CHALMERS
UNIVERSITY OF TECHNOLOGY

Symptom Quantification for Parkinson's Disease

Master's thesis in Systems, Control and Mechatronics

MAREIKE WENDEBOURG

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's Thesis EX017/2016

Symptom Quantification for Parkinson's Disease

Mareike Wendebourg

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden 2016

Symptom Quantification for Parkinson's Disease
Mareike Wendebourg

Supervisors: Anders Ericsson & Fredrik Ohlsson, Acreo Swedish ICT AB, Gothenburg, Sweden
Examiner: Tomas McKelvey, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden

Master's Thesis EX017/2016
Department of Signals and Systems
Chalmers University of Technology
SE-41296 Gothenburg
Phone number: +46 31 772 1000

Gothenburg, Sweden 2016

Symptom Quantification for Parkinson's Disease

Mareike Wendebourg
Department of Signals and Systems
Chalmers University of Technology

Abstract

Today, Parkinson's disease is the second most common age related degenerative disorder presenting a complex set of both cognitive and motor symptoms. Medication for the treatment of motor symptoms exists but the development of effective treatment plans without technical aids is tedious. These aids could include sensor systems for the objective evaluation and quantification of symptoms in short- and long-term settings for both the clinical and the home environment. Recently, several studies have shown the feasibility of symptom quantification with the help of gyroscopes or accelerometers. Utilizing such measurements, this work aims at a comparison of several supervised learning methods in order to find the most suitable model structure and therefore the best modeling approach for the quantification of bradykinesia in Parkinson's patients using the example of repeated forearm-rotation, which is a routine motion from Parkinson's test protocols.

The measurement characteristics applied for model development were based on knowledge about the considered movement and motion patterns in Parkinson's disease as well as on insights provided by the literature on other studies concerning the quantification of Parkinson's symptoms. The considered parametric and non-parametric models were developed for a number of sensor subsets and compared in terms of cross-validated mean squared prediction errors obtained for data not utilized during model development. As expected, it was found that when considering only gyroscopes, those measurements of angular velocities around the axis of the forearm were most relevant to model development. Additionally, results generally improved when using principal component analysis for dimension reduction prior to model development. The best results were obtained for local regression when applied to only two characteristics of measurements of angular velocities around the forearm, closely followed by linear regression using the same two characteristics.

Keywords: Symptom quantification, Parkinson's treatment, statistical learning, machine learning, supervised learning, bradykinesia, gyroscope, accelerometer

Acknowledgement

Firstly, I would like to thank my supervisors Anders and Fredrik for their continuous support and enthusiastic interest in my work, for many suggestions, for thoroughly reading and commenting on this thesis, for including me into the MuSyQ project group and especially for always readily taking however much time was needed for in-depth discussions. Additionally, I am thankful to my examiner Tomas for close reading of the thesis as well as for regular meetings. Inside the MuSyQ group, a special thanks goes to Marina for patiently answering all my questions regarding Parkinson's rating scales and the conduction of the study where the data used in this work was obtained, and to Dag for double-checking patient 15's videos. Thanks are due to ACREO for the printing of this thesis and to all its employees for creating the best work environment I have encountered so far. Apart from the obvious suspects, I am grateful to Olaf, Timo and Bine for proof-reading in their spare time. Without Olaf's and Timo's detailed review concerning precise nuances in formulations and mathematical notations or Bine's scrupulous feedback regarding the overall language and the logical implications of any letter from the first to the last page, this thesis would not be what it is. Last but not least, I would like to thank my parents for enabling me to come to where I am and Timo for all his encouragement and emotional support throughout this work.

Mareike Wendebourg, Gothenburg, June 10th, 2016

Contents

Chapter 1

Introduction

For the past decades, interest in technical solutions to medical problems has risen rapidly. Simultaneously, an ever advancing standard of living and improvements in medical care have led to an increase in life expectancy in many parts of the world. The resulting ageing population presents one of the major challenges imposed on health care systems today, and besides further development of rehabilitation techniques, advances in technical aids are needed.

According to Przedborski [24], the most common age related degenerative disorders are Alzheimer's and Parkinson's disease. The latter presents easily measurable but highly variable motor symptoms including e.g. bradykinesia (slowness of movements), postural instability and freezing of gait. The variability of symptom severity complicates the development of effective treatment plans. Imprecise medication is a great concern as overmedication promotes the development of side effects as e.g. involuntary and exaggerated movement known as dyskinesia in absence of voluntary movement, while sufficient medication is necessary to prevent a reduced quality of life as movement restricting symptoms may limit a patient's ability to perform daily tasks independently.

Today, symptoms are evaluated and manually quantified by physicians with help of patient journals and through observation. However, most patients struggle to report symptom severity and nature accurately [5]. This leads to the undesirable situation of a physician's assessment depending mainly on the display of symptoms within a very short time frame.

Additionally, similar symptoms may be rated differently depending on the physician's training and experience. Consequently, the development of each individual patient's treatment plan as well as its adjustment as the disease progresses is cumbersome and time-consuming, since an objective evaluation of patient symptoms is not available. Therefore, the development of technical aids for short- and long-term objective symptom evaluation in both the clinical and the home environment is necessary in order to improve the treatment of each patient as well as the objective assessment of new treatment methods.

According to Marsden and Schachter [17], devices for measuring tremor (involuntary shaking or repeated movement) in Parkinson's patients with help of mechanical and pneumatic components were invented as early as in the 1880s. In the second half of the 20th century, these devices were replaced by electromyographs (EMG) and accelerometers which were utilized for studies on the possibility of objective descriptions of several symptoms present in Parkinson's disease.

Recently, several research groups have successfully quantified symptoms using accelerometers and gyroscopes (compare [1, 2, 3, 5, 9, 21, 26, 31]). These studies illustrate that symptoms can be quantified using measurements of both pre-defined motion protocols

and motions resulting from daily-life activities. However, most studies so far considered only a small number of measurement characteristics referred to as predictors or features and each only a single statistical or machine learning method. Counterexamples include Patel et al. [21] who utilized support vector machines (SVM) with three different kernels as well as Cancela et al. [3] and Tsipouras et al. [31] who compared several non-parametric methods including k-nearest neighbors (KNN), decision trees and neural networks with regard to bradykinesia and/or dyskinesia.

Additionally, Patel et al. [21] quantified tremor. Both bradykinesia and tremor respond well to medication, thus their objective quantification may be used to evaluate medical treatment [16]. However, tremor is not seen as one of the most disabling symptoms of Parkinson's disease [16]. Therefore, this work focuses on the quantification of bradykinesia which describes slowness of both initiation and execution of movements.

While bradykinesia has been quantified by several research groups, this work aims to provide a comparison of the usefulness of various statistical machine learning[1] methods in terms of minimal prediction errors using the example of one pre-defined motion. For this purpose, several parametric and non-parametric models are considered including linear regression, forward selection, backward selection, ridge regression, the lasso, local regression, multivariate adaptive regression splines, KNN, decision trees, support vector machines and principal component analysis as a pre-processing step for linear regression, local regression, smoothing splines, KNN and decision trees.

Parametric models are of interest because they may reflect the underlying mathematical correlation between motion characteristics of movements of Parkinson's patients and the corresponding rating of their symptoms, while non-parametric models can offer greater flexibility since they avoid the explicit assumption of a pre-specified model structure.

Furthermore, a number of different predictor sets are evaluated. Within each predictor set, predictor selection is left to statistical machine learning methods as e.g. forward selection and the lasso which are designed to choose the most useful out of all available predictors. The 120 predictors defined in this work are based on the literature (compare [1, 2, 3, 5, 9, 21, 26, 31, 34]) as well as on knowledge regarding the measured movement and symptoms of Parkinson's disease.

The results are reported in the form of the average mean square error (MSE) where the average MSE is determined as the average of all test errors resulting from comparison of known and predicted symptom scores for data sets which were not used for model development. Due to the limited amount of available data, the test errors are estimated using cross-validation. Additionally, nested cross-validation is used for parameter selection and model assessment for models requiring the selection of a tuning parameter.

The following chapters include some information on Parkinson's disease and the utilized data (chapter 2), an overview of the applied statistical machine learning methods (chapter 3) as well as their implementation (chapter 5), a description of the defined predictors (chapter 4) and a discussion of the obtained results (chapters 6 - 7).

[1]The term "statistical machine learning" describes machine learning using statistical methods and will be used in order to avoid ambiguities connected to the terms "machine learning" and "statistical learning".

Chapter 2

Background

The objective of this work is to find the most suitable model structure and therefore the best modeling approach for the quantification of bradykinesia in Parkinson's patients. This requires insights not only into modeling but also into Parkinson's disease and the data utilized for model development. The mathematical methods applied in this work are described in chapter 3, while an overview of Parkinson's disease and the utilized data is provided in this chapter.

2.1 Parkinson's Disease

Parkinson's disease is a degenerative disorder mainly affecting dopamine generating cells in the substantia nigra located in the mesencephalon (midbrain) which is part of brain stem [24].

According to the Swedish Parkinson's Association (Parkinsonförbundet) about 22.000 people are living with Parkinson's disease in Sweden [20]. Studies from all over the world indicate that about 5 to 26 per 100.000 persons are diagnosed with Parkinson's disease annually [27]. These numbers imply between 363.000 and 1.926.200 new diagnoses worldwide every year.

It has been estimated that $10 - 15\%$ of all Parkinson's cases are inherited, but the "vast majority of PD cases occur sporadically with no obvious cause" [18]. According to McNaught et al. [18], several risk factors as toxins, infectious agents and occupational hazards have been investigated without conclusive results. Some studies found indications of correlations between Parkinson's disease and exposure to pesticides, rural living, well-water drinking, employment in agriculture and repeated head trauma, but according to Schrag [27] neither one of these factors is likely to be the single cause of Parkinson's disease.

Parkinson's symptoms include motoric problems as tremor, rigidity, bradykinesia (slowness of movements), postural instability and freezing of gait (walking) but also dementia, depression, sleeping disorders and autonomic dysfunctions as constipation as well as urinary and sexual dysfunction [22, 24, 32]. Furthermore, sensory symptoms as numbness, burning, tingling, heat, coldness and pain emerge in $40 - 50\%$ of all Parkinson's patients [22]. A diagnosis requires the presence of bradykinesia as well as tremor, postural instability or rigidity for which no other cause can be found [32].

Parkinson's patients face shortened life expectancies although it is not evident whether their increased mortality is due to the disease itself or due to the physical impairment caused by the disease [24, 27]. Untreated or advanced Parkinson's patients are more likely to suffer from aspiration pneumonia (lung inflammation caused by the inhalation of foreign

matter), pressure sores, malnutrition, dehydration and potentially disabling or even fatal falls [24].

Symptomatic treatment includes dopamine-replacing drugs such as levodopa as well as dopamine agonists [30]. These treatments are crucial for the prolongation of patients' employability and independence and may even increase life expectancy [30]. Unfortunately, drug treatments can cause serious side-effects, as for example insomnia, somnolence (sleepiness), nausea, vomiting, dizziness, bradycardia (slow heart rate), hallucinations, psychosis and leg edema (swelling) [30, 22].

Generally, the process of determining the correct medication dosage is extremely difficult due to the fact that Parkinson's symptoms vary greatly depending on fatigue, stress, time of the day, medication, and many other factors, such as bowel movements. Additionally, long-term usage of levodopa induces dyskinesia (involuntary movement) and an increase in symptom fluctuations due to shortening of the intervals in which medication is effective [30].

Usually, medication intake results in periods of diminished motor symptoms described as "on" phases followed by a return of symptoms in so called "off" phases. In advanced Parkinson's disease dyskinesia may be present in both the "on" and the "off" stage, which can alternate rapidly. These symptom variations explain why according to Tolosa and Katzenschlager [30], "It often requires a determined patient and a doctor with patience to achieve significant improvements".

Evidently, patients would benefit immensely from simultaneously more accurate and easily adaptable medication schemes. Readily accessible quantitative long-term information regarding symptom severity and development could aid physicians in the preparation of such schemes. However, although research has shown the feasibility of the utilization of technical tools for the generation of quantitative long-term information, much is still unknown about their optimal set-up. This work aims to contribute to the development of such systems.

2.2 Available Data

The data used in this work was obtained at Uppsala's Akademiska Sjukhuset in a study incorporated in the MuSyQ project (multimodal motor symptoms quantification platform for individualized Parkinson's disease treatment), project number 2014-03727 [33], financed by Vinnova, Sweden's innovation authority.

Measurements were recorded with commercial Shimmer3 sensors [29] fixed to the participants' wrists, ankles and chest with elastic bands in order to enable recording of motions of all limbs as well as the overall posture. Each of these sensor units included three-axial gyroscopes and accelerometers. 19 patients diagnosed with advanced Parkinson's disease participated in the study. All patients provided signed consent prior to participation and all tests were approved by the Uppsala Ethical Review Board.

Each patient completed a set of tasks including some tasks defined in the unified Parkinson's disease rating scale (UPDRS) protocol. One of those tasks was the continuous alternating rotation of the forearm as shown in figure 2.1, which is the task whose wrist sensor measurements will be utilized in this study. This movement corresponds to item 25 in the UPDRS protocol. The execution of all tasks was filmed and the individual symptoms present were evaluated independently by three physicians trained in the field of movement disorders. The resulting scores on individual UPDRS items, the total UPDRS and the treatment response scale (TRS) were available for this work.

According to Kompoliti et al. [13], the UPDRS is "the international gold standard of

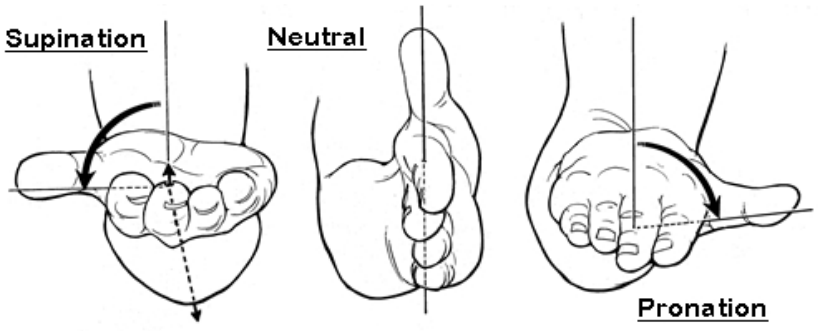

Figure 2.1: Rotation, i.e. supination and pronation, of the forearm [6]

clinical rating scales for PD". It ranges from 0 to 4, where a score of 0 indicates the absence of symptoms while a patient receives a 4 if he or she is not remotely able to execute the task. Parkinson's symptoms are often more severe on one side of the patient's body. In this study, the evaluating physicians were instructed to assign one UPDRS item score for both hands and in case of mismatch of symptoms in both hands, to give more weight to the hand displaying more severe symptoms.

