuber part along with the overlapping variables in purple (regression coefficients) and green (VIP score). The spectral curves were passed through the Savitzky-Golay filter before doing regression and variable selection. There are 73 overlapping variables in purple and 72 overlapping variables in green.

5.3.3. Explanatory matrix first standardized, then SG-filtered

Feature selection	none		Regression coefficients			VIP measure		
Montfrin	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.253142	0.708290	174	0.240338	0.723046	168	0.291632	0.663936
Pith	0.418500	0.517739	49	0.241453	0.721760	160	0.493748	0.431027
Mean	0.267389	0.691872	148	0.243613	0.719271	165	0.291025	0.664636
Average	**0.313914**	**0.640703**	**128-129**	**0.249756**	**0.714025**	**163-164**	**0.351393**	**0.597836**

Table 5.9: Predictive performance from Montfrin. The explanatory matrix X is first standardized and then SG-filtered. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

Feature selection	none		Regression coefficients			VIP measure		
SPNA	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.289259	0.650101	67	0.192616	0.767004	95	0.396980	0.519797
Pith	0.316677	0.616935	45	0.229549	0.722329	85	0.439510	0.468352
Mean	0.214108	0.741007	218	0.190922	0.769053	100	0.248049	0.699951
Average	**0.273348**	**0.669348**	**110**	**0.204362**	**0.752795**	**93-94**	**0.361513**	**0.562700**

Table 5.10: Predictive performance from SPNA. The explanatory matrix X is first standardized and then SG-filtered. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

Feature selection	none		Regression coefficients			VIP measure		
Veenklooster	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Cortex	0.332486	0.332485	38	0.305506	0.623802	199	0.309361	0.619056
Pith	0.474108	0.416188	72	0.367728	0.547183	174	0.535427	0.340680
Mean	0.294904	0.636858	282	0.279936	0.655289	170	0.382638	0.528823
Average	**0.367166**	**0.461843**	**130-131**	**0.317723**	**0.608758**	**181**	**0.409142**	**0.496186**

Table 5.11: Predictive performance from Veenklooster. The explanatory matrix X is first standardized and then SG-filtered. From left to right: no variable selection used, variable selection by sorting through the regression coefficients and variable selection by means of the VIP measure.

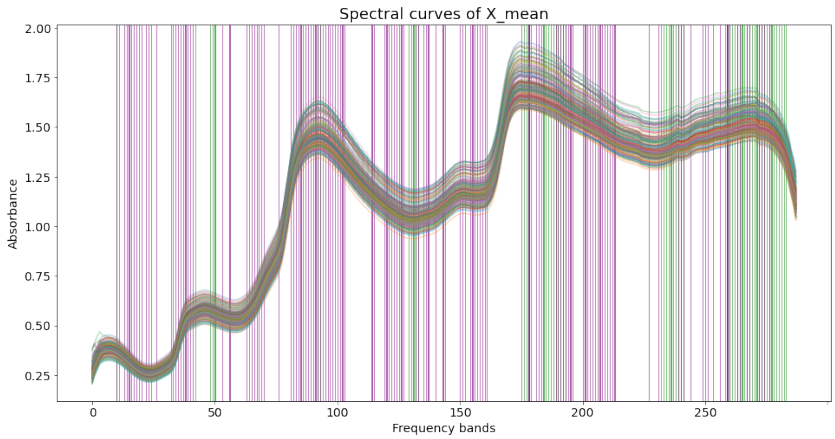


Figure 5.8: Spectral curves of the mean tuber part along with the overlapping variables in purple (regression coefficients) and green (VIP score). The spectral curves were first standardized and then passed through the Savitzky-Golay filter before regression and variable selection. There are 136 overlapping variables in purple and 62 overlapping variables in green.

5.4. Normality test on the residuals

Partial Least Squares (and other Least Squares methods) assumes that the relationship between the explanatory space and the response are linear, that is, the relationship between the hyperspectral signatures and the vitality data. Furthermore, the underlying errors, or residuals $\boldsymbol{\epsilon} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - X\boldsymbol{\beta}$, are assumed to be normally distributed. There are two ways to test normality on the residuals. One is done through visual inspection and the other is done with hypothesis testing. In the second case, the null hypothesis states that the residuals are indeed normally distributed, whereas the alternative hypothesis states that the residuals are not normally distributed. Each statistical test returns a p-value and if it is less than the predetermined significance level, say $\alpha = 0.05$, then the null hypothesis can be rejected. This implies that residuals do not form a normal distribution, which violates the assumption about linear relations. The following two normality tests are used: the Shapiro-Wilk test and the D'Agostino's K-squared test. The Shapiro-Wilk test measures how likely the sample was drawn from a Gaussian distribution and the D'Agostino's K-squared test calculates the skewness and kurtosis from the sample to determine if this sample departs from the normal distribution. Skewness quantifies asymmetry in the distribution and kurtosis quantifies how much of the sample is present in the tail.

In this section only the residuals $\boldsymbol{\epsilon}$ of each field on $X = X_{\text{mean}}$ is given, where X is first standardized and then SG-filtered. These residuals are shown in Figures 5.9, 5.10 and 5.11 for fields Montfrin, SPNA and Veenklooster respectively. In all three figures the scatter plots (top-left) and the residual by predicted plots (top-right) are for the most part scattered randomly around the center line of zero. However when inspecting the QQ-plots (bottom-right) on each figure, outliers are present on the tail end of the curve. From these figures it seems that normality is present in the residuals.