The TRS ranges from -3 to $+3$. Negative scores imply bradykinesic symptoms as derived from the UPDRS, while positive scores describe dyskinetic symptoms and a score of 0 implies the absence of symptoms. However, often bradykinesia and dyskinesia are present simultaneously. In such cases, the TRS score describes the more pronounced symptom.

Both the UPDRS and the TRS are commonly used for clinical evaluation of Parkinson's symptoms and allow the intuitive interpretation of measured symptoms by physicians and patients. Generally, dyskinetic symptoms are less prevalent during voluntary movement as the forearm-rotation considered here. Due to this and the ambiguity of TRS scores for concurrent bradykinetic and dyskinetic symptoms, this work focuses on the quantification of bradykinesia. More specifically, the symptoms present in the wrist sensor measurements will be quantified in terms of and compared to corresponding ratings for item 25 of the UPDRS.

In order to allow the recording of symptoms present in an unmedicated state, i.e. the state of maximal bradykinesia, the study's participants were asked to refrain from medication intake the morning of the study. After each of the tasks specified in the test protocol was completed once, the patients took medication equivalent to 150% of their usual morning dosage to enable measurements of a wide range of symptoms.

Following the medication intake, the test protocol was repeated four times in 20 minute intervals, and then in 30 minute intervals for as long as the patient was willing to continue. The patients could abort the study at any time and resume their normal medication intake. Considering wrist measurements of only each patients' more affected hand, 10 to 15 measurements of approximately 20 seconds of forearm rotation were obtained per patient and sensor.

Both, the total number of wrist measurements recorded for each patient's more affected hand and the number of measurements of each patient's more affected hand corresponding to each UPDRS score are summarized in table 2.1. Here, the provided UPDRS score is the median of the scores assigned by the three physicians. The last row of the table shows the total number of measurements and the number of measurements available for each score on the UPDRS scale for each patient's more affected hand.

The indicated fold numbers were utilized for cross-validation purposes described in

later chapters. Cross-validation is explained in section 3.2.2.

Considering table 2.1, one may observe that the UPDRS scores of the obtained measurements are not evenly distributed. Especially measurements with the most extreme UPDRS scores 0 (no symptoms) and 4 (not able to execute movement) are lacking while many measurements received UPDRS scores of 1 (mild symptom severity) or 2 (moderate symptom severity). Additionally, all measurements which received the highest possible UPDRS score belong to only two different patients.

Table 2.1: Distribution of UPDRS scores over available measurements considering only the hand with more severe symptoms

Patient	Fold	Measurements	Measurements per UPDRS score				
			0	1	2	3	4
1	1	13	2	9	2	0	0
2	2	14	0	12	2	0	0
3	3	11	1	8	2	0	0
4	4	12	2	5	4	1	0
5	5	12	2	6	4	0	0
6	6	13	0	10	3	0	0
7	7	11	0	9	2	0	0
8	8	13	0	7	6	0	0
9	9	14	0	11	3	0	0
10	10	14	0	12	2	0	0
11	11	11	0	0	11	0	0
12	12	15	0	0	15	0	0
13	13	12	0	0	11	1	0
14	14	10	0	0	1	6	3
15	-	14	0	0	14	0	0
16	15	14	0	10	4	0	0
17	16	11	0	0	4	5	2
18	17	14	0	9	5	0	0
19	18	14	0	0	4	10	0
$\sum$ of measurements		242	7	108	99	23	5

For a better understanding of the available UPDRS scores, a confusion matrix showing the physicians' agreement among each other is given in figure 2.2. This matrix illustrates how often the individual physicians agreed with the median score of their ratings which was used as the assumed true UPDRS score for model development.

The numbers in the green boxes on the diagonal indicate the number of ratings for which a physician's score agreed with the median of the three scores assigned to the same measurement, while the numbers in the red boxes show how often a physician assigned the UPDRS score denoted on the vertical axis when the median of the assigned UPDRS

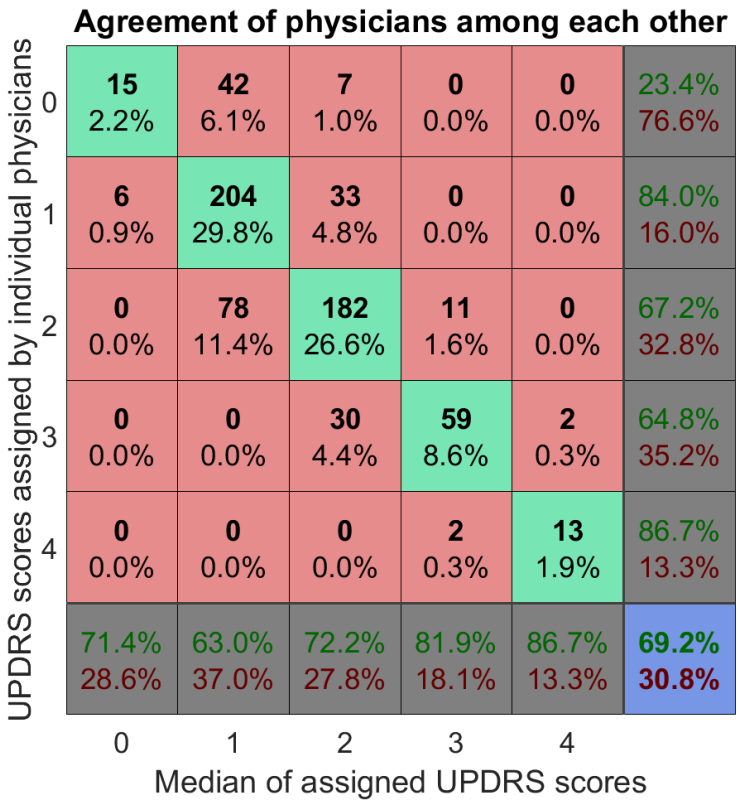

Figure 2.2: Confusion matrix relating the UPDRS scores assigned by each individual physician to the median of the UPDRS scores assigned by all three physicians

scores for the same measurement corresponded to the score given on the horizontal axis.

For example, for seven measurements the median of the UPDRS scores assigned by the three physicians was 0. However, out of all 21 ratings assigned to these seven measurements, 15 were equal to 0 while six were equal to 1. Consequently, an assumed true UPDRS score of 0 is supported by 71.4% of all physicians' ratings. On the other hand, an UPDRS score of 0 was assigned 64 times in total. However, in 76.6% of all cases when one physician assigned an UPDRS score of 0, the other two physicians assigned a higher score.

Of course, at least one third of the individually assigned UPDRS scores are equal to the median UPDRS score by design since this median score is defined as one of the three UPDRS scores assigned to each measurement. Nonetheless, one may observe that the physicians' assessment frequently differed by one step on the UPDRS scale but seldom by more. Furthermore, the physicians disagreed most often in the lower range of the UPDRS scale.

Overall, only 69.2% of all ratings corresponded to the UPDRS score that was used for model development. This number illustrates the difficulties of manual symptom quantification.

Chapter 3

Statistical Machine Learning

Statistical machine learning describes a number of methods useful for understanding data. These include supervised learning methods which, according to James et al. [10], aim at the prediction of some outcome based on knowledge gained from similar observations whose outcomes are known, where each observation is described by p characteristics referred to as predictors or features. In other words, given n outcomes or responses $Y := \{y_1, \ldots, y_n\}$ and corresponding observations $X := \{x_1, ..., x_n\}$, each represented by p predictors such that $x_i = (x_{i1}, ..., x_{ip})$, one strives to estimate the mathematical relationship between X and Y in order to enable the prediction of unknown responses corresponding to new observations. This estimation process includes both the formulation of a model, also known as training, and the validation of the obtained model.

Once a model has been developed, a measure for the models quality is given by the mean square error (MSE), where x_i and y_i are the observations and responses of a new data set while $\hat{f}$ describes the previously determined model:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{f}\left(x_i\right) \right)^2 . \tag{3.1}$$

In this work, the median of the ratings assigned by the three physician individually was used as the true response y_i.

3.1 Model Formulation

Today, a vast number of approaches for model formulation are available. These include models for continuous as well as discrete response values, usually described as regression and classification models respectively. Regression models may be utilized for the prediction of discrete response values as well, if the response values have a natural ordering and thus, the continuous responses may be rounded in order to obtain discrete response values.

The data available has response values 0, 1, 2, 3 and 4 corresponding to UPDRS scores, where an increase of the response value implies an increase in symptom severity. Therefore, both regression and classification methods may be applied.

In this work, linear regression, forward selection, backward selection, ridge regression, the lasso, local regression, smoothing splines, multivariate adaptive regression splines, k-nearest neighbors, decision trees and support vector machines were applied. Furthermore, principal component analysis was utilized for dimension reduction as a pre-processing step for some methods including linear regression, local regression, smoothing splines, k-nearest neighbors and decision trees. Each of the mentioned methods is explained in detail in the following utilizing information given by James et al. [10] and Hastie et al. [7].

3.1.1 Linear Regression

Linear regression is the most basic statistical machine learning method which assumes a linear relationship between the observations X and the corresponding responses Y:

$$\begin{aligned} y_i &\approx \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} \\ &\approx \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}. \end{aligned} \tag{3.2}$$

It is insensitive to scaling of predictors since the model coefficients $\beta_0, \ldots, \beta_p$ are adjusted accordingly.

The optimal model coefficients in the least-squares sense are estimated as the coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$ minimizing the residual sum of squares (RSS). The RSS describes the squared prediction error of the model for training data, where $\hat{y}_i$ denotes the response y_i as estimated by the linear regression model:

$$\begin{aligned} RSS &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip} \right)^2. \end{aligned} \tag{3.3}$$

The RSS defines a training error, i.e. a measure of the error resulting from prediction of responses for data that was utilized for development or so called training of the prediction model. This implies that the RSS will probably be considerably smaller than the error resulting from application of the model to previously unknown data.

Nonetheless, the minimization of the RSS provides the best linear fit for the training data, i.e. data used during model development, and may be utilized for comparison of how well two models obtained using the same training data, model structure, tuning parameter (if applicable) and number of predictors are able to fit to the data.

3.1.2 Forward Selection

One disadvantage of the linear regression approach is that it takes into account all p provided predictors. However, in reality some of the p predictors may not actually be related to the response. These predictors will act as noise in the obtained model. Therefore, it would often be preferable to have a model which selects the predictors most relevant for the response. One may attempt to find the best subset of predictors by testing all 2^p possible combinations, but this approach easily becomes computationally infeasible as the number of predictors p increases.

A more efficient alternative is provided by forward selection, which adds predictors to the model one by one. However, one should be aware that forward selection is a greedy approach, i.e. it always chooses the currently best option without planning ahead and does not reconsider the value of choices of previous iterations. Consequently, forward selection does not guarantee to find the optimal model out of all 2^p possible models. The procedure is the following:

1. Determine the null model $\mathcal{M}_0$ containing only the intercept $\hat{\beta}_0$
2. For each number of predictors $k = 1, \ldots, p$:

(a) Find all $o = p-k+1$ models $\mathcal{M}_{ko}$ that result when each of the remaining $p-k+1$ predictors is added to the predictors used in the previously determined model $\mathcal{M}_{k-1}$

(b) From all models $\mathcal{M}_{ko}$, select the one with the smallest RSS as the best model for k predictors and define it as $\mathcal{M}_k$

3. Choose the best model from all models $\mathcal{M}_0, \ldots, \mathcal{M}_p$ with help of the cross-validated test error (explained in section 3.2.2)

According to James et al. [10], forward selection is suitable for model development in high-dimensional predictor spaces and can even be applied when the number of predictors exceeds the number of observations with known responses available for model development. Furthermore, it is possible to consider only models utilizing a smaller number of predictors than available, hereby limiting the number of iterations of the shown procedure.

3.1.3 Backward Selection

A similar idea as for forward selection is employed in backward selection, but instead of adding the predictor with the greatest positive impact to the model, one removes the predictor with the least positive impact. Initially, backward selection considers the linear regression model containing all predictors, referred to as $\mathcal{M}_p$. The subsequent procedure is the following:

1. For $k = 1, \ldots, p$:

   (a) Find all $o = p - k + 1$ models $\mathcal{M}_{ko}$ that result when each of the remaining $p - k + 1$ predictors is removed from the previously determined model $\mathcal{M}_{p-k+1}$

   (b) From all models $\mathcal{M}_{ko}$, select the one with the smallest RSS as the best model for $p - k + 1$ predictors and define it as $\mathcal{M}_{p-k}$

   (c) Remove the predictor not present in $\mathcal{M}_{p-k}$ from the predictor set

2. Choose the best model from all models $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using the cross-validated test error as described in section 3.2.2

As forward selection, backward selection is a greedy approach which does not guarantee to deliver the optimal model but is suitable for regression in high-dimensional settings. However, contrary to forward selection, backward selection requires the number of available training observations to be greater than the number of predictors in order to enable the development of the full model $\mathcal{M}_p$.

3.1.4 Ridge Regression

Instead of selecting specific predictors as in forward and backward selection, one may alter their relative importance. Modification of linear regression through the addition of a shrinkage penalty to the objective function given by the RSS as defined in equation (3.3) results in weighting of the coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$ relative to each other while shrinking them towards zero.

Contrary to linear regression, ridge regression requires the standardization of the predictors to a comparable scale prior to model development. This holds true for most modeling methods which rely on either the distance between observations in the predictor space or penalize the magnitude of model coefficients.

The quantity minimized for ridge regression is the following:

$$RSS + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2 = \sum_{i=1}^{n} \Big(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip} \Big)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2$$
$$= \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2. \tag{3.4}$$

Here, the shrinkage penalty $\lambda \sum_{j=1}^{p} \beta_j^2$ promotes the selection of small coefficients which are desirable as they decrease the variance of the resulting predictions at the price of a small increase in bias. These effects stem from a reduction in flexibility of the ridge regression model in comparison to the linear regression model due to the constraints imposed by the shrinkage penalty. According to James et al. [10], less flexible models are more useful in high-dimensional predictor spaces.

The shrinkage penalty's impact is determined by the tuning parameter λ. One may estimate the value of the tuning parameter resulting in the optimal trade-off between variance reduction and bias enlargement with help of the nested cross-validated test error as described in section 3.2.3.

3.1.5 The Lasso

Contrary to ridge regression, the lasso selects features by shrinking some coefficients to exactly zero for sufficiently large values of the tuning parameter λ due to its more restrictive shrinkage penalty of the form $\lambda \sum_{j=1}^{p} |\beta_j|$. Hereby, it can reduce the predictor space and increase model interpretability. Furthermore, its ability to exclude predictors means that the lasso is useful for model development in high-dimensional settings [10].

The lasso utilizes the following minimization criterion for estimation of coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$:

$$RSS + \lambda \sum_{j=1}^{p} |\hat{\beta}_j| = \sum_{i=1}^{n} \Big(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip} \Big)^2 + \lambda \sum_{j=1}^{p} |\hat{\beta}_j|$$
$$= \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\hat{\beta}_j| \tag{3.5}$$

The tuning parameter λ can be optimized using nested cross-validation as explained in section 3.2.3.

3.1.6 Local Regression

One method for obtaining a non-linear approximation of the mathematical relationship between observations and responses is given by local regression.

Given a new observation x_0, a non-zero weight $K_{i0} = K(x_i, x_0)$ is assigned to each known observation x_i within some pre-defined distance l from the new observation. These weights may be uniformly distributed within some radius l and zero outside the radius l, or weights may be assigned to smoothly decrease with increasing distance from the new observation. The optimal area to consider for a certain model may be estimated with help of nested cross-validation as explained in section 3.2.3.

Once the weight of each utilized observation has been determined, a weighted linear regression problem is solved through minimization of the following criterion:

$$\sum_{i=1}^{n} K_{i0} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 . \tag{3.6}$$

Then, the response of the new observation can be estimated with help of the obtained linear function as $\hat{f}(x_0) = \hat{\beta}_0 + \sum_{j=1}^{p} \beta_j x_{0j}$. However, note that all training data is needed for the prediction of responses of unknown observations with help of a local regression model.

One may solve the minimization problem for a quadratic form of the regression function given in equation (3.2) as well. Furthermore, local regression is suitable for unevenly distributed data sets since distant data points have less or no effect on other parts of the model even if their occurrence in some region of the predictor space is prevalent. However, local regression may not perform well for many more than three or four predictors per observation [10].

3.1.7 Regression and Smoothing Splines

Splines describe a modeling approach involving the definition of a number of interconnected non-overlapping d-degree polynomial functions for different regions of the predictor space, hereby allowing a flexible non-linear model. Several approaches for the derivation of splines exist, out of which regression and smoothing splines are explained here.

Smoothing splines result from an adapted optimization criterion, namely the minimization of

$$RSS + \lambda J(f) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda J(f) \tag{3.7}$$

where λ denotes a tuning parameter and $f(x_i)$ is some function to be determined. The optimal function $f(x_i)$ found with help of this criterion will be a piecewise cubic polynomial function which is continuous in its first and second derivatives at all knots ξ located at each observation x_i.

The penalty function $J(f)$ differs depending on the dimensions of the predictor space [7]. For a single predictor x_i,

$$J(f) = \int \left(f''(x_i) \right)^2 \, \mathrm{d}x_i, \tag{3.8}$$

while in two dimensions, i.e. for $x_i = (x_{i1}, x_{i2})$,

$$J(f) = \iint_{\mathbb{R}^2} \left[\left(\frac{\partial^2 f(x_i)}{\partial x_{i1}^2} \right)^2 + 2 \left(\frac{\partial^2 f(x_i)}{\partial x_{i1} \partial x_{i2}} \right)^2 + \left(\frac{\partial^2 f(x_i)}{\partial x_{i2}^2} \right)^2 \right] \mathrm{d}x_{i1} \mathrm{d}x_{i2}. \tag{3.9}$$

According to Hastie et al. [7], this optimization criterion does not scale well with dimensionality of the predictor space and therefore, in this work, smoothing splines were only used in combination with principal component analysis as described in section 3.1.12.

A different approach is given by regression splines. The idea fundamental to regression splines is the non-linear transformation of predictors to a number of basis functions which are then used to develop a linear model. This linear model can be easily constructed

utilizing a least-squares approach while the resulting model is non-linear in terms of the original predictors.

The model functions of a regression spline are continuous in their derivatives up to degree $d-1$ at the points in the predictor space where two functions meet. These points are called knots ξ. Considering K knots and some polynomial basis functions $b(x_i)$, a regression spline can be described by

$$y_i \approx \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_{K+d} b_{K+d}(x_i). \tag{3.10}$$

Here, most basis functions are described by so called truncated power functions, i.e. these basis functions are equal to zero in some parts of the predictor space. For a single predictor x, i.e. a one-dimensional predictor space, the set of basis functions is defined by

$$b_b(x) = x^b, \qquad b = 1, \dots, d \tag{3.11}$$

$$\begin{aligned} b_{k+d}(x) &= (x - \xi_k)_+^d, \qquad k = 1, \dots, K \\ &:= \begin{cases} (x - \xi_k)^d, & x > \xi, \\ 0, & x \leq \xi, \end{cases} \end{aligned} \tag{3.12}$$

where $\xi \in \mathbb{R}$ are knots placed at pre-defined locations in the predictor space while x describes all points on the predictor axis.

Knots may be placed anywhere in the predictor space but usually a uniform distribution is chosen. The number of knots under consideration can be determined with help of nested cross-validation as described in section 3.2.3.

As for linear regression models described in section 3.1.1, the model coefficients $\beta_0, \dots, \beta_{K+d}$ of the regression spline may be estimated through minimization of the RSS defined by

$$\begin{aligned} RSS &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} \Big(y_i - \hat{\beta}_0 - \hat{\beta}_1 b_1(x_i) - \hat{\beta}_2 b_2(x_i) - ... - \hat{\beta}_{K+d} b_{K+d}(x_i) \Big)^2. \end{aligned} \tag{3.13}$$

According to Hastie et al. [7], the number of basis functions grows exponentially with increasing dimension of the predictor space. Therefore, one should employ a greedy alternative as for example multivariate adaptive regression splines (explained in section 3.1.8) for high-dimensional predictor spaces as the ones considered here. Consequently, multivariate adaptive regression splines were applied instead of regression splines in this work.

3.1.8 Multivariate Adaptive Regression Splines

Multivariate adaptive regression splines (MARS) produce piecewise linear models which, contrary to simple regression splines, are suitable for high-dimensional predictor spaces.

For this purpose, MARS creates a knot ξ_i at the location of each observation x_i. Then, a so called reflected pair of basis functions is created at each knot, composed of one basis function which is zero on the left hand side of the knot and one basis function which is zero

on the right hand side of the knot with respect to the predictor axis under consideration:

$$\begin{aligned} b_l(x) &= (x_i - x)_+, \\ &= (\xi_i - x)_+, \\ &= \begin{cases} \xi_i - x, & x > \xi_i, \\ 0, & x \leq \xi_i, \end{cases} \end{aligned} \tag{3.14}$$

$$\begin{aligned} b_r(x) &= (x - x_i)_+, \\ &= (x - \xi_i)_+, \\ &= \begin{cases} x - \xi_i, & x < \xi_i, \\ 0, & x \geq \xi_i. \end{cases} \end{aligned} \tag{3.15}$$

Here, x denotes values on the axis of each of the p predictors x_p under consideration, leading to a total number of $2np$ basis functions.

After construction of the basis functions and an initial model with a constant value of 1, MARS iteratively selects the reflected pair of basis functions which results in the largest decrease of training MSE for multiplication with the model defined in the previous iteration. This multiplication corresponds to the addition of basis functions to the model of the previous iteration. The training MSE results if the definition of the MSE as given in equation (3.1) is applied to the training data.

The resulting model will usually overfit the available data, i.e. it will resemble the data used for model development too closely to be applicable to new data. Therefore, another iterative procedure is employed to delete those basis functions from the model whose deletion increases the RSS the least until some stop criterion is reached. One possible stop criterion is the specification of a number of basis functions to be used in model after the removal process. The optimal number of basis functions used in the final model may be determined with help of nested cross-validation as described in section 3.2.3.

3.1.9 K-Nearest Neighbors

The application of k-nearest neighbors (KNN) is based on the assumption that the response of an observation is likely to resemble the responses of observations in close proximity in the p-dimensional predictor space P. In the classification case, i.e. when only discrete response values are considered, the estimation of this conditional probability assumption can be described mathematically as

$$\Pr\left(y_i = r | x_i = x_0\right) = \frac{1}{K} \sum_{l \in \mathcal{N}_0} I\left(y_l = r\right). \tag{3.16}$$

where $\Pr\left(y_i = r | x_i = x_0\right)$ denotes the probability that the response y_i has some value r, if the observation x_i has the characteristics x_0, i.e. it is located at x_0 in the predictor space. Here, $\mathcal{N}_0$ denotes the neighborhood surrounding the point $x_0 \in P$ which contains K other observations. The indicator variable $I\left(y_l = r\right)$ has a value of 1 if the response y_l has the value r, otherwise it is 0. In this work, the response variables represent scores on the UPDRS and therefore $r \in \{0, 1, 2, 3, 4\}$.

Contrary to the parametric methods discussed in sections 3.1.1 to 3.1.5, no underlying structure is assumed. Instead, all training data must be available for the prediction of the responses for new observations. Then, given some new observation x_{new}, KNN identifies the K training observations in closest proximity to the new observation and estimates the probability of the response y_{new} of the new observation x_{new} to have each of the response

values r:

$$\hat{y}_i = \arg \max_{r=0,\ldots,4} \Pr\left(y_i = r | x_i = x_0\right) \tag{3.17}$$

According to James et al. [10], KNN performs well when many observations, each described by only a few predictors, are available. However, in the case of high-dimensional predictor spaces, an observation's nearest neighbors may not be in close proximity and consequently, KNN's basic assumption looses meaningfulness. This reflects a phenomenon known as the curse of dimensionality and applies to other neighborhood based methods as e.g. local regression described in 3.1.6 as well.

In this work, KNN was applied to both a high-dimensional predictor space and a predictor space composed of the first two principal components (compare section 3.1.12). Nested cross-validation as described in section 3.2.3 was used for selecting the optimal number of neighbors and for model assessment.

3.1.10 Decision Trees

Tree based methods aim to stratify the predictor space into rectangular subsets corresponding to the possible responses. This is achieved through minimization of the residual sum of squares (RSS) as defined for decision trees which is given by

$$\sum_{l=1}^{L} \sum_{i \in R_l} \left(y_i - \bar{y}_{R_l}\right)^2 . \tag{3.18}$$

Here, $\bar{y}_{R_l}$ describes the mean value of all responses located inside the lth region R. Note how, in comparison with the RSS defined for linear models in section 3.1.1, this definition of the RSS is adapted to the classification setting.

However, a greedy approach called recursive binary splitting is employed since consideration of all possible subsets R would be computationally infeasible. Using this approach, the predictor space is iteratively divided into subsets R_l called branches, where in each step the subset division resulting in the largest decrease in RSS is chosen. The response of a new observation is predicted as the most commonly occurring response for the training observations located in the same region as the new observation.

Unfortunately, the best number of divisions is unknown and consequently, decision trees tend to overfit the available data. This issue should not be resolved by limiting the minimum decrease in RSS per division, since some beneficial divisions may require the presence of some seemingly unimportant subset separations before they can be considered.

Therefore, a better approach is to first grow a large tree T_0 where each region R_l contains only a handful of observations and to subsequently remove or prune any outside branches R_m lacking a reasonably strong effect on the RSS. This may be achieved with help of the tuning parameter λ and the following minimization criterion:

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} \left(y_i - \bar{y}_{R_m}\right)^2 + \lambda |T|. \tag{3.19}$$

Here, $|T|$ denotes the number of final nodes of the subtree $T \subset T_0$ while $\bar{y}_{R_m}$ describes the mean of the available responses of the observations in the region R_m. The optimal value for the tuning parameter λ may be estimated using nested cross-validation as explained in section 3.2.3.

According to James et al. [10], trees present easily interpretable prediction models which may perform better than parametric models when applied to data exhibiting complex highly non-linear relationships. However, their predictions are often less accurate and may change drastically for small changes in the training data.

3.1.11 Support Vector Machines

Support vector machines (SVM) are based on the idea that the most likely boundary to distinctively define two response classes will usually cause some cases of misclassification and that the optimal decision boundary may be predicted with some confidence through consideration of those potentially misclassified observations as well as observations in close proximity of the decision boundary.

The hyperplane defining the decision boundary for a binary classification problem with responses $y_i \in \{-1, 1\}$ can be found using the optimization criterion

$$\begin{aligned} \max M \qquad & \text{s.t. } y_i \left(\beta_0 + \sum_{j=1}^{p} \sum_{k=1}^{d} \beta_{jk} x_{ij}^k \right) \geq M(1 - \epsilon_i), \\ & \sum_{j=1}^{p} \sum_{k=1}^{d} \beta_{jk}^2 = 1, \\ & \sum_{i=1}^{n} \epsilon_i \leq C, \\ & \epsilon_i \geq 0 \end{aligned} \tag{3.20}$$

where M describes the width of an area known as the margin around the resulting decision boundary while C denotes a tuning parameter limiting the number and severity of margin violations by training observations. ϵ_i are so called slack variables which allow some observations to fall into the margin. Those observations form the set of support observations.

According to James et al. [10], one may replace the polynomial function

$$f(x_i) = \beta_0 + \sum_{j=1}^{p} \sum_{k=1}^{d} \beta_{jk} x_{ij}^k \tag{3.21}$$

in the maximization problem described in equation 3.20 by

$$f(x_i) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(x_i, x_{i'}). \tag{3.22}$$

Here, the optimization variables $\beta_{11}, \ldots, \beta_{pd}$ are replaced by combinations of α_i and $x_{i'j}$ where $x_{i'j}$ denotes all training observations x_{ij}, i.e. $x_i x_{i'}$ describes the sum of the products of an observation with all other observations and with itself.