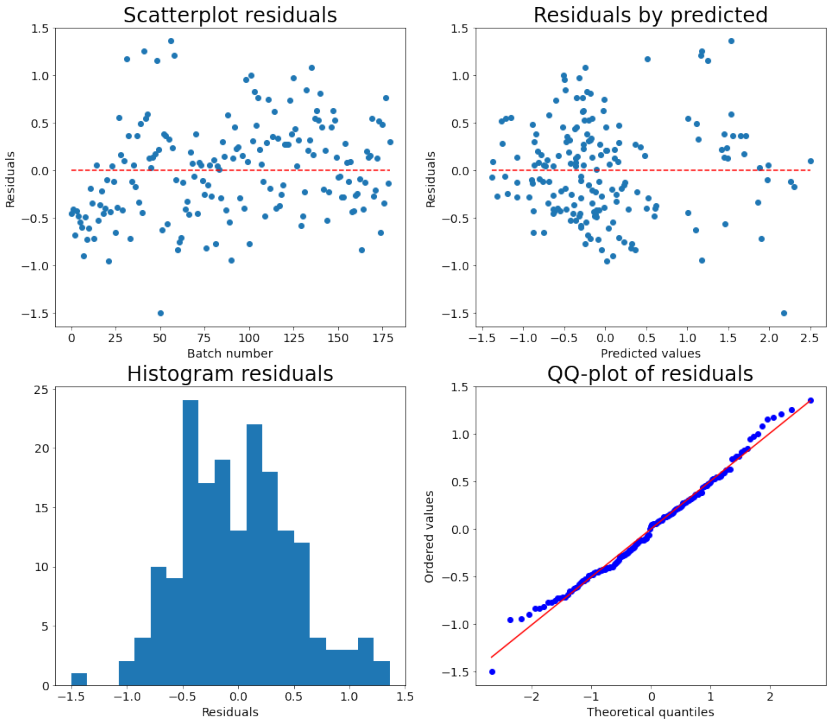


Figure 5.9: Scatterplot of the residuals (top-left), residuals vs predicted values (top-right), histogram of residuals (bottom-left), QQ-plot of residuals (bottom-right). This model regresses the field Montfrin on the mean tuber part.

The p-values of the corresponding hypothesis tests are found in Table 5.12. From this table the residuals follow a normal distribution for field Montfrin and SPNA ($p > \alpha$), but not on the field Veenklooster ($p \leq \alpha$).

	Montfrin	SPNA	Veenklooster
Shapiro-Wilk	0.220	0.081	0.000
D'Agostino	0.405	0.584	0.000

Table 5.12: p-values of each statistical test. The model regression each field on the mean tuber part.

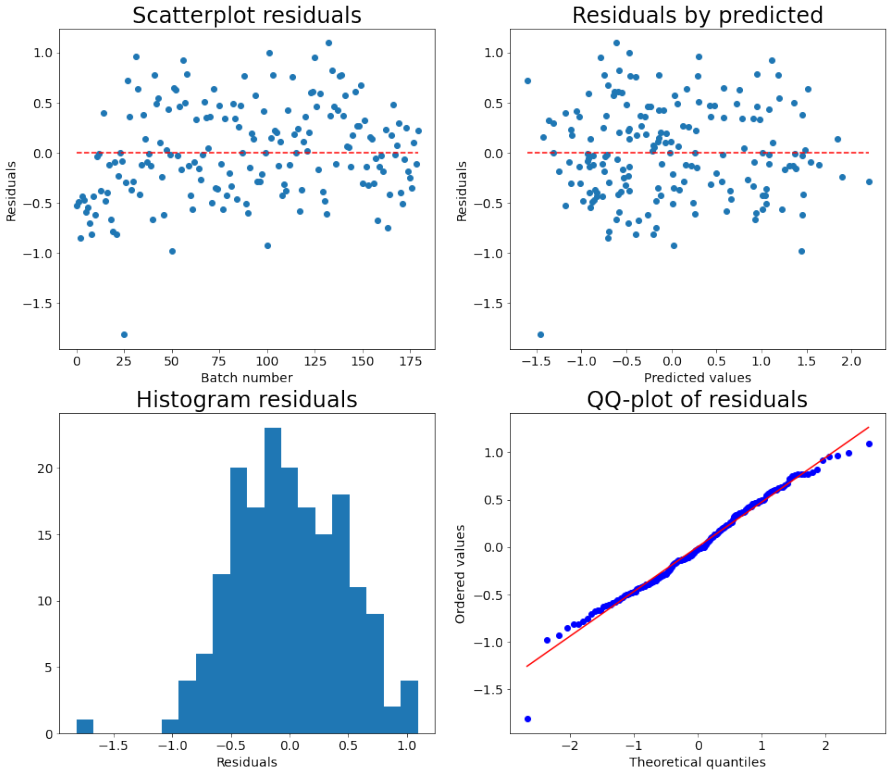


Figure 5.10: Scatterplot of the residuals (top-left), residuals vs predicted values (top-right), histogram of residuals (bottom-left), QQ-plot of residuals (bottom-right). This model regresses the field SPNA on the mean tuber part.

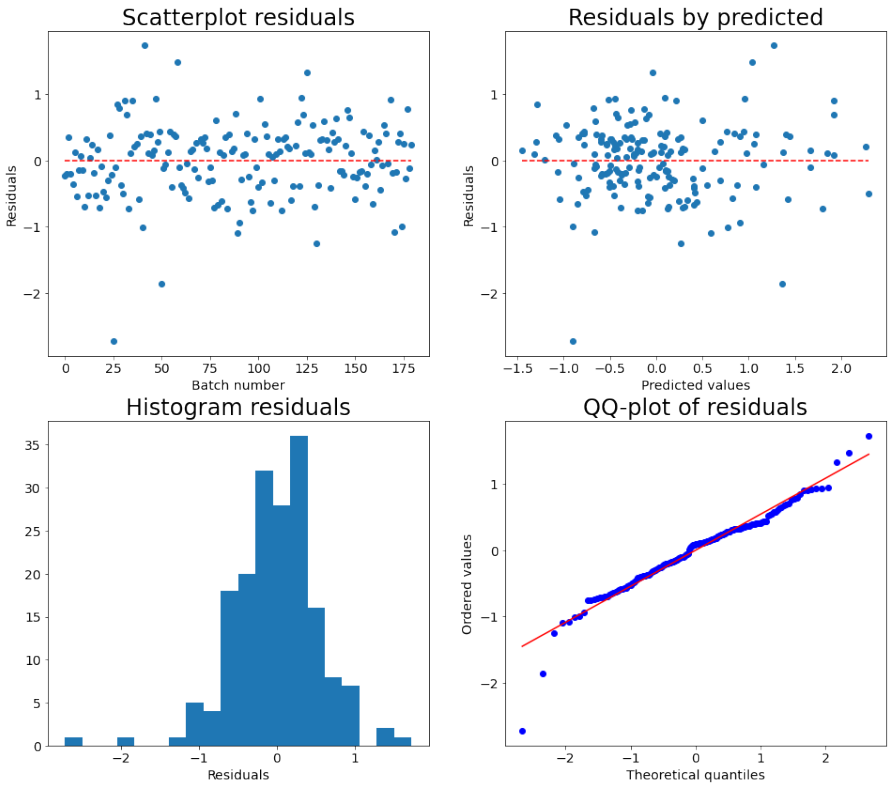


Figure 5.11: Scatterplot of the residuals (top-left), residuals vs predicted values (top-right), histogram of residuals (bottom-left), QQ-plot of residuals (bottom-right). This model regresses the field Veenklooster on the mean tuber part.