K is referred to as a kernel and may take various forms. For example, a linear kernel is defined by

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}, \tag{3.23}$$

a polynomial kernel by

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^2, \tag{3.24}$$

and a radial kernel by

$$K(x_i, x_{i'}) = \mathrm{e}^{-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}. \tag{3.25}$$

Once the optimal parameters β_0 and α_i have been estimated, one may utilize the resulting function

$$f(x_0) = \hat{\beta}_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i K(x_0, x_i). \tag{3.26}$$

for the prediction of responses for new observations x_0, where $\mathcal{S}$ is the set of indices of the support observations. Note that $\hat{\alpha}_i$ will be zero for all training observations which are not located inside the margin.

However, in this work, the response variable y_i is not binary. Therefore, a so called one-versus-all approach is applied, where one SVM is constructed for each pair of response classes. Eventually, the response of a new observation is predicted as the response assigned most often after consideration of each of the constructed SVMs.

The optimal value of the tuning parameter C can be estimated with help of the cross-validated test error described in section 3.2.3. If the two classes, i.e. observations with the response -1 and observations with the response 1, can be separated perfectly, the support vector machine will aim to find the decision boundary with the largest distance to those observations of both classes which are closest to each other.

3.1.12 Principal Component Analysis

In many settings, a large number of predictors is available and one may need to reduce the predictor space's dimensions. One method for dimension reduction is given by principal component analysis which is based on the assumption that the spread of observations in the p-dimensional predictor space contains information about the relative importance of predictors for the corresponding response values. This is not necessarily the case as responses are not taken into account, but according to James et al. [10] "it often turns out to be a reasonable enough approximation to give good results".

One may utilize the assumed information to reduce the predictor space's dimension by considering each observation's projection onto the axes of largest spread instead of using the original predictor. The predictors for each observation can be projected onto the principal components z_{im} as follows:

$$z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}. \tag{3.27}$$

Here, $m = 1, \ldots, M$ where $M < p$ defines the number of principal components or axes considered while $j = 1, \ldots, p$ represents the predictors for each observation.

The transformation factors ϕ_{jm} can be determined through consideration of the spread of the observations along every axis in the predictor space:

$$\max \mathrm{Var} \left(\sum_{j=1}^{p} \phi_{jm} \sum_{i=1}^{n} \left(x_{ij} - \frac{\sum_{i=1}^{n} x_{ij}}{n} \right) \right) \qquad \text{s.t. } \|\phi_m\| = \sum_{j=1}^{p} \phi_{jm} = 1 \tag{3.28}$$

Then, one may fit models to the principal components instead of to all predictors. This is especially useful for the development of models which do not handle high-dimensional predictor spaces well.

The optimal number of principal components M can be approximated with help of nested cross-validation as described in section 3.2.3. Other approaches include the utilization of all principal components needed to reflect some specified fraction of the overall variance present in the available data set.

In this work, the first two principal components will be utilized as predictors for linear regression, local regression, smoothing splines, KNN and decision trees as described in sections 3.1.1, 3.1.6, 3.1.7, 3.1.9 and 3.1.10. Furthermore, a number of principal components chosen with help of nested cross-validation as explained in section 3.2.3 will be used for the development of a linear regression model.

3.2 Validation

In contrast to models obtained for e.g. system identification, statistical machine learning models are intended to be used on observations of similar but not necessarily identical systems. Unfortunately, the variation between different data sets is usually unknown. Therefore, testing of the obtained model on independent observations with known responses is crucial to ensure that a model does not only describe the relationship of observations and responses of the data set used for model formulation, but that the model is applicable to new data sets as well.

The simplest validation method is known as the validation set approach. In this work, the concepts of cross-validation and nested cross-validation, each explained in the following, were used.

3.2.1 Validation Set Approach

Often, independent observations are not available for testing of the model. A simple solution to this problem is the division of the original data set into a training set and a test set prior to model development. Only the training set is utilized for model building. Then, the obtained model can be used to predict the response of each of the test set observations. The test error of the model may be estimated as the validation or test set error with help of equation (3.1).

However, a smaller training set may decrease the quality of the model since less data is available for training while the test set has to be sufficiently large to represent potential future data. Furthermore, the test MSE may be highly variable depending on the choice of training and test set. Consequently, the validation set approach performs well if a large amount of data is available.

3.2.2 Cross-Validation

Cross-validation describes the repeated use of the validation set approach on various divisions of the original data set into different training and test data sets. In each iteration, one subset of data is used as test data while the remaining data is utilized for model development as depicted in figure 3.1. The subsets of data which are combined to form a large training set as well as the subset forming the test set are referred to as folds.

The number of cross-validation iterations corresponds to the number of subsets into which the original data is divided. Therefore, one may speak of F-fold cross-validation where F denotes the number of folds into which the data has been divided.

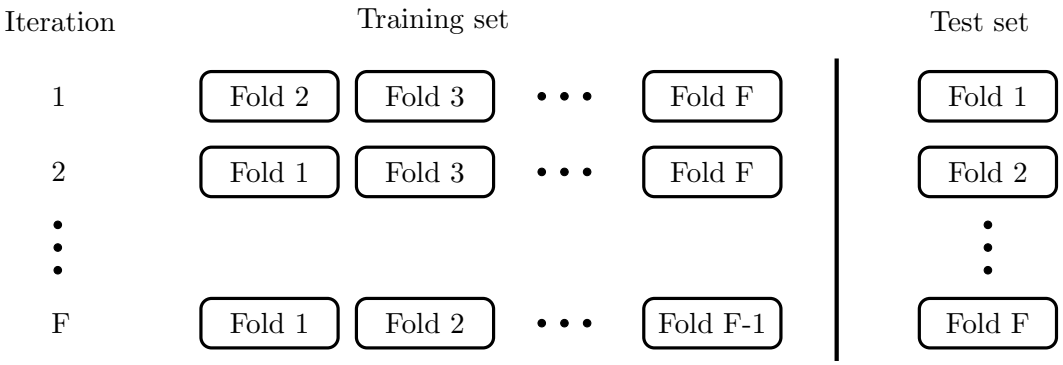

Figure 3.1: Composition of training and test data for each iteration of cross-validation

The cross-validation test error is given by the average MSE of all repetitions. The procedure can be described in more detail as follows:

1. Divide data set into F folds
2. For $f = 1, ..., F$:
   (a) Define f-th fold as test set $TEST_f$ and all other folds as training set TR_f
   (b) Fit a model to the training set TR_f
   (c) Use model to predict responses of the test set $TEST_f$
   (d) Calculate MSE of predicted responses and define it as MSE_f
3. Determine the cross-validated MSE as $MSE = \frac{1}{F} \sum_{f=1}^{F} MSE_f$

In contrast to the validation set approach, cross-validation is suitable for validation of models obtained from small data sets as well. However, it does not provide one single model with a corresponding MSE. Instead, the average MSE gives an estimate of the MSE that can be expected if the model is tested on new data after it was trained on all available data.

3.2.3 Nested Cross-Validation

Many statistical learning methods require the selection of tuning parameters used during model development. In order to find a reasonably good value for the tuning parameter, one may utilize cross-validation for comparison of models resulting from various possible values of the tuning parameter. The choice of tuning parameter resulting in the smallest average MSE approximates the optimal value of the tuning parameter.

Once the optimal tuning parameter has been estimated, it can be utilized for model development. However, the model determined using the estimated optimal tuning parameter must be validated with help of the validation set approach or cross-validation as well.

In case cross-validation is chosen, two cross-validation loops result, where one is encapsulated by the other, i.e. a cross-validation procedure for tuning parameter selection is applied to each of the outer loop training sets. Once a tuning parameter has been chosen, it is utilized for model development using the outer loop training set for which it

was determined. Then, the MSE resulting from application of the developed model to the outer loop test set is calculated.

This procedure is repeated for each iteration of the outer cross-validation loop as summarized in the following:

1. Divide data set into F folds
2. For $f = 1, ..., F$:
   (a) Define f-th fold as outside test set $TEST_{out,f}$ and all other folds as outside training set $TR_{out,f}$
   (b) For $f_{in} = 0, ..., F-2$ and the outside training set $TR_{out,f}$:
      i. Define $TEST_{in,k}$ where $k = (f + f_i \mod F) + 1$ as inside test set and all sets currently not defined as inside or outside test set as inside training set $TR_{in,k}$
      ii. For each discrete tuning parameter value λ under consideration:
         A. Build a model using the inside training set $TR_{in,k}$
         B. Use the model to predict responses of inside test set $TEST_{in,k}$
         C. Calculate the MSE of the predicted responses and define it as $MSE_{k,\lambda}$
   (c) Calculate the average MSE for each tuning parameter λ and define it as $MSE_{ave,\lambda}$
   (d) Choose tuning parameter corresponding to smallest average MSE and define it as λ_f^*
   (e) Build model using the chosen tuning parameter λ^* and the outside training set $TR_{out,f}$
   (f) Use model to predict responses of outside test set $TEST_{out,f}$
   (g) Calculate the MSE of the predicted responses and define it as MSE_f
3. Calculate the average MSE over all folds as the average of MSE_f and define it as the test error MSE_{ave}
4. Calculate an estimate of the optimal value for the tuning parameter λ as the average of the chosen tuning parameters λ_f^* over all folds

The use of cross-validation for parameter selection and model assessment as well as the application of nested cross-validation for parameter selection followed by model assessment has been described by Krstajic et al. [14]. Furthermore, Krstajic et al. [14] illustrate the benefits of repeated nested cross-validation where the folds are composed of different measurements in each repetition.

Chapter 4

Predictors

Statistical machine learning methods relate observations X to corresponding responses Y, with the aim to apply the determined relations to the prediction of unknown responses Y_{new} for comparable sets of observations X_{new}. However, in many settings the observations may be described by a large number of characteristics. In such situations, the prior definition of characteristics to consider in model development becomes a necessity.

The choice of characteristics applied, also known as features or predictors, is crucial for the success of any statistical learning method. Choosing predictors which are not related to the response under investigation will not yield a useful model.

Simultaneously, it is not advisable to choose characteristics for their great correlation to the response since this strong relationship may only be present in the current data set. In other words, choosing predictors whose strong correlation to the response of the given data is known, will lead to a seemingly well-fitting model but its application to a new data set might give poor results, i.e. the model may have been overfitted.

The discrepancy between model fits arises due to the bias introduced through the choice of predictors. In other words, the obtained model is too well-fit for the application to other datasets. For statistical machine learning purposes, a less well-fit model is preferable in order to enable the model's application to comparable data sets.

Similarly, even consideration of the obtained observations can jeopardize objectiveness in predictor selection since one will be tempted to favor features that appear promising to the human eye. The only way to ensure that knowledge of the data at hand is not applied for predictor selection is to not consult the data before selecting predictors. However, when using statistical machine learning one has no choice but to select predictors manually. Instead of utilizing knowledge of the collected data, one may use general knowledge of the observed events in order to formulate potentially relevant predictors.

Most likely, the number of formulated predictors may exceed the number of truly relevant features. The number of predictors used in the final model can be determined with help of model reduction techniques. Additionally, one might hypothesize several predictor sets to be tested on various model types, while keeping in mind that a large number of options increases the probability that a set of predictors combined with a certain model type may produce good results purely by chance. Therefore, it is advisable to not only cross-validate the computed models, but to employ another previously unseen set of test data for comparison of the obtained models. Unfortunately, this was not possible in this work due to a lack of available data. However, for the number of considered models it is unlikely that a low MSE will be obtained by chance.

The considered predictors and predictor set hypotheses as well as necessary pre-processing of the available measurement data are presented in the following.

4.1 Pre-Processing of Measurement Data

The sensors recording the utilized data were started before each patient's first conduction of the test protocol and shut off when a patient decided to return to their usual medication intake. The time intervals of interest were generously extracted by hand with consideration of starting times specified in the study protocol.

Then, the relevant time interval within each extracted time series was determined through selection of a threshold value related to each measurement's largest amplitude as well as visual inspection. Finally, the utilized measurements were defined as the data present within as well as 2 seconds before and after the relevant time interval.

After the raw measurement data was available in the desired format, further pre-processing was needed to remove measurement characteristics that may conceal the characteristics of interest for this study. For instance, the accelerometer measurements not only reflect a person's movement but also the accelerometer's orientation in space. Hence, a motionless accelerometer provides non-zero measurement components directed away from the center of the earth. Additionally, both accelerometer and gyroscope measurements may present drift over time. These issues effect only very low frequency components.

On the other hand, intentional human movement seldom exceeds frequencies of 3.3Hz (compare [25]). Bradykinesia by definition implies slower than normal movement and thus, it is well represented in frequencies below 3.3 Hz. According to Salarian et al. [26], frequencies between approximately 4 and 6 Hz are common for tremor, another symptom of Parkinson's disease which is not investigated in this work.

Therefore, an IIR Butterworth bandpass filter with a lower passband frequency of 0.75 Hz and an upper passband frequency of 3 Hz was used to filter the measurement data forward and backward with help of MATLAB's `filtfilt` function before predictors were calculated. The lower and upper stopband frequencies were chosen as 0.25 Hz and 3.5 Hz respectively, each with a stopband attenuation of 3 dB. The effect of this filter on the measurements is by way of example shown in figures 4.1 and 4.2.

4.2 Defined Predictors

A number of predictors were chosen based on knowledge about motion patterns in Parkinson's disease and the experimental set-up as well as on insights provided by the literature on other studies concerning the quantification of Parkinson's symptoms (compare [1, 2, 3, 5, 9, 21, 26, 31, 34]). Some of the chosen predictors may measure similar characteristics of the observed motion. However, one can only guess the usefulness of each of these predictors and therefore, the choice was mostly left to statistical machine learning methods aiming at model reduction as e.g. forward selection, backward selection and the lasso described in chapter 3. Furthermore, principal component analysis was used for dimension reduction as a pre-processing step for several other methods.

The predictors x_p were derived from measurements $x[n]$[1] obtained from a continuous motion signal $x(t)$ where t denotes the time. The continuous signal was sampled at a sampling frequency of $f_s = 102.4\text{Hz}$ ($T_s = 98\text{ms}$), which according to Oppenheim and Schafer [19] results in the discrete time signal

$$x[n] = x\left(\frac{n}{f_s}\right) = x\left(nT_s\right) = x\left(n \cdot 98\text{ms}\right), \qquad n = 1, 2, \ldots, N. \tag{4.1}$$

[1] In this work, all discrete-time signals are denoted with square brackets while continuous-time signals are described using parentheses.

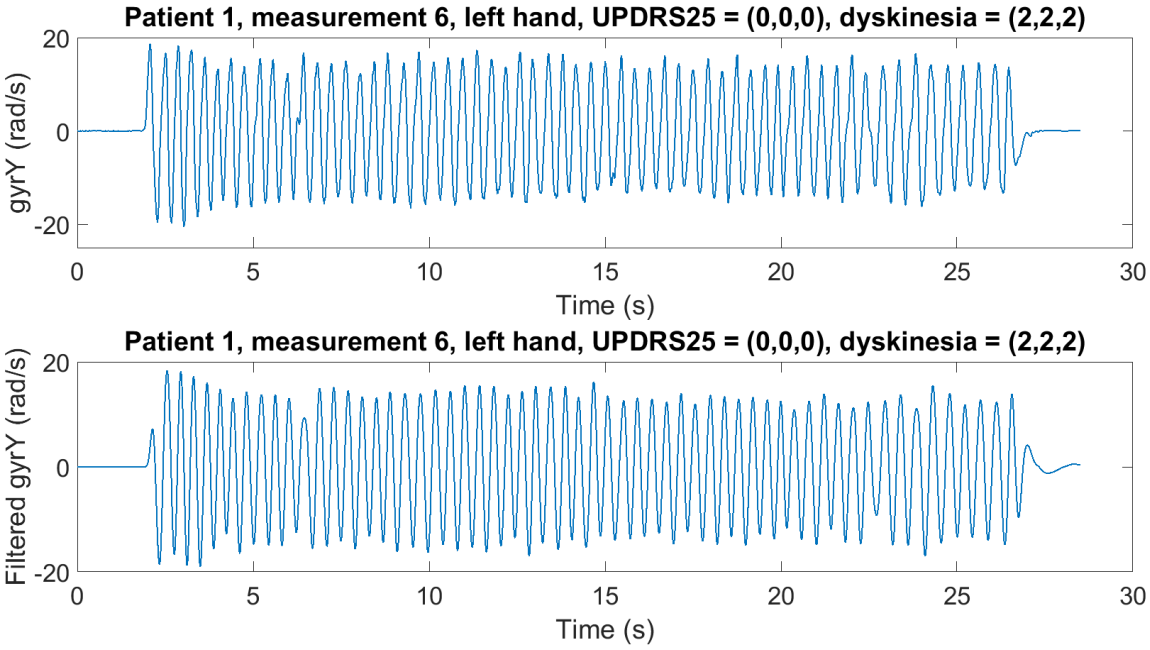

Figure 4.1: Recorded and bandpass filtered gyroscope measurement of the more affected hand of patient 1 of UPDRS item 25 on the 6th run of the UPDRS protocol

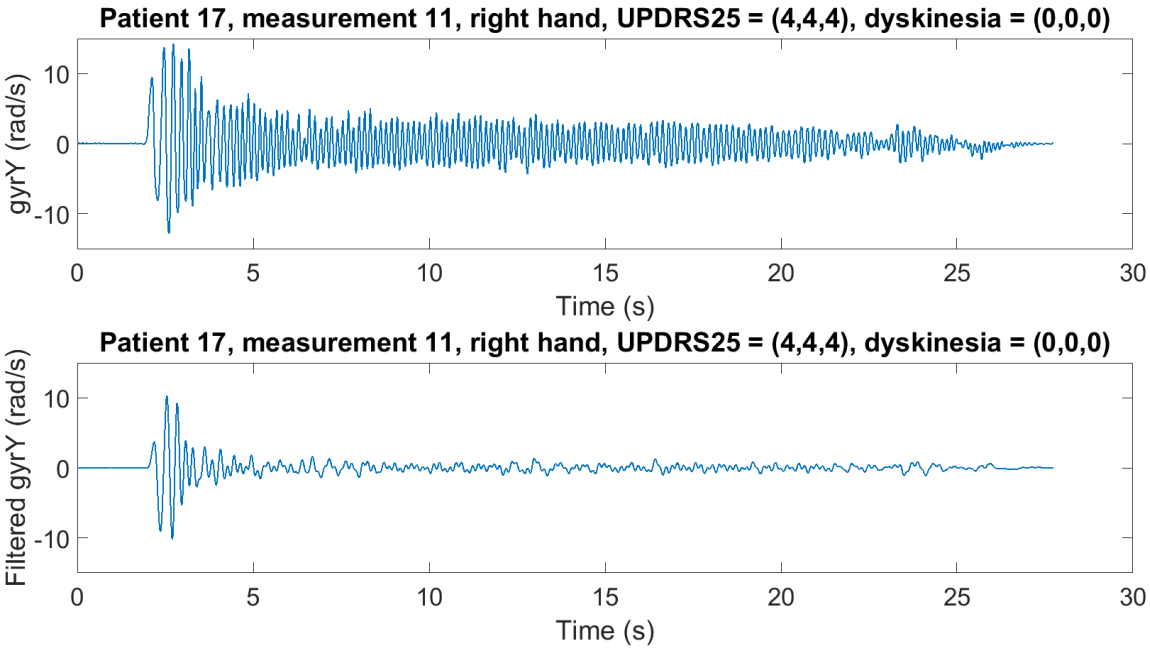

Figure 4.2: Recorded and bandpass filtered gyroscope measurement of the more affected hand of patient 17 of UPDRS item 25 on the 11th run of the UPDRS protocol

Here, N denotes the index of the final measurement sample.

This notation is valid for the measurements of all six sensors, for the three gyroscope measurements as well as for the three accelerometer measurements. Each of the measured signals covers approximately 20 seconds and each feature was calculated once for each measurement.

Additionally to several time domain predictors, some features were based on the frequency domain representations of the measured signals as some characteristics may not become apparent in the time domain. One can determine the frequency domain representation of a signal with help of the Discrete Fourier Transform (DFT) as described by Proakis and Manolakis [23]:

$$X[k] = \sum_{n=0}^{N-1} x[n] \mathrm{e}^{-j\frac{2\pi kn}{N}}. \tag{4.2}$$

As shown in appendix A, this definition of the DFT is not affected by the sampling frequency. The relationship between a frequency index $k = 0, \ldots, K/2 = 0, \ldots, N/2$ and the corresponding continuous time frequency f is given by

$$f = \frac{k}{N} f_s. \tag{4.3}$$

Apart from time and frequency domain characteristics, the variations of the proposed predictors as time progresses are of interest as well since the measured motion was specifically designed to emphasize changes in movement patterns. This design derived from the observation that, according to Wipenmyr and Bergquist [34], a steady decrease of amplitude and frequency of motion measurements presents one of the major symptoms of Parkinson's disease.

According to the Dictionary of Physics [15], the standard deviation σ is a measure of the distribution of a signal around its mean value μ given by

$$\sigma = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (x[n] - \mu)^2} \tag{4.4}$$

$$= \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(x[n] - \frac{1}{N} \sum_{n=1}^{N} x[n] \right)^2}. \tag{4.5}$$

In order to calculate the standard deviation of a predictor x_p instead of the standard deviation of a measurement $x[n]$ at hand, one may divide each measurement of length N into M overlapping intervals of 2s length and with 50% overlap. Then, one can calculate the value of a predictor x_p for each of these intervals, i.e. $x_p[m]$ denotes the value of a predictor derived from samples of the measurement $x[n]$ which lie within the m-th time window. Once the value of a predictor has been calculated for each of the M time windows, the standard deviation of the predictor can be determined as

$$\sigma_p = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(x_p[m] - \frac{1}{M} \sum_{m=1}^{M} x_p[m] \right)^2}. \tag{4.6}$$

However, a predictor with slightly varying large values may exhibit the same standard deviation as a predictor with strongly varying small values. Therefore, the standard

deviation σ_p of a predictor relative to the total value of each predictor x_p as obtained when considering the whole measurement sequence was taken into account:

$$\sigma_x^r = \frac{\sigma_p}{x_p} = \frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(x_p[m] - \frac{1}{M}\sum_{m=1}^{M} x_p[m]\right)^2}}{x_p}. \tag{4.7}$$

A convenient tool for the implementation of the standard deviation of frequency domain features is the MATLAB function `spectrogram` which calculates the Short Time Fourier Transform (STFT) $X_{STFT}[m, f_{STFT}]$, also known as time-dependent Fourier Transform, as well as the frequencies f_{STFT} and an estimate of the energy spectrum $E_{STFT}[f_{STFT}[m]]$ over time. According to Oppenheim and Schafer [19], the STFT is defined by

$$X_{STFT}[s, f_{STFT}] = \sum_{n=-\infty}^{\infty} x\left[s+n\right] w[n] \mathrm{e}^{-j2\pi f_{STFT} n} \tag{4.8}$$

where s describes a discrete time variable. The window function $w[n]$ allows the calculation of the DFT for some windowed time intervals.

One should increase the number of time domain samples in each window to a multiple of 2 by adding samples with the value zero in order to allow for a more efficient computation of the DFT with help of the Fast Fourier Transform (FFT) algorithm (compare [19]). To avoid confusion between the original number of time domain samples per window and the zero-padded measurement sequence per window, the number of samples of the latter is denoted as l_{STFT}, namely the length of the DFT used in the STFT. The frequency indices k_{STFT} of $f_{STFT} = k_{STFT} f_s / l_{STFT}$ are defined as $k_{STFT} = 0, \ldots, l_{STFT}/2$.

For the purpose of calculating the DFT in M windowed intervals, the `spectrogram` function allows the specification of window length (2 s), overlap (50%), length of the DFT (l_{STFT}) and sampling frequency (f_s). In order to divide the original measurement $x[n]$ into 2 s intervals, the `spectrogram` function applies a Hamming window designed to minimize the largest sidelobes.

In the following, each of the predictors considered in this work is described in detail. Any constant factors equivalent for all measurements may be neglected in the implementation of the feature calculation but are included here for completeness. These factors have no impact on the developed models since the models utilize only differences between observations which are unaffected by constants.

4.2.1 Greatest Acceleration and Angular Velocity

Bradykinesia is characterized by slowness of both the initiation and execution of movements which should be reflected by the measured angular velocities and accelerations of bradykinetic and dyskinetic patients.

Considering that the usefulness of the greatest acceleration as a predictor has been shown by Griffith et al. [5], the maximum absolute values x_{max} of both the measured accelerations and angular velocities was taken into account. These are defined as

$$x_{max} = \max_{n=1,\ldots,N} \left(|x[n]|\right). \tag{4.9}$$

4.2.2 Standard Deviation of Greatest Acceleration and Angular Velocity

The standard deviation of the greatest acceleration and angular velocity $x_{max}[m]$ of each time interval m relative to the overall greatest acceleration and angular velocity respectively, may be calculated with help of equation (4.7) as follows:

$$\sigma^r_{x_{max}} = \frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(x_{max}[m] - \frac{1}{M}\sum_{m=1}^{M} x_{max}[m]\right)^2}}{x_{max}}. \tag{4.10}$$

4.2.3 Range of Accelerations and Angular Velocities

One may consider the range of accelerations as measured by the accelerometer, as shown by Cancela et al. [3], as well as the range of angular velocities detected with help of the gyroscope. These ranges r_x are given by

$$r_x = \max_{n=1,\ldots,N}(x[n]) \; - \min_{n=1,\ldots,N}(x[n])\,. \tag{4.11}$$

4.2.4 Standard Deviation of Range of Accelerations and Angular Velocities

With help of equation (4.7), the standard deviations of the range of accelerations and angular velocities relative to the total range of accelerations or angular velocities can be determined as follows:

$$\sigma^r_{r_x} = \frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(r_x[m] - \frac{1}{M}\sum_{m=1}^{M} r_x[m]\right)^2}}{r_x}. \tag{4.12}$$

4.2.5 Signal Energy or Root Mean Square Value

Both, the signal energy and the Root Mean Square (RMS) value are measures of the amount of movement exhibited.

According to Haykin and van Veen [8], the energy E of a continuous time signal is given by

$$E = \int_{-\infty}^{\infty} x^2(t)dt \tag{4.13}$$

while the energy of a discrete time signal is defined as

$$E = \sum_{n=-\infty}^{\infty} x^2[n]. \tag{4.14}$$

However, the formulation in equation (4.14) only approximates the integral in equation (4.13) when the sequence $x[n]$ is sampled at a frequency of $f_s = 1$ Hz resulting in a sampling period $T_s = 1/f_s = 1$ s.

Instead, at a sampling frequency of $f_s = 102.4$ Hz the time difference between two samples $x[n]$ and $x[n+1]$ amounts to $\Delta t = 1/f_s = 98$ ms. Consequently, the energy of the available measurements is given by

$$E_t = \frac{1}{f_s}\sum_{n=1}^{N} x^2[n]. \tag{4.15}$$

In statistics, the Root Mean Square (RMS) value of a signal is given by

$$\mathrm{RMS}_x = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x^2[n]}. \tag{4.16}$$

as described in the Dictionary of Physics [15].

The RMS was also used for accelerometer signals by Patel et al. [21] and Cancela et al. [3] and for gyroscope signals by Salarian et al. [26] while the signal energy's usefulness has been shown by Bonato et al. [1] for accelerometer measurements. Additionally, Tsipouras et al. [31] utilized the energy of gyroscope and accelerometer measurements for the assessment of dyskinesia.

As may be seen in the formulas given above, the energy of a signal and its RMS are closely related and consequently, only the energy was applied in this work. However, since the signal energy increases with the length of the measured signal, the total energy E_t of each measurement was divided by the measurement's duration, namely $t_{tot} = N/f_s$, to achieve comparability between the energies of different measurements:

$$\begin{aligned} E_{t,c} &= \frac{E_t}{t_{tot}} \\ &= \frac{\frac{1}{f_s}\sum_{n=1}^{N} x^2[n]}{\frac{N}{f_s}} \\ &= \frac{\sum_{n=1}^{N} x^2[n]}{N} \end{aligned} \tag{4.17}$$

4.2.6 Standard Deviation of Signal Energy

Considering equation (4.7), the standard deviation of the signal energy relative to the total energy per second can be calculated as

$$\sigma^r_{E_{t,c}} = \frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(E_{t,c}[m] - \frac{1}{M}\sum_{m=1}^{M} E_{t,c}[m]\right)^2}}{E_{t,c}} \tag{4.18}$$

where $E_{t,c}[m]$ describes the signal energy for each of the M time intervals.