6

Experimentation

In this chapter we conduct a number of experiments with the baseline model derived from chapter 5. The first experiment aims to predict the growth from one field by learning from the other field(s), to see if there is any predictive information. The second experiment does something similar as it tries to predict the spectral curves of pith, while learning from the cortex and vice versa. This is a multivariate regression problem, compared to the previous models. Variable selection is also considered on both experiments to see if there any overlapping variables found. The next experiment learns from five different varieties and evaluates on the remaining one. The final experiment compares NIPALS with LSQR, in terms of speeds and accuracy.

6.1. Evaluating performances from different fields

In this experiment the baseline PLS model is fitted on the mean tuber part and on one field (or on multiple fields), which produces an estimated growth of said field(s). This estimation is then compared with the existing growth of the remaining field(s). The evaluated MSE and R^2 values are found in Table 6.3. The explanatory matrix X is standardized and variable selection is not considered yet. Furthermore, splitting the matrix X into training and testing sets is in this experiment not necessary.

	Montfrin		SPNA		Veenklooster	
	MSE	R^2	MSE	R^2	MSE	R^2
Montfrin	-	-	0.485859	0.480469	0.412393	0.518464
SPNA	0.492843	0.478778	-	-	0.478808	0.440913
Veenklooster	0.357156	0.622279	0.418465	0.552533	-	-
TWO SETS	0.376363	0.601966	0.422445	0.548278	0.377595	0.559096

Table 6.1: Each cell contains the MSE value on the left and the R2 value on the right. Rows are the training sets and columns are the testing sets. The last row in this table reports the performance of the model that has trained on the other two fields combined. Partial Least Squares is used where X is taken from the mean tuber part. Variable selection is left out.

It appears from Table 6.3 that the baseline model is able to predict the growth of a (new) field quite well, while withholding any prior information about said field. This also seems to apply when training on two fields, as indicated by the final row of Table 6.3, because the evaluated MSE and R^2 values have improved a bit. It appears that the field located in Veenklooster has the best predictive information and SPNA seems to have the worst, although it comes close with the field located in Montfrin, in terms of MSE and R^2 values.

The same experiment is repeated again, only this time variable selection has been taken into consideration. Only the first method will be used, where the regressions coefficients are sorted by their absolute values. The results of this experiment can be found in Table 6.4, in which the structure is similar as in Table 6.3. The results of adding variable selection does slightly increase the performance. By including variable selection on all three fields, we can also look for overlapping variables within the fields. These variables are illustrated in green vertical lines in Figure 6.1. In total, 45 out of 288 variables are overlapping in all three fields.

	Montfrin			SPNA			Veenklooster		
	features	MSE	R^2	features	MSE	R^2	features	MSE	R^2
Montfrin	-	-	-	85	0.4638	0.5040	120	0.3977	0.5356
SPNA	68	0.4794	0.4930	-	-	-	105	0.4750	0.4454
Veenklooster	70	0.3560	0.6235	131	0.4098	0.5619	-	-	-
TWO SETS	116	0.3706	0.6081	110	0.4020	0.5701	106	0.3682	0.5701

Table 6.2: Each cell contains the MSE value on the left and the R^2 value on the right. Rows are the training sets and columns are the testing sets. The last row in this table reports the performance of the model that has trained on the other two fields combined. Partial Least Squares is used where X is taken from the mean tuber part. Variable selection is now considered, but only the first method is used.

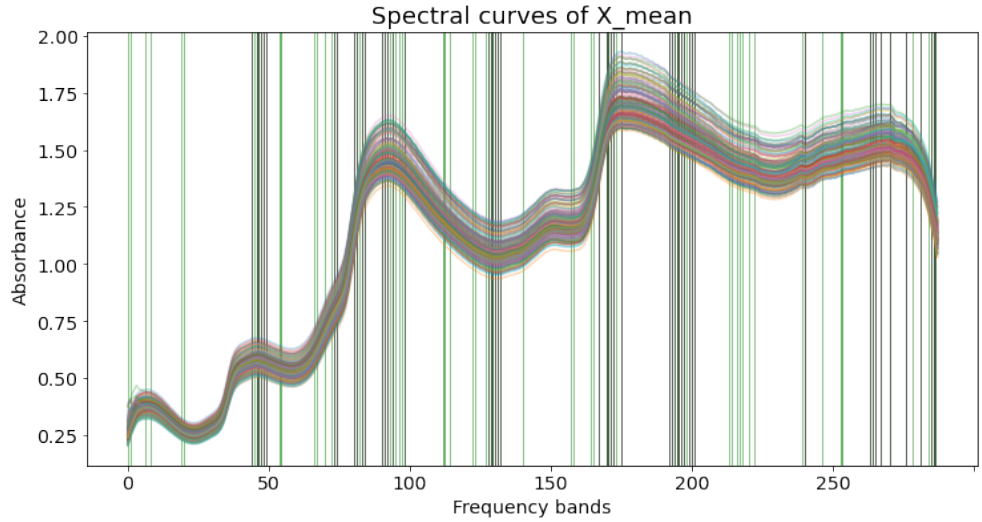


Figure 6.1: Overlapping variables displayed in purple vertical lines after performing feature selection on the six different runs mapping one field to the other field and green vertical lines for the three runs that maps two fields to one. The columns, that were left out after variable selection, are interpreted as the frequency bands of the spectral signature.