4.2.7 Signal Entropy

Signal entropy is a measure of the information content of a signal, i.e. of the uncertainty associated with the signal. According to Tsipouras et al. [31], it shows strong distinctions for dyskinetic movements. However, it was also used for the quantification of bradykinesia, dyskinesia and tremor with help of accelerometer measurements by Patel et al. [21].

Signal entropy was defined by Shannon [28] as

$$H = -K\sum_{b=1}^{B} p_b \log p_b \tag{4.19}$$

where K is a positive constant assumed to be $K = 1$ in this work. This assumption has no relevance for model development in this work since all predictors were standardized as described in section 5.1 before any statistical learning methods was applied.

In the context of equation (4.19), p_b denotes the probability of samples of the signal $x[n]$ to lie within a certain range. For this purpose, the full range of samples was divided into intervals or bins of uniform width and each sample was assigned to the bin within whose range it fell. The total number of bins is denoted as B, while each individual bin is described by b_b where $b = 1, ..., B$. The number of samples assigned to a bin b_b is given by k_b. Then, the probability of samples to lie within each of the defined ranges may be estimated as follows:

$$p_b = \frac{k_b}{\sum_{b=1}^{B} k_b}. \tag{4.20}$$

These probabilities can be estimated with help of MATLAB's `histcounts` function which utilizes a binning algorithm to determine the number of bins b optimal for revealing the shape of the distribution of $x[n]$. The chosen bins b_b have a uniform width and cover the full range of $x[n]$. The number of bins b was not pre-defined since the shape of the distribution of the various measurements $x[n]$ may vary considerably. However, its value remains constant throughout the calculation of signal entropy for each measurement.

4.2.8 Standard Deviation of Signal Entropy

Similarly as for the signal energy, the standard deviation of the signal entropy relative to the total signal entropy can be determined with help of equation (4.7):

$$\sigma_H^r = \frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(H[m] - \frac{1}{M}\sum_{m=1}^{M} H[m]\right)^2}}{H}. \tag{4.21}$$

Here, $H[m]$ describes the signal entropy in each of the M time intervals.

4.2.9 Dominant Frequency

Once the frequency domain representation of a measurement has been calculated with help of the DFT as defined in equation (4.2), one may determine the frequency index corresponding to the dominant frequency f_{dom} as the index k_{dom} for which the frequency amplitude $X[k]$ is maximal, or equivalently as the index k_{dom} with the greatest energy content:

$$k_{dom} = \arg \max_{k=0,...,N/2} \left(X[k]\right). \tag{4.22}$$

Then, with help of equation (4.3), the dominant frequency f_{dom} of the one-sided spectrum follows from the obtained frequency index $k_{dom} \leq N/2$:

$$f_{dom} = \frac{k_{dom}}{N} f_s. \tag{4.23}$$

The dominant frequency component was also used as a predictor by Patel et al. [21] and Bonato et al. [1] for accelerometer measurements. Furthermore, Burkhard et al. [2] used the dominant frequency for the quantification of dyskinesia using gyroscope measurements.

4.2.10 Standard Deviation of Dominant Frequency

Utilizing equation (4.7) defining the relative standard deviation and MATLAB's `spectrogram` function, one may estimate the relative standard deviation $\sigma_{f_{dom}}$ of the dominant frequency f_{dom} as the standard deviation of frequencies $f_{STFT} = k_{STFT} f_s / l_{STFT}$ corresponding to the greatest amplitudes of the STFT $X_{STFT}[m, f_{STFT}]$ defined in equation (4.8), where $\hat{f}_{dom}[m]$ denotes the estimated dominant frequency for each time interval m:

$$\hat{f}_{dom}[m] = \arg \max_{f_{STFT}} \left(X_{STFT}[m, f_{STFT}] \right), \tag{4.24}$$

$$\Rightarrow \quad \hat{\sigma}^r_{f_{dom}} = \frac{\sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}_{dom}[m] - \frac{1}{M} \sum_{m=1}^{M} \hat{f}_{dom}[m] \right)^2}}{f_{dom}}. \tag{4.25}$$

4.2.11 Dominant Frequency Energy

Burkhard et al. [2] employed the amplitude X_{max} of the dominant frequency as a predictor for dyskinesia using gyroscope measurements, while Hoff et al. [9] applied the same predictor to accelerometer measurements in order to quantify dyskinesia. The objective of this study is the quantification of bradykinesia and not dyskinesia. However, the amplitude or energy of the dominant frequency may be useful in this context as well since both provide a measure of the dominance of the most prevalent frequency f_{dom}.

The greatest frequency amplitude X_{max} independent of the signal length is defined as

$$X_{max} = \max_{k=0,\ldots,N/2} \left(\frac{2X[k]}{N} \right). \tag{4.26}$$

A comparable measure which was used in this work is provided by the energy of the dominant frequency.

According to Proakis and Manolakis [23], the energy of a discrete frequency domain signal[2] is defined by

$$E = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2. \tag{4.27}$$

However, a similar argument as for the energy of the signal in the time domain may be employed to find that the energy in the frequency domain in this case corresponds to

$$E_f = \frac{f_s}{N} \sum_{k=0}^{N-1} |X[k]|^2. \tag{4.28}$$

In order to gain comparable frequency domain energies, one must divide the energy given in equation (4.28) by the overall signal length N:

$$E_{f,c} = \frac{f_s}{N^2} \sum_{k=0}^{N-1} |X[k]|^2. \tag{4.29}$$

[2]The definition of the energy is dependent on the definition of the DFT and inverse DFT. Other authors as e.g. Haykin and van Veen [8] define the DFT as $X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \mathrm{e}^{-j \frac{2\pi k n}{N}}$ which corresponds to the energy definition $E = \sum_{k=0}^{N-1} |X[k]|^2$.

To avoid effects caused by the resolution of the frequency domain, the dominant frequency energy was approximated as the average of the frequency energy of the dominant frequency and the nearest surrounding frequencies:

$$E_{dom,c} \approx \frac{2}{3} \cdot \frac{f_s}{N^2} \sum_{k=k_{dom}-1}^{k_{dom}+1} |X[k]|^2. \tag{4.30}$$

4.2.12 Standard Deviation of Dominant Frequency Energy

The results of the `spectrogram` function used for determining the estimated dominant frequencies $\hat{f}_{dom}[m]$ may be utilized for finding an estimate of the energy corresponding to the dominant frequency $E_{dom}[m]$ of each time interval relative to the signal length as

$$\hat{E}_{dom,c}[m] \approx \frac{2}{3} \cdot \frac{f_s}{l_{STFT}^2} \sum_{i=\hat{k}_{dom}[m]-1}^{\hat{k}_{dom}[m]+1} \left| X\left[m, \frac{i f_s}{l_{STFT}}\right] \right|^2 \tag{4.31}$$

where the index $\hat{k}_{dom}[m]$ of the dominant frequency of each window is given by

$$\hat{k}_{dom}[m] = \frac{\hat{f}_{dom}[m]}{f_s} l_{STFT}. \tag{4.32}$$

Therefore, an estimate of the standard deviation $\sigma_{E_{dom}}$ of the dominant frequency energy $E_{dom,c}$ relative to the total dominant frequency energy can be calculated with help of equation (4.7) as follows:

$$\hat{\sigma}^r_{E_{dom}} = \frac{\sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\hat{E}_{dom,c}[m] - \frac{1}{M} \sum_{m=1}^{M} \hat{E}_{dom,c}[m] \right)^2}}{E_{dom,c}}. \tag{4.33}$$

4.2.13 Ratio of Dominant Frequency Energy to Total Energy

Another possible predictor is given by the ratio r_E of the energy of the frequency index k_{dom}, which corresponds to the dominant frequency f_{dom}, to the total energy $E_{f,c}$, given by

$$r_E = \frac{E_{dom,c}[k_{dom}]}{E_{f,c}}. \tag{4.34}$$

This predictor was also used for accelerometer signals by Patel et al. [21].

4.2.14 Standard Deviation of Energy Ratio

The energy spectral density $E_{STFT}[f_{STFT}[m]]$ obtained with help of the `spectrogram` function describes the energy corresponding to each frequency. Therefore, one may estimate the ratio r_E between the estimated dominant frequency energy $\hat{E}_{dom,c}[m]$ and the estimated total energy $\hat{E}_{STFT}[m]$ of each 2s time interval as follows:

$$\hat{r}_E[m] = \frac{\hat{E}_{dom,c}[m]}{\hat{E}_{STFT}[m]}. \tag{4.35}$$

Applying equation (4.7), one can determine an estimate of the relative standard deviation σ_{r_E} of the ratio between the dominant frequency energy and the total energy of each 2s time interval as follows:

$$\hat{\sigma}^r_{r_E} = \frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\hat{r}_E[m] - \frac{1}{M}\sum_{m=1}^{M}\hat{r}_E[m]\right)^2}}{r_E}. \tag{4.36}$$

4.2.15 Energy Content in Three Frequency Bands

The energy content of various frequency bands may provide further insights beneficial to the quantification process, although its value for quantification of Parkinson's symptoms has only been shown for the dyskinesia case using broad sub-bands of the frequency spectrum [31]. However, energy content of narrow sub-bands has been used successfully as a predictor in the context of epileptic seizure detection by Czarnecki and Gustafsson [4].

In this work, the energy content of the frequency bands $0.75 - 1.5$ Hz, $1.5 - 2.25$ Hz and $2.25 - 3$ Hz was considered. These may be determined as follows, where b_i^l and b_i^u describe the lower and upper frequency band boundaries respectively while $i = 1, 2, 3$:

$$EC_{b_i^l - b_i^u} = 2\frac{f_s}{N^2}\sum_{k=\frac{b_i^l N}{f_s}}^{\frac{b_i^u N}{f_s}} |X[k]|^2. \tag{4.37}$$

4.2.16 Standard Deviation of Energy Content in Three Frequency Bands

In order to calculate the standard deviation of the energy content, one may utilize the STFT defined in equation (4.8) as calculated with help of the `spectrogram` function. The energy content $EC_{b_i^l - b_i^u}[m]$ of each frequency band for each time interval m is given by

$$EC_{b_i^l - b_i^u}[m] = 2\frac{f_s}{l_{STFT}^2}\sum_{i=\frac{b_i^l l_{STFT}}{f_s}}^{\frac{b_u^l l_{STFT}}{f_s}} \left| X_{STFT}\left[m, \frac{if_s}{l_{STFT}}[m]\right]\right|^2. \tag{4.38}$$

Then, one may use equation (4.7) to calculate the relative standard deviation of the energy content of each of the previously defined frequency bands as follows:

$$\sigma^r_{EC_{b_i^l - b_i^u}} = \frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(EC_{b_i^l - b_i^u}[m] - \frac{1}{M}\sum_{m=1}^{M}EC_{b_i^l - b_i^u}[m]\right)^2}}{EC_{b_i^1 - b_i^u}}. \tag{4.39}$$

4.3 Predictor Set Hypotheses

In order to achieve a broader picture of feasible qualities of model fits as well as possible configurations of sensors in a future monitoring systems, several predictor set hypotheses were considered.

Each hypothesis proposes the usage of a different subset of the 120 available predictors resulting from the calculation of each of the predictors described in section 4.2 per gyroscope and accelerometer for each of the three axes.

The formulated hypotheses include the following:

- only features of the gyroscope around the y-axis are relevant (20 predictors)
- only gyroscope features are relevant (60 predictors)
- only accelerometer features are relevant (60 predictors)

In the narrowest setting, it was assumed that only gyroscope measurements around the longitudinal axis of the forearm are of interest for symptom quantification. This assumption appears natural since the measured motion is rotational about this axis and it implies the consideration of 20 predictors for each measurement.

One may wonder whether the fundamental assumption of the first predictor set hypothesis is correct or whether the addition of gyroscope measurements around the x- and z-axes might improve model performance. Therefore, the second predictor set hypothesis investigated in this work includes the utilization of predictors for gyroscope measurements around all axes, i.e. 60 predictors per movement execution per patient.

In the literature, most studies considered accelerometer measurements along three axes [1, 3, 5, 9, 21]. Although rotational motion is measured in form of centripetal acceleration by accelerometers as well, the usage of accelerometers for the quantification of symptoms from rotational movement may seem less intuitive than the utilization of gyroscope measurements. However, the results of this third hypothesis are especially interesting with an eye to future applications particularly in the home environment, since the use of accelerometers would be more convenient for the patient as accelerometers typically consume less power than gyroscopes, which require constant excitation. Less power consumption implies either the need of smaller batteries or a longer battery life, both of which would be desirable.

The predictor sets defined for the three hypotheses may be used in the modeling methods directly. Other options include the utilization of the predictors' principal components or of a subset of predictors.

In this work, a number of principal components determined with help of cross-validation were applied to linear regression, later referred to as principal component regression. Furthermore, the first two principal components of all predictors of each predictor set hypothesis were used for the development of decision trees as well as local regression models, smoothing splines and KNN models since these are known to be susceptible to the curse of dimensionality.

The number of principal components used on the latter was not varied since this would have required the selection of two tuning parameters. A large number of settings under consideration increases the probability of obtaining a low MSE by chance. Unfortunately, the present lack of data does not allow for further validation after model development and therefore, the number of principal components applied to methods requiring the choice of a tuning parameter was not investigated further.

Alternatively, one may consider only a subset of the original predictor set for model development. As explained in the beginning of this chapter, predictors should not be chosen subjectively. Instead, one may take into account the first few predictors selected by forward selection, the last predictors which remain when employing backward selection, or the predictors contributing most to the first few principal components, if a distinct tendency towards a subset of predictors is identifiable.

Chapter 5

Model Development

This chapter describes the application of the defined predictors as well as the implementation of the considered statistical learning methods.

5.1 Standardization of Predictors

Some statistical machine learning methods, as e.g. the lasso and KNN, require predictors to be provided in a comparable form, namely, the predictors must be normalized to have zero mean and standard deviations scaled to identical values.

According to James et al. [10], one may achieve equal standard deviations with help of the following equation:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}. \tag{5.1}$$

Here, $\bar{x}_j$ describes the mean value of each predictor $j = 1, \ldots, p$ over all observations x_i. Zero mean can be achieved by extending equation (5.1) through the subtraction of the mean value $\bar{x}_j$:

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}. \tag{5.2}$$

The standardized predictors $\tilde{x}_{ij}$ as defined in equation (5.2) were used for all models.