6.2. Regress pith on cortex and vice versa

Since the baseline model is also capable of doing multivariate regression, one can look at doing regression on the tuber parts. In this experiment regression is done for pith on cortex and the other way around. The mean tuber part is left out, since this tuber part consist of both the cortex and pith already. This experiment yields two regression matrices $B_{c\to p}, B_{p\to c}$, instead of regression vectors. Both matrices have dimensions $(n \times n)$, where each column of the explanatory space corresponds to one frequency band and assigns weights to the entire spectra in the response space. The rows of these regression matrices reports how the entire spectra of the explanatory space assigns weights to one frequency band of the response space. A visual illustration of the shape of both matrices $B_{c\to p}, B_{p\to c}$ is given in Figure 6.2.

$$X_{\text{cortex}} = X_{\text{mean}} B_{p\to c} + E_{p\to c} \tag{6.1}$$

$$X_{\text{mean}} = X_{\text{cortex}} B_{c\to p} + E_{c\to p} \tag{6.2}$$

The steps taken for this experiment are as follows. Following the same procedure as described in sections 5.1 and 5.2.1, the number of latent variables used is optimized and variable selection is done on a fixed split (training, validation and testing sets). The results of this experiment, in terms of its predictive performance, is found in Tables 6.3 (no variable selection) and 6.4 (variable selection included, based on regression coefficients).

		Pith		Cortex	
	# latent variables	MSE	R^2	MSE	R^2
Pith	9	-	-	0.0850	0.8992
Cortex	7	0.0847	0.9101	-	-

Table 6.3: Predictive performance of one tuber part on the other. The rows are the training sets and the columns are the testing sets.

		Pith			Cortex		
	# latent variables	features	MSE	R^2	features	MSE	R^2
Pith	9	-	-	-	180	0.0831	0.9012
Cortex	7	183	0.0844	0.9105	-	-	-

Table 6.4: Predictive performance of one tuber part on the other. The rows are the training sets and the columns are the testing sets. Only the first variable selection method is utilized.

Given one tuber part, the model is capable of accurately predicting the hyperspectral curves from the other tuber part. Including variable selection does not give a significant increase in performance, but the number of variables used is lowered to around 180. The regression matrices $B_{c\to p}, B_{p\to c}$ have a unique shape, as outlined in Figure 6.2. The values of each regression vector inside both matrices closely resemble each other, since almost every row gives off one color. Furthermore, when comparing both matrices with each other, it seems that the signs alternate between each other as the colors alternate too. In both matrices, it seems that the columns can be divided in three sections. The vertical lines that distinguishes these sections are roughly found in columns 80 and 165 when reading both matrices from left to right.

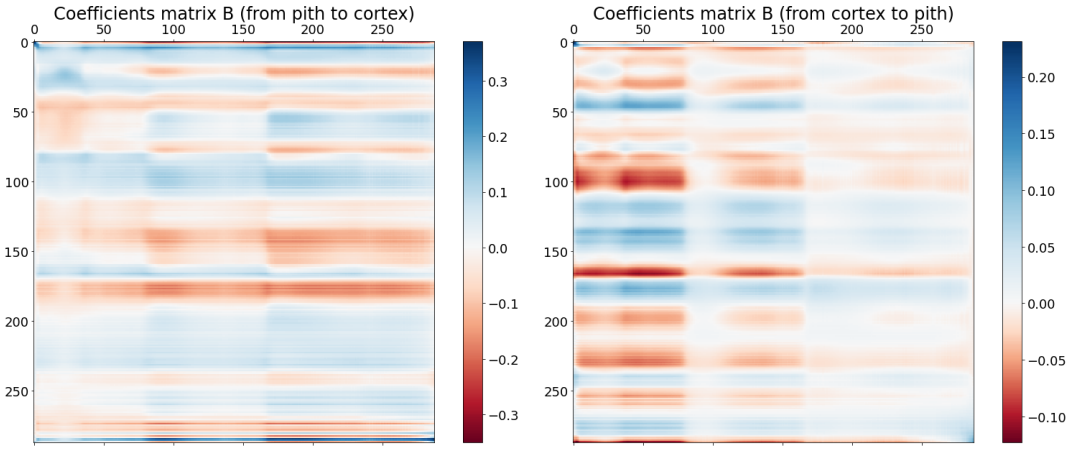


Figure 6.2: Rough outline of regression matrices $B_{p\to c}$ on the left and $B_{c\to p}$ on the right. Shades of blue correspond to positive elements and shades of red to negative matrix elements.

6.3. Leave-one-variety-out Cross Validation

In this experiment one variety is left out as the testing set and the model trains with the remaining varieties. The performance of this experiment gives an indication of how this model is going to interpret new varieties. However, with the current explanatory matrix X taken from different tuber parts, there is a certain variety-effect present during the normalization process. Moreover, this effect is dominant. If ignored, the regression vector will not perform well on the test set. By dividing the matrix X into six blocks, where each block corresponds with one variety, and then apply normalization on each block, then this effect is removed from the matrix X. The same process is repeated for the response vector $\mathbf{y}$ for each field.

$$\tilde{\mathbf{y}} = \tilde{X}\tilde{\boldsymbol{\beta}} \quad \rightarrow \quad \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \\ \tilde{\mathbf{y}}_3 \\ \tilde{\mathbf{y}}_4 \\ \tilde{\mathbf{y}}_5 \\ \tilde{\mathbf{y}}_6 \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \\ \tilde{X}_4 \\ \tilde{X}_5 \\ \tilde{X}_6 \end{bmatrix} \boldsymbol{\beta} \quad \rightarrow \quad \mathbf{y}_i = X_i\boldsymbol{\beta}, \tag{6.3}$$

$$\mathbf{y}_i = \mathrm{var}_i^{-1} N_i \tilde{\mathbf{y}}_i, \quad X_i = N_i \tilde{X}_i V_i^{-1}, \quad N_i = I - n_i^{-1}\mathbf{1}\mathbf{1}^T, \tag{6.4}$$

where $n_i = n/6 = 30$ in the present case and the diagonal matrices V_i contain the variances of the columns of the matrices $\tilde{X}_i$.

After this alternative normalization, both the explanatory matrix X and response vector $\mathbf{y}$ are split into six different training and testing sets, where one variety is left out from the other five. With this kind of split the model learns from five varieties and evaluates its performance on the remaining variety. The reported MSE and R^2 values are listed in Table 6.5.

	Montfrin		SPNA		Veenklooster	
Variety left out	MSE	R^2	MSE	R^2	MSE	R^2
Variety 1	0.9995	0.0005	1.0599	-0.0599	1.2428	-0.2428
Variety 2	1.2947	-0.2947	1.4126	-0.4126	1.6545	-0.6545
Variety 3	1.5797	-0.5797	1.1477	-0.1477	1.7183	-0.7183
Variety 4	1.5850	-0.5850	1.0910	-0.0910	1.3975	-0.3975
Variety 5	1.5596	-0.5596	1.2378	-0.2378	1.6688	-0.6688
Variety 6	1.3142	-0.3142	1.1058	-0.1058	1.7774	-0.7774
Average	**1.3888**	**-0.3888**	**1.1758**	**-0.1758**	**1.5766**	**-0.5766**

Table 6.5: Predictive information of each split, where one variety is left out as a testing set. The data is split in 6 different varieties, then standardized on each of them.

This experiment is identical to performing 6-Fold cross-validation on the entire data. The performance however is rather poor, when observing the reported MSE and R^2 values in Table 6.5. Also, the R^2 values are all negative on each field. Regardless, even after altering the standardization process, it seems that the model is not able to give good predictions on new (incoming) varieties. Even though this experiment is done on the average growth of all three fields, the growth of one particular field (Montfrin), while normalizing the growth on each variety, is shown in Figure 6.3. Overall, there is a strong variety-effect present in the data and one regression vector $\boldsymbol{\beta}$ is unable to do any predictions on the vitality data.

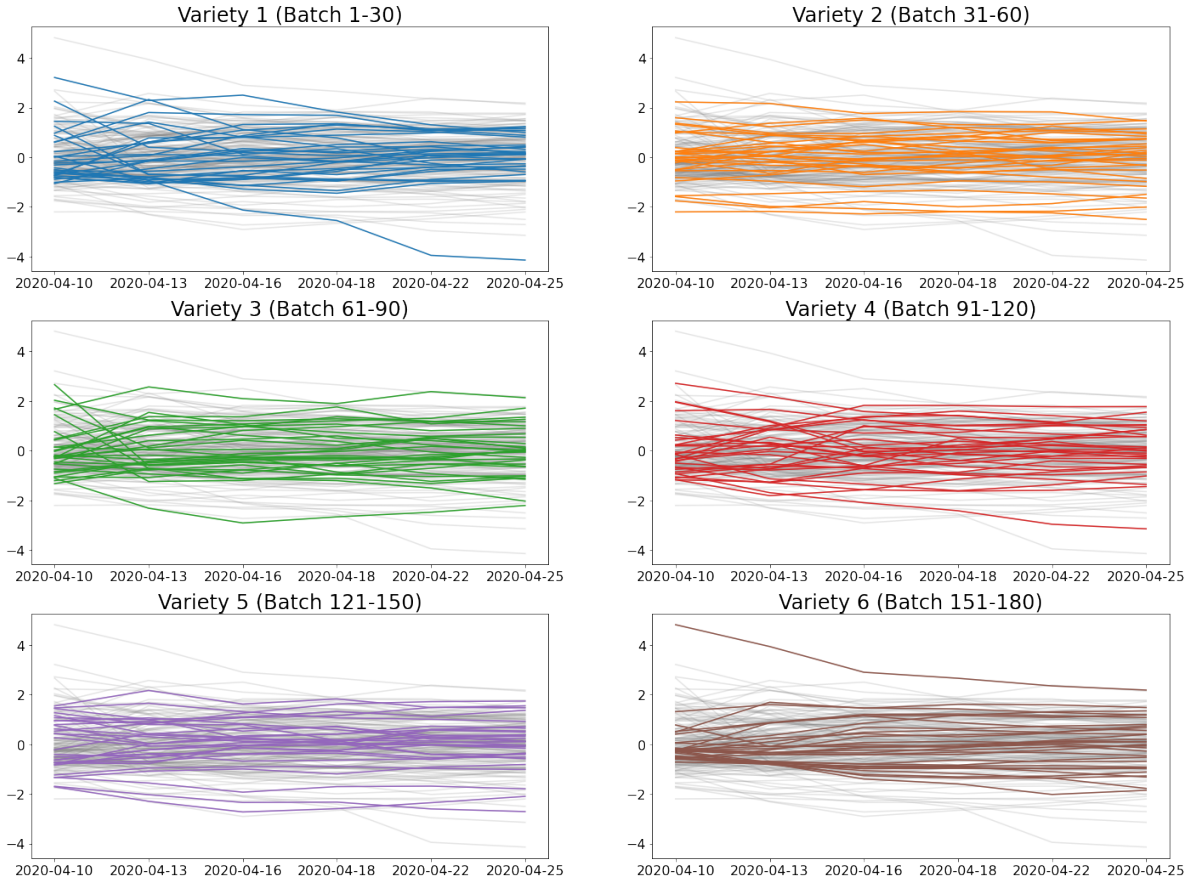


Figure 6.3: Montfrin, where each variety is normalized, as described in equation (6.3).

6.4. NIPALS-PLS vs LSQR

In this section a comparison is made between the standard NIPALS-PLS method and the possibly faster LSQR method. These methods use different approaches to find solutions over the same Krylov subspace. Therefore it is not guaranteed that the found regression vectors and the associated scores are the same. A numerical experiment is conducted, where NIPALS-PLS and LSQR are compared in terms of speed and performance. For this experiment one fixed split (same split for both methods) is considered on matrix X and $\mathbf{y}$, where the explanatory matrix $X = X_{\text{mean}}$ is first standardized and then SG-filtered.

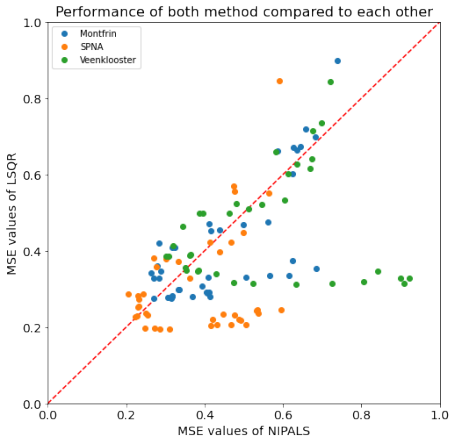


Figure 6.4: The performance of NIPALS (horizontal axis) compared to the performance of LSQR (vertical axis). Each dot points the MSE values of both estimators on a fixed number of iterations/latent variables. Each dot is also given a different color for each field. These dots should be on the red dotted line.