5.2 Rounding of Predicted Responses

The applied regression and spline models predict responses on a continuous scale. However, it is unclear whether a physician would utilize the provided information on a continuous scale or might instead round the received UPDRS score to an integer before its utilization, since the currently used UPDRS allows only integer scores. Consequently, the MSE as defined in equation (3.1) may result in overly optimistic test error estimates.

In order to obtain an estimate of the test error relevant in the clinical context, one can round the predicted response to an integer value before comparing it to the true response. This leads to the following adaptation of the MSE which was applied for all implemented regression and spline models:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \text{round}\left(\hat{f}\left(x_i\right)\right)\right)^2. \tag{5.3}$$

5.3 Division of Data Set for Cross-Validation

It was found that the measurements of patient 15 have a three to five times higher amplitude than all of the other patient's measurements while the videos featuring patient 15's movement show a comparably slow motion. Therefore, one may conclude that the sensors used for the measurements of patient 15 were calibrated incorrectly and consequently, those measurements will not be considered further.

After removal of patient 15's measurements, the utilized data includes 228 measurements obtained from 18 different patients. Unfortunately, all measurements of one patient are implicitly correlated due to their dependence on the patient's general physiology, physical proportions, their characteristic movement patterns and personal manifestation of Parkinson's symptoms. Therefore, a natural division of all data into cross-validation folds is given by patient affiliation, resulting in 18 folds containing 10 to 15 measurements each as shown in table 2.1.

Due to this predefinition of folds, it is not possible to repeat nested cross-validation for a number of recomposed folds as recommended by Krstajic et al. [14]. The benefits of the suggested procedure are outweighed by the disadvantage of overly optimistic test error estimates which can be expected when training and test data are correlated.

5.4 Implementation

The pre-processing and predictor calculation as well as all model development were executed in MATLAB. The functions, settings and procedures used for model development are summarized in the following. The range and frequency of tuning parameters under consideration were estimated through visual inspection of the resulting cross-validated MSEs, i.e. values exceeding the utilized ranges were found to either be infeasible, or to lead to large constant or monotonically increasing MSEs.

5.4.1 Linear Regression

The linear regression model for each cross-validation training set (compare section 3.2.2) was obtained with help of the MATLAB function `fitlm(X,Y)`. Each calculated model was then used to predict the responses of the respective test set utilizing MATLAB's evaluation function `feval(model,Xnew)`. The predicted responses were compared with the true responses and the MSE was determined according to equation (5.3). Finally, the average MSE over all cross-validation test sets was calculated.

5.4.2 Forward Selection

Forward selection requires the choice of a tuning parameter in the form of the number of predictors considered. Therefore, nested cross-validation as described in section 3.2.3 can be applied, where the maximal number of predictors under consideration was specified as the number of all available predictors. The necessary calculations are summarized in the following:

1. For each outer loop, i.e. for $f = 1, ..., 18$:
   (a) Define f-th fold as outside test set $TEST_{out,f}$ and all other folds as outside training set $TR_{out,f}$
   (b) For $f_{in} = 1, ..., 17$ and the outer loop training set $TR_{out,f}$:

i. Define $TEST_{in,k}$ where $k = ((f + f_{in}) \mod (18) + 1)$ as inside test set and all sets currently not defined as inside or outside test set as inside training set $TR_{in,k}$

ii. Determine the null model $\mathcal{M}_{0,k}$ containing only the intercept $\hat{\beta}_0$ using the inside training set $TR_{in,k}$ and MATLAB's `fitlm` function

iii. Predict responses of inside test set $TEST_{in,k}$ using $\mathcal{M}_{0,k}$ and MATLAB's `feval` function.

iv. Compare predicted and true responses and calculate the resulting inner test error $MSE_{0,k}$

v. For each number of predictors $j = 1, ..., p$:

A. Using the function `fitlm` in MATLAB, find all models $\mathcal{M}_{jo}$ that result when each of the remaining $p-j+1$ predictors is added to the predictors used in the previously determined model $\mathcal{M}_{j-1}$

B. From all models $\mathcal{M}_{jo}$, select the one with the smallest RSS as the best model for j predictors and define it as $\mathcal{M}_{j,k}$

C. Use model $\mathcal{M}_{j,k}$ and MATLAB's `feval` function to predict responses of inside test set $TEST_{in,k}$

D. Compare predicted and true responses and calculate the resulting inner test error $MSE_{j,k}$

(c) Calculate the average MSE of the inner loop for each number of predictors using $MSE_{j,k}$

(d) Choose the best number of predictors p_f^b as the number corresponding to the lowest average MSE

(e) Determine the null model $\mathcal{M}_{0,f}$ containing only the intercept $\hat{\beta}_0$ using the outside training set $TR_{out,f}$ and MATLAB's `fitlm` function

(f) For each number of predictors $j = 1, ..., p_f^b$:

i. Using MATLAB's `fitlm` function, find all models $\mathcal{M}_{jo}$ that result when each of the remaining $p_f^b - j + 1$ predictors is added to the predictors used in the previously determined model $\mathcal{M}_{j-1}$

ii. From all models $\mathcal{M}_{jo}$, select the one with the smallest RSS as the best model for j predictors and define it as $\mathcal{M}_{j,f}$

(g) Use model $\mathcal{M}_{p_f^b,f}$ and MATLAB's `feval` function to predict responses of outside test set $TEST_{out,f}$

(h) Compare predicted and true responses and calculate the resulting outer test MSE denoted by MSE_f

2. Calculate the average MSE over all outside loop folds as the average of MSE_f and define it as the cross-validated test error denoted by MSE_{ave}

3. Calculate an estimate of the optimal number of predictors as the average of the chosen best number of predictors p_f^b over all folds

The resulting average test error is an approximation of the MSE that can be expected when a forward selection model trained on all available data is applied to new data.

5.4.3 Backward Selection

As in forward selection, the number of predictors to consider in the backward selection model as well as the resulting average test MSE were determined with help of nested cross-validation. The procedure is identical to the one described for forward selection with the exception of steps 1.(b)ii.-v. and 1.(f). Steps 1.(b)ii.-v. are replaced as follows:

ii. Determine the full model $\mathcal{M}_{p,k}$ considering all predictors using the inside training set $TR_{in,k}$ and MATLAB's `fitlm` function

iii. Predict responses of inside test set $TEST_{in,k}$ using $\mathcal{M}_{p,k}$ and MATLAB's `feval` function

iv. Compare predicted and true responses and calculate the resulting inner test error $MSE_{p,k}$

v. For each number of predictors under consideration $j = 0, ..., p-1$:

A. Using the function `fitlm` in MATLAB, find all models $\mathcal{M}_{p-j-1_o}$ that result when each of the remaining $p-j-1$ predictors under consideration is removed from the predictor set used in the previously determined model $\mathcal{M}_{p-j,k}$

B. From all models $\mathcal{M}_{p-j-1_o}$, select the one with the smallest RSS as the best model for j predictors and define it as $\mathcal{M}_{p-j-1,k}$

C. Use model $\mathcal{M}_{p-j-1,k}$ to predict responses of inside test set $TEST_{in,k}$

D. Compare predicted and true responses and calculate the resulting inner test error $MSE_{p-j-1,k}$

Similarly, step 1.(f) is replaced by

(f) For each number of predictors $j = 1, ..., p_f^b$:

i. Using the function `fitlm` in MATLAB, find all models $\mathcal{M}_{p_f^b-j-1_o}$ that result when each of the remaining $p_f^b - j - 1$ predictors under consideration is removed from the predictor set used in the previously determined model $\mathcal{M}_{p-j}$

ii. From all models $\mathcal{M}_{p_f^b-j-1_o}$, select the one with the smallest RSS as the best model for j predictors and define it as $\mathcal{M}_{p_f^b-j-1,f}$

5.4.4 Ridge Regression

A convenient tool for the implementation of ridge regression is provided by the MATLAB function `ridge(X,Y,λ,0)` which returns the $p+1$ model coefficients $\beta_0, \ldots, \beta_p$. The return of the intercept β_0 is enabled by the final function entry 0. One may employ nested cross-validation as described in section 3.2.3 for selection of the tuning parameter λ as well as model assessment in terms of the cross-validated MSE as follows:

1. For $f = 1, ..., 18$:

(a) Define f-th fold as outside test set $TEST_{out,f}$ and all other folds as outside training set $TR_{out,f}$

(b) For $f_{in} = 1, ..., 17$ and the outer loop training set $TR_{out,f}$:

i. Define $TEST_{in,k}$ where $k = ((f + f_{in}) \mod (18) + 1)$ as inside test set and all sets currently not defined as inside or outside test set as inside training set $TR_{in,k}$

ii. For each discrete value of the tuning parameter λ under investigation:
   A. Find model $\mathcal{M}_{\lambda,k}$ using the function `ridge` in MATLAB
   B. Using the coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$ of the model $\mathcal{M}_{\lambda,k}$ predict responses of inside test set $TEST_{in,k}$
   C. Compare predicted and true responses and calculate the resulting inner test error $MSE_{\lambda,k}$

(c) Calculate the average MSE of the inner loop for each value of the tuning parameter λ using $MSE_{\lambda,k}$

(d) Choose the best tuning parameter λ_f^b as the tuning parameter corresponding to the lowest average MSE

(e) Using MATLAB's `ridge` function and λ_f^b as the tuning parameter, find the model $\mathcal{M}_{\lambda_f^b}$

(f) Use the coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$ of the model $\mathcal{M}_{\lambda_f^b}$ to predict responses of outside test set $TEST_{out,f}$

(g) Compare predicted and true responses and calculate the resulting outer test MSE_f

2. Calculate the average MSE over all outside loop folds as the average of MSE_f and define it as the test MSE_{ave}

3. Calculate an estimate of the optimal value of the tuning parameter λ as the average of the estimated optimal tuning parameters λ_f^b over all folds

In this work, tuning parameter values between 0 and 5000 were considered.

5.4.5 The Lasso

The implementation of the lasso is equivalent to the one of ridge regression with the difference that the MATLAB function `ridge` is replaced by the MATLAB function `lasso`. Here, the coefficients $\hat{\beta}$ including the intercept may be obtained by `[`$[\hat{\beta}_1, ..., \hat{\beta}_p]$`,`$\hat{\beta}_0$`] = lasso(`$X$`,`$Y$`,'Alpha',1,'Lambda',`$\lambda$`)` where `Alpha=1` specifies that the shrinkage penalty is used as described in section 3.1.5. Values of the tuning parameter λ between 0 and 0.5 were considered.

5.4.6 Local Regression

Unfortunately, local regression is only defined in MATLAB for up to two predictors. Instead, the function `lwppredict(X,Y,parameters,Xnew)` provided in the Locally Weighted Polynomials toolbox by Jekabsons [11] was employed for the prediction of responses of test data using a local regression model.

The needed input parameters are calculated by the function `lwpparams(kernel, degree,useKNN,windowSize,weightIterations,[],standardization,safety)` given in the same toolbox.

The kernel, i.e. the region employed for model development, was chosen to have a tricubic shape (`'TRS'`) and the degree was set to 1 for locally linear functions. The option `useKNN` was set to `false` implying that the kernel size is defined as a metric window and not as a neighborhood. The size of the kernel assigned in `windowSize` as a fraction of the largest distance between observations in the predictor space presents the tuning parameter determined in the inner loop of nested cross-validation (compare section 3.2.3).

The number of iterations for the calculation of the predictor's weights were set to 5 while the standardization option was set to false. The last option, the safety option specifying whether a response should only be predicted in the presence of a sufficient number of observations in proximity, was applied (`true`).

The nested cross-validation procedure for selection of the best kernel or window size relative to the space spanned by all observations as well as assessment of the obtained models were similar to those described in sections 5.4.4 and 5.4.5 for ridge regression and the lasso. For local regression, the relative window size represents the tuning parameter λ and instead of the model coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$, the function `predict` is used for the prediction of responses.

The relative window size was investigated for values between 0.4 and 1. Values below 0.4 proved infeasible due to the presence of a number of observations insufficient for local model development within smaller relative windows. A relative window size of 1 implies consideration of all available observations.

5.4.7 Smoothing Splines

MATLAB provides the function `tpaps(X,Y,`λ`)` for the development of smoothing splines in a two-dimensional predictor space. In this work, the function `tpaps` was applied in the predictor space spanned by the first two principal components as described in section 5.4.12.

Responses can be predicted with help of MATLAB's function `fnval(model,Xnew)` and λ may be selected in a cross-validation scheme. Here, tuning parameter values between 0 and 0.2 were considered.

5.4.8 Multivariate Adaptive Regression Splines

MARS is not part of MATLAB's standard functions. Therefore, its implementation in the ARESLab by Jekabsons [12] was utilized. Using this toolbox, one may build a multivariate adaptive regression spline with help of the function `aresbuild(X,Y,parameters)`.

The necessary parameters are determined by the function `aresparams` which allows, among other options, the choice of piecewise-linear or piecewise-cubic models and the specification of a maximal number of basis functions including the intercept. In this work, a piecewise-linear model was chosen while the greatest number of basis functions was selected with help of nested cross-validation. Numbers of basis functions between 0 and 20 were investigated.

The overall procedure for the selection of a number of basis functions and assessment of the obtained model is similar to the one utilized for ridge regression as described in section 5.4.4. For MARS, one may use the toolbox's function `arepredict(model, Xtest)` to predict the responses of some new observations instead of using model coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$ as for ridge regression.

5.4.9 K-Nearest Neighbors

KNN can be implemented with help of the MATLAB functions `fitcknn(X,Y)` and `predict (model,Xnew)` for model development and response prediction respectively. The optimal number of neighbors may be estimated using nested cross-validation as described in section 3.2.3. Here, it was decided to investigate numbers of neighbors ranging from 1 to 190.

The procedure is similar to the one employed for ridge regression, the lasso, local regression and MARS as described in sections 5.4.4 - 5.4.8 with the difference that the values

of the tuning parameter λ are replaced by the number of neighbors under consideration and that the responses of new data are predicted with help of the function `predict` instead of using model coefficients $\hat{\beta}_0, \ldots, \hat{\beta}_p$.

5.4.10 Decision Trees

The procedure implemented for the development and assessment of decision trees is almost identical to that of ridge regression, the lasso, local regression, MARS and KNN. In this case, the full tree may be determined with help of the MATLAB function `fitctree(X,Y, 'Prune','on')` where the option `'Prune'` defines whether or not an estimate of the optimal sequence of pruned subtrees should be included in the model description.