The steps taken in this experiment are similar to the experiment described in chapter 5, but doing variable selection with the VIP-score is not possible. This is because LSQR does not utilize the scoring and loadings matrices of explanatory matrix X. Doing variable selection based on the regression coefficients is still possible. It is assumed that the number of iterations used in LSQR are identical to the number of latent variables used in PLS, as explained in chapter 4. Figure 6.4 compares the MSE values between both methods for iterations ranging from 1 to 50. This should produce a straight line, but the figure shows something different. This goes against the assumption, since LSQR gives smaller MSE values than NIPALS-PLS for dots under the diagonal line and larger MSE values than NIPALS-PLS for dots above the diagonal line. Figure 6.5 gives a better illustration how both methods evolve over each iteration, by looking at the absolute difference of the MSE values of both methods.

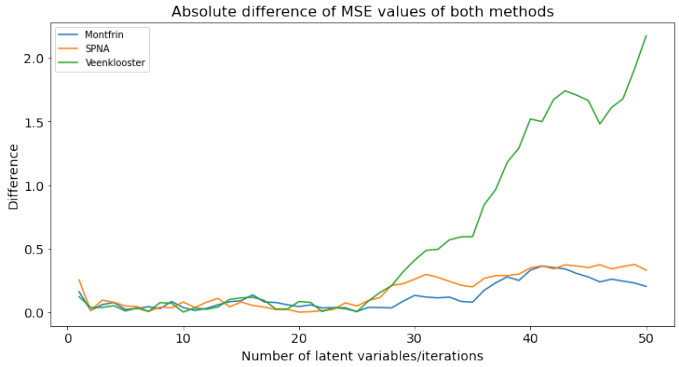


Figure 6.5: Absolute difference of the MSE values of both methods (NIPALS-PLS and LSQR) evolved over each number of latent variables/iterations.

From Figures 6.4 and 6.5 we conclude that the predictions of NIPALS-PLS and LSQR are not the same. Regardless, one can still look at the performance of LSQR as an independent estimator. The hyperparameter of LSQR, which is the number of iterations used, is first optimized with (stratified) cross-validation and then evaluated on the testing set. Moreover, the duration of this experiment is now recorded as well. The recorded times are all derived from one machine that did all computations, which ensures that there is some consistency present in these numbers. The results can be found in Table 6.6 and the duration of this entire experiment took roughly 195 seconds. Doing this exact experiment on NIPALS-PLS takes around 420 seconds. From this we see that LSQR does a similar prediction on the growth, but does the computation twice as fast.

	Montfrin			SPNA			Veenklooster		
	features	MSE	R^2	features	MSE	R^2	features	MSE	R^2
No variable selection	-	0.2820	0.6750	-	0.2208	0.7329	-	0.3046	0.6250
With variable selection	114	0.2299	0.7350	168	0.2021	0.7555	154	0.2764	0.6596

Table 6.6: Performance of LSQR as an independent estimator, where $X = X_{\text{mean}}$. In this experiment variable selection is done too. This selection method is based on the regression coefficients.

7

Discussion

This chapter discusses the overall performance of the (basline) NIPALS-PLS method and investigates which variables are left after doing two different kinds of variable selection. Finally this chapter discusses normality present in the residuals

Tables 5.3 to 5.11 report the predictive performance of every possible combination of regression, where we predict the growth of each field from each tuber part and also specify whether variable selection is used. This is done three times for each type of data normalization. From these tables we observe that the field Veenklooster has the worst predictive information compared to the other two fields, since the MSE value is higher and the R^2 value is lower in all cases. Moreover, the prediction of the growth in SPNA is in almost all cases the best. Doing variable selection, where we sort through the regression coefficients, gives the best improvement in Tables 5.3 to 5.11. However the number of variables selected is fairly inconsistent, compared to doing variable selection based on the VIP score. This selection method relies only on the values of the regression estimator defined in (4.9), and thus relies on both X and $\mathbf{y}$. The same statement can be made for the other variable selection method, which is based on the VIP score, but this method also has a fixed interval for which variables are selected. The interval used for this thesis is $[0.83; 1.12]$, as recommended by Mehmood et al. [15], but this decreases the performance compared to not doing any variable selection. Performing regression of each field on the pith tuber part results in the worst performance, compared to tuber parts cortex and mean. This is found in all tables, except for Table 5.4. Moreover, when comparing the performance of the cortex with the performance of the mean the results are for the most part similar in terms of MSE and R^2 values. From this it seems that the pith tuber part is generally doing worse prediction on the growth, than the other two tuber parts.

Figure 5.6 shows the overlapping variables that are present in Tables 5.3, 5.4 and 5.5 for both selection methods. The same is done for Figure 5.7 with Tables 5.6, 5.7 and 5.8 and Figure 5.8 with Tables 5.9, 5.10 and 5.11. From Figure 5.6 the selection method, in which variables are selected based on the regression coefficients, has less variables overlapping than the other selection method (58 compared to 138). This is as expected, since the other selection method has on average more variables selected. Therefore, when the hypercubes are standardized, we see that sorting through the regression coefficients yields less consistent selected variables than selecting variables based on the VIP score. For the second figure we have that both methods are equally consistent (73 and 72 overlapping variables). For the third figure, Figure 5.8, we have that selecting variables based on the regression coefficients is more consistent than based on the VIP score (136 and 62). The selection of variables differs for each normalization method and thus there is no consistency present between the pre-processing methods themselves.

Combining the results of Figures 5.9, 5.10 and 5.11 with Table 5.12 it appears that there is some normality present in the residuals. Even though both statistical tests reject the null hypothesis on the field Veenklooster, the null hypothesis was not rejected on the other fields. The reason why the null hypothesis was rejected, might be because of the presence of outliers in the FtV data.

8

Conclusion

This thesis models the relation between the hyperspectral signatures of the potato tuber X with the average growth of the plants $\mathbf{y}$, by means of regression. Partial Least Squares is used to find latent variables that explain the variance in X as best as possible, while providing good predictions on the growth $\mathbf{y}$. PLS does that by extracting scores and loadings matrices from the Flight to Vitality data, which is done with the Nonlinear Iterative Partial Least Squares algorithm (NIPALS). The matrix found in this algorithm has vector that span the Krylov Subspace of $X^T X$ and $X^t \mathbf{y}$, which gives rise to other Krylov Subspace methods, such as LSQR. For the PLS model, the number of latent variables used is found with cross-validation. Furthermore, both X and $\mathbf{y}$ are split into training and testing sets. With this split the performance of PLS is generated. From the results we can conclude that regression does relate the growth of the potato plants (after planting) on the NIR hypercubes of the potato tubers (prior to planting). Furthermore after variable selection, the predictive performance does increase when sorting through the regression coefficients, but decreases when selecting variables based on the VIP score.

The first selection method guarantees an improvement, but the number of variables omitted is inconsistent. The second selection decreases the performance, but the selected variables are consistent. One reason why the performance decreases might be because of the chosen interval for which the variables are omitted, even tough this interval was recommended by Mehmood et al. [15]. Despite the decrease, calculating the VIP score is faster than omitting variables based on the assigned weights, since there is only one computation done. Therefore looking at different intervals for the VIP score is highly recommended for future work. More recommendations for future work are discussed in the next section.

With the established implementation of PLS, we perform a total of four experiments on the Flight to Vitality data. From the first experiment, where we train on one field to predict the other two, we see that the model is still capable of predicting the growth of one field even tough the information is withheld ($R^2 \approx 0.6$). In the second experiment one tuber part is regressed on the other tuber part and vice versa. We see from this experiment that the model can almost fully predict the spectral curves of the other tuber part ($R^2 \approx 0.9$). Furthermore, the corresponding regression matrices have a unique shape, where the entire spectra of the explanatory matrix assigns almost the same weights to each frequency band of the response space. The third experiments does cross-validation on the FtV data, where each variety is left out as the testing set. Moreover, the data is normalized on each variety in order to remove any variety-effect present. From the results we conclude that the model does not eliminate the variety-effect present, since the model is incapable of doing any prediction on (new) incoming varieties ($R^2 < 0$). The last experiment compares the predictive performance of NIPALS with LSQR. LSQR fails to produce the same prediction as NIPALS-PLS, but the computation speed is much greater. Therefore by interpreting it as a separate estimator, where its own hyperparameter is optimized, the performance of this estimator is also presented in this thesis. Compared to NIPALS, the estimator derived from LSQR is doing similar predictions, while the duration of the computation is lowered significantly.

8.1. Future work

The rectangles that labels the tuber parts cortex and pith in Figure 2.1 are not identifying the actual parts properly. The image made by Laimbeer et al. gives the actual distinction of what constitutes a cortex and a pith, as illustrated in Figure 8.1. The process of making this new distinction is a separate job that involves more image processing, rather than doing regression on the vitality data. Once these distinctions are made for each recorded tuber, then the mean spectrum on the new areas can be computed, as done before with the existing matrices. The explanatory matrices that arises from this new distinction $X'_{\text{cortex}}, X'_{\text{pith}}$ can be compared with the existing matrices to see if this there is any improvement present in terms of predictive information.

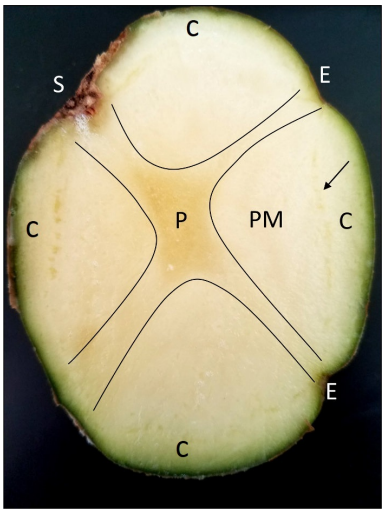


Figure 8.1: Internal morphology of a potato tuber. The separation between pith (P) is illustrated with black lines. The vascular ring, which separates parenchyma (PM) from the cortex (C) is denoted with a black arrow [13].

This thesis has implemented two filtering methods for variable selection on the Flight to Vitality data. Mehmood et al. have presented multiple selection methods for variable selection with PLS. This paper categorizes each method into three categories: filtering, wrapping and embedding. An overview of multiple variable selection methods used with PLS is given in Figure 8.2. For future work, on can look at which selection method produces the best improvement in terms of predictive performance or which method does the fastest selection, while improving the performance.

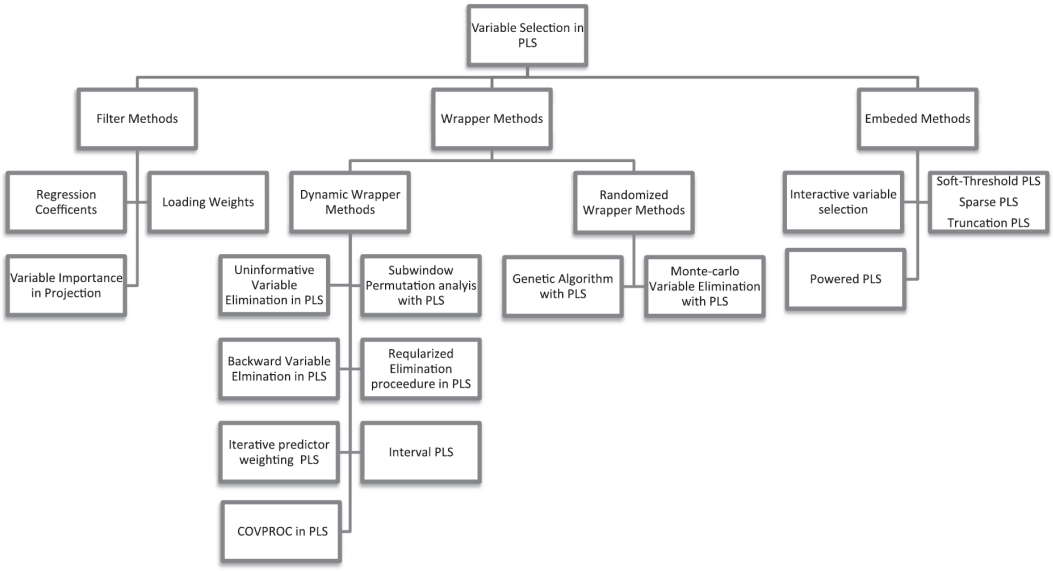


Figure 8.2: An overview of methods used for variable selection with PLS, made by Mehmood et al. [15].

Since the algorithm NIPALS is based on Krylov Subspace methods, one could also explore this connection even further. This thesis attempted to relate NIPALS with LSQR, but this has somehow failed in terms of equivalence. For future work, one could investigate why the two estimators are giving different predictions with one fixed number of iterations/latent variables.

Another recommended future works would be to spend more time on detecting outliers in the FtV data. This could have been the reason why both statistical tests rejected the null hypothesis in section 5.4. Furthermore, removing outliers makes the model perform better, since outliers have great (negative) influence on the fitting of a model.

As a final recommendation, there exist a preprocessing method, called **O**rthogonal **P**rojections to **L**atent **S**tructures (O-PLS). According to Trygg and Wold, this method removes variation from the explanatory space that is not correlated to the response space [23]. Adopting O-PLS in that paper resulted in an improvement in terms of predictive performance. By following the algorithm described in that paper, one could improve the existing implementation by removing variations in X that are correlated to Y from the Flight to Vitality data and repeat the existing experiment to see if there is any improvement found.

Bibliography

[1] Åke Björck and Ulf G Indahl. Fast and stable partial least squares modelling: a benchmark study with theoretical comments. *Journal of Chemometrics*, 31(8):e2898, 2017.

[2] Mélanie Blazere, Fabrice Gamboa, and Jean-Michel Loubes. Partial least squares a new statistical insight through orthogonal polynomials. In *19th European Young Statisticians Meeting*, volume 13, page 12, 2015.

[3] Franziska Buchholz, Livio Antonielli, Tanja Kostic, Angela Sessitsch, and Birgit Mitter. The bacterial community in potato is recruited from soil and partly inherited across generations. *PLOS ONE*, 14: e0223691, 11 2019. doi: 10.1371/journal.pone.0223691.

[4] Alison J Burnham, Roman Viveros, and John F MacGregor. Frameworks for latent variable multivariate regression. *Journal of chemometrics*, 10(1):31–45, 1996.

[5] International Potato Center. How potato grows, Sep 2017. URL `https://cipotato.org/potato/how-potato-grows/`.

[6] Merel Engelsman. Flight to vitality. URL `https://www.tudelft.nl/agtech/projects/flight-to-vitality`.

[7] Hannes Feilhauer, Gregory P Asner, Roberta E Martin, and Sebastian Schmidtlein. Brightness-normalized partial least squares regression for hyperspectral data. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 111(12-13):1947–1957, 2010.

[8] LLdiko E Frank and Jerome H Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[9] Silvia Gazzola, Enyinda Onunwor, Lothar Reichel, and Giuseppe Rodriguez. On the lanczos and golub–kahan reduction methods applied to discrete ill-posed problems. *Numerical Linear Algebra with Applications*, 23(1):187–204, 2016.

[10] Aoife Gowen, James Burger, Carlos Esquerre, Gerry Downey, and Colm O'Donnell. Near infrared hyperspectral image regression: on the use of prediction maps as a tool for detecting model overfitting. *Journal of Near Infrared Spectroscopy*, 22(4):261–270, 2014.

[11] Averis Seeds B.V. HZPC Holding B.V. Projectplan flight to vitality. `https://bo-akkerbouw.nl/ajax_frontoffice/filemanager/files/2020/Projecten/PROJECTPLAN-Fligth-to-Vitality-16062017.pdf`.

[12] Roger King, Chris Ruffin, F.E. LaMastus, and David Shaw. Analysis of hyperspectral data using savitzky-golay filtering-practical issues (part 2). volume 1, pages 398 – 400 vol.1, 02 1999. ISBN 0-7803-5207-6. doi: 10.1109/IGARSS.1999.773512.

[13] Parker Laimbeer, Melissa Makris, and Richard Veilleux. Measuring endoreduplication by flow cytometry of isolated tuber protoplasts. *Journal of Visualized Experiments*, 2018, 03 2018. doi: 10.3791/57134.

[14] Renfu Lu. Detection of bruises on apples using near–infrared hyperspectral imaging. *Transactions of the ASAE*, 46(2):523, 2003.

[15] Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118: 62–69, 2012.

[16] Govindarajan Konda Naganathan, Lauren M Grimes, Jeyamkondan Subbiah, Chris R Calkins, Ashok Samal, and George E Meyer. Visible/near-infrared hyperspectral imaging for beef tenderness prediction. *Computers and electronics in agriculture*, 64(2):225–233, 2008.

[17] Christopher C Paige and Michael A Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):43–71, 1982.

[18] Aloke Phatak and Frank de Hoog. Exploiting the connection between pls, lanczos methods and conjugate gradients: alternative proofs of some properties of pls. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 16(7):361–367, 2002.

[19] Philip H Plaisted. Growth of the potato tuber. *Plant physiology*, 32(5):445, 1957.

[20] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"*, pages 34–51. Springer, 2005.

[21] V. Sowmya, K. P. Soman, and M. Hassaballah. *Hyperspectral Image: Fundamentals and Advances*, pages 401–424. Springer International Publishing, Cham, 2019. ISBN 978-3-030-03000-1. doi: 10.1007/978-3-030-03000-1_16. URL https://doi.org/10.1007/978-3-030-03000-1_16.

[22] Randall D Tobias et al. An introduction to partial least squares regression. In *Proceedings of the twentieth annual SAS users group international conference*, volume 20. SAS Institute Inc Cary, 1995.

[23] Johan Trygg and Svante Wold. Orthogonal projections to latent structures (o-pls). *Journal of Chemometrics: A Journal of the Chemometrics Society*, 16(3):119–128, 2002.

[24] Svante Wold, Arnold Ruhe, Herman Wold, and WJ Dunn, Iii. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.