After the tree has been calculated, the MATLAB function `prune(model,'Alpha',`λ`)` can be utilized to prune it. In this function, the option `'Alpha'` specifies that the tuning parameter λ should be used as described in section 3.1.10. In this work, tuning parameter values between 0 and 1 were considered. After pruning, the function `predict (model,Xnew)` may be used to predict responses for new data.

5.4.11 Support Vector Machines

SVMs may be implemented using MATLAB's function `fitcecoc([X,Y],'Learners', parameters)` where the kernel type can be specified with help of the function `parameters = templateSVM('KernelFunction','polynomial','BoxConstraint',C)`. In this work, a polynomial kernel was chosen since Patel et al. [21] concluded that for the quantification of Parkinson's symptoms, SVMs with polynomial kernels are more suitable than SVMs with radial or exponential kernels.

After model building, responses for new observations can be predicted using the MATLAB function `predict(model,Xnew)`. In this case, the tuning parameter to be optimized is the parameter C as defined in section 3.1.11. Its optimal value was estimated with help of nested cross-validation using the same procedure as described for ridge regression in section 5.4.4. In this work, tuning parameter values between 0 and 0.2 were investigated.

5.4.12 Principal Component Analysis

Principal component analysis describes the calculation of principal components prior to the development of a model, where the determined components can be utilized as predictors in another statistical machine learning method. One may find the principal components of a predictor set with help of the MATLAB function `[`ϕ`, z] = pca(X,'Centered',false)`. This function returns the coefficients necessary for the transformation of predictors, i.e. ϕ, as well as the resulting principal components $z_{im} = \sum_{j=1}^{p} \phi_{jm} x_{ij}$ as described in section 3.1.12.

The option 'Centered' instructs the function to subtract the predictor mean from each predictor prior to the calculation of principal components. Its default setting is `true`, i.e. predictors are centered unless specified otherwise. However, centering is not necessary here since the predictors have all been standardized already.

The model development and assessment procedure for models requiring the selection of a tuning parameter is summarized in the following:

1. For $f = 1, ..., 18$:
   (a) Define f-th fold as outside test set $TEST_{out,f}$ and all other folds as outside training set $TR_{out,f}$

(b) Determine the principal components $z_{im,out}$ and coefficients $\phi_{jm,out}$ of the outside training set

(c) For $f_{in} = 1, ..., 17$ and the outer loop training set $TR_{out,f}$:

i. Define $TEST_{in,k}$ where $k = ((f + f_{in}) \mod (18) + 1)$ as inside test set and all sets currently not defined as inside or outside test set as inside training set $TR_{in,k}$

ii. Determine the principal components $z_{im,in}$ and coefficients $\phi_{jm,in}$ for the inside training set using MATLAB's `pca` function

iii. For each discrete value of the tuning parameter λ under investigation:

A. Find model $\mathcal{M}_{\lambda,k}$ using some modeling method on the principal components $z_{im,in}$

B. Determine the principal components of the test set $TEST_{in,k}$ through multiplication with the coefficients $\phi_{jm,in}$

C. Using the model $\mathcal{M}_{\lambda,k}$ and the principal components of the inside test set $TEST_{in}$, predict responses of $TEST_{in,k}$

D. Compare predicted and true responses and calculate the resulting inner test error $MSE_{\lambda,k}$

(d) Calculate the average MSE of the inner loop for each value of the tuning parameter λ using $MSE_{\lambda,k}$

(e) Choose the best tuning parameter λ_f^b as the tuning parameter corresponding to the lowest average MSE

(f) Using some modeling method, the principal components $z_{im,out}$ of the outer training set and λ_f^b as the tuning parameter, find the model $\mathcal{M}_{\lambda_f^b}$

(g) Determine the principal components of the outer test set $TEST_{out,f}$ through multiplication with the coefficients $\phi_{jm,out}$

(h) Using the model $\mathcal{M}_{\lambda_f^b}$ and the principal components of the outer test set $TEST_{out,f}$, predict responses of outside test set

(i) Compare predicted and true responses and calculate the resulting outer test error denoted byMSE_f

2. Calculate the average MSE over all outside loop folds as the average of MSE_f and define it as the cross-validated test error MSE_{ave}

3. Calculate an estimate of the optimal value of the tuning parameter λ as the average of the estimated optimal tuning parameters λ_f^b over all folds

This procedure may be utilized for any of the modeling methods described in 5.4.4 - 5.4.11. For modeling methods as e.g. linear regression without tuning parameters, steps 1.(c)-(e) can be omitted.

Additionally, one may estimate the optimal number of principal components to employ for methods which do not require the selection of a tuning parameter. In such a cross-validation procedure, the greatest possible number of principal components is equal to the number of available predictors and instead of values of the tuning parameter λ, possible numbers of principal components are investigated.

Chapter 6

Results

The models were implemented as described in chapter 5 and assessed in terms of the test MSE of predicted responses. Cross-validation was employed for evaluation of models obtained using methods without tuning parameters. Methods which require the choice of a tuning parameter were addressed with help of nested cross-validation as described in section 3.2.3, where an inner cross-validation loop was applied in order to approximate the optimal tuning parameter, while the outer cross-validation loop provided the MSEs utilized for model assessment.

Additionally, the MSE that would result if the UPDRS scores assigned individually by the three physicians were assumed to be predicted UPDRS scores was determined for comparison. This MSE provides a measure of the physicians' agreement among each other and of the intrinsic variance of the available data. Hence, one may not expect the statistical learning methods to perform better than the trained human raters.

For clarity of notation, the cross-validated MSE defined as the average of the test MSEs of all folds will be denoted as MSE_{cv}. Unless indicated otherwise, the MSE_{cv} refers to the cross-validated MSE of the outer cross-validation loop used for model assessment.

6.1 Results for First Predictor Set Hypothesis

The first predictor set hypothesis claims that the 20 predictors calculated for the gyroscope measurements around the y-axis are sufficient for the quantification of bradykinesia present in the measured motion. This hypothesis was evaluated for all modeling methods directly applying the 20 predictors, as well as using the first two principal components and only two predictors for local regression, smoothing splines, KNN and decision trees.

6.1.1 Consideration of 20 Predictors

The MSE_{cv} as well as the standard error (SE) of the MSE_{cv} for models obtained directly considering the 20 predictors derived using only the gyroscope measurements around the y-axis are summarized in table 6.1. The standard errors were always rounded upward to avoid underestimation of confidence intervals. One may observe that, due to large standard errors, most of the obtained results do not differ significantly on the one standard error level.

As explained in chapter 2, the physicians evaluated motor symptoms of each patient as recorded over the course of one day. Therefore, the discrete-time sequence of assigned ratings describes the development of symptoms over time. Some examples of the true

Table 6.1: Results of modeling methods utilizing first predictor set hypothesis

Method	Cross-validated MSE ± SE	Tuning parameter ± SE
Linear regression	0.80 ± 0.18	-
Forward selection	0.57 ± 0.12	2.56 ± 0.50
Backward selection	0.75 ± 0.15	2.56 ± 0.44
Ridge regression	0.59 ± 0.13	408 ± 28
Lasso	0.67 ± 0.15	0.12 ± 0.02
Principal component regression	0.55 ± 0.11	1.89 ± 0.34
Local regression	0.78 ± 0.14	0.57 ± 0.01
Linear MARS	0.69 ± 0.15	3.44 ± 0.82
K-nearest neighbors	0.74 ± 0.18	82.3 ± 5.8
Decision tree	0.71 ± 0.16	$(52 \pm 6) \times 10^{-3}$
SVM linear kernel	0.75 ± 0.17	$(41 \pm 4) \times 10^{-3}$
SVM quadratic kernel	1.03 ± 0.18	$(11 \pm 2) \times 10^{-3}$
SVM 3rd degree polynomial kernel	0.90 ± 0.16	$(67 \pm 12) \times 10^{-3}$
Physicians' ratings	0.35 ± 0.30	-

UPDRS assigned by the physicians compared to the UPDRS as predicted by the linear regression model are shown in figure 6.1.

The depicted UPDRS scores correspond to measurements recorded in chronological order. The first test set contains measurements of patient 9 which received the smallest MSE. The second depicted patient (test fold 18) provided the result closest to the MSE_{cv} while the last patient received the worst result (test fold 16). The correspondence between patients and test folds is given in table 2.1.

One may observe that the linear model was able to predict most of patient 9's UPDRS scores correctly. For patient 19, the UPDRS scores predicted by the linear regression model followed the overall shape of the UPDRS score's variation as it changed over time, indicating a strong correlation, but for most time instances, the model did not predict the precise correct score. The last test fold received a poor MSE because the model was not able to predict the worst UPDRS scores correctly. Considering the lack of measurements assigned a UPDRS score of 4 and linear regression's equal consideration of all observations in the predictor space, this is not a surprising result.

A graphical comparison of the MSE_{cv} for linear regression, forward selection and backward selection as well as the test MSEs resulting in each cross-validation loop are depicted in figure 6.2, while a similar comparison of linear regression, ridge regression and the lasso is given in figure 6.3. In both figures, one may observe that neither one of the five compared methods resulted in a smaller test MSE than the other depicted models for all folds. The two test folds containing measurements of the two patients with the highest possible UPDRS score (test fold 14 and 16) received some of the worst MSEs. However, patient 5 was assigned a comparably high MSE as well, despite the utilization of measurements with similar UPDRS scores for model development. Similar illustrations for all other methods using 20 predictors are given in figures B.19 to B.21 in appendix B.

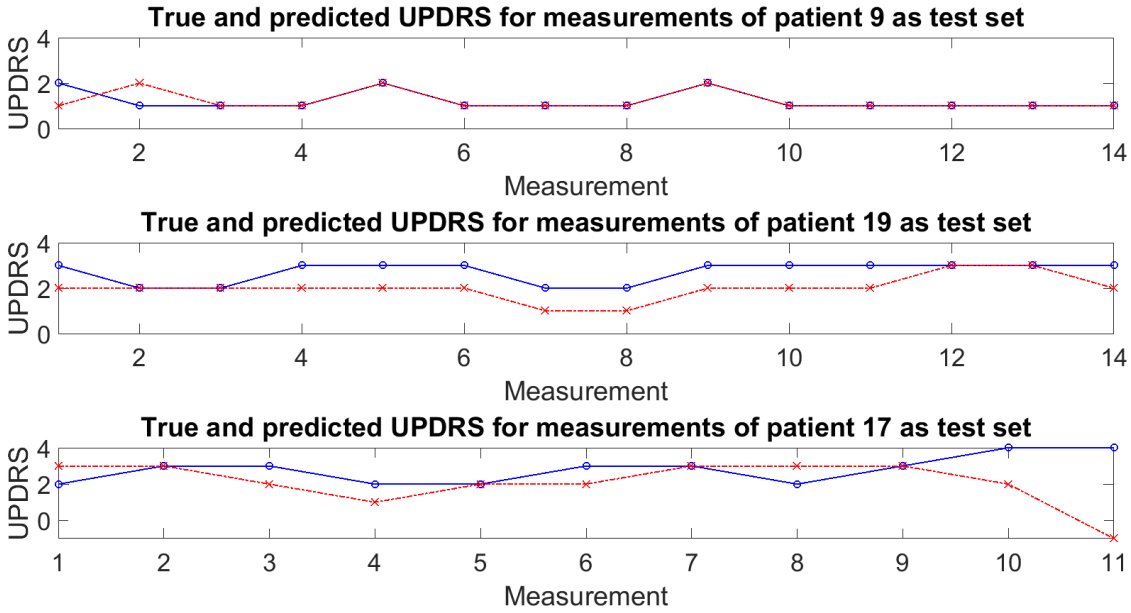

Figure 6.1: True UPDRS scores (blue) and UPDRS scores as predicted by linear regression models (red) for three patients, each used as test fold during model development

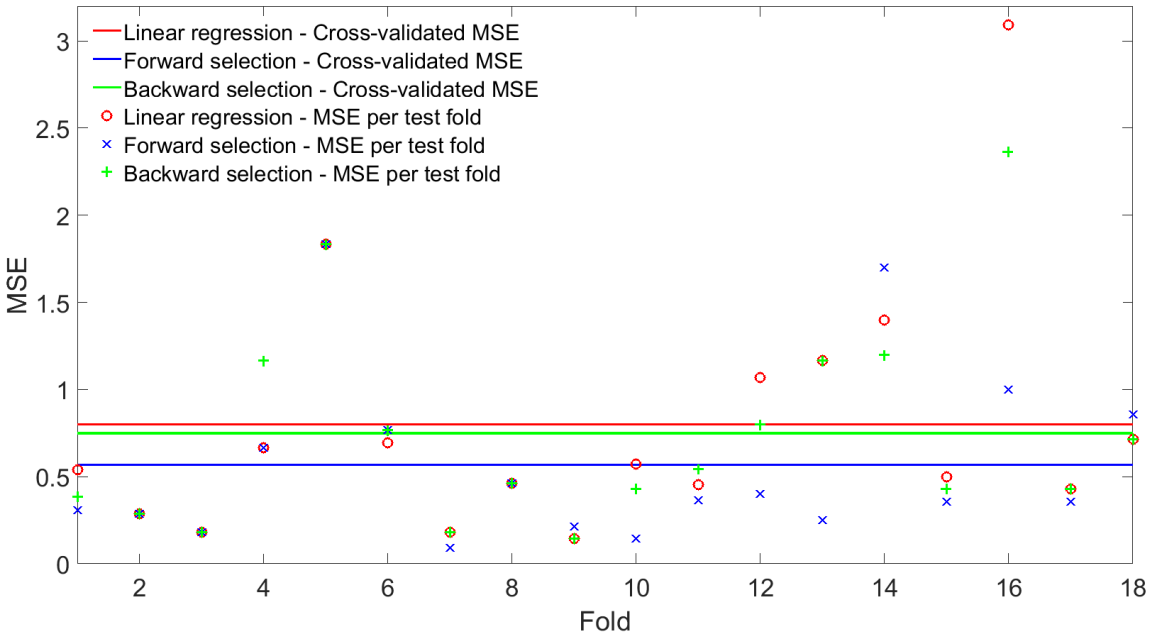

Figure 6.2: MSE corresponding to each outer loop test fold and MSE_{cv} for linear regression (red), forward selection (blue) and backward selection (green) respectively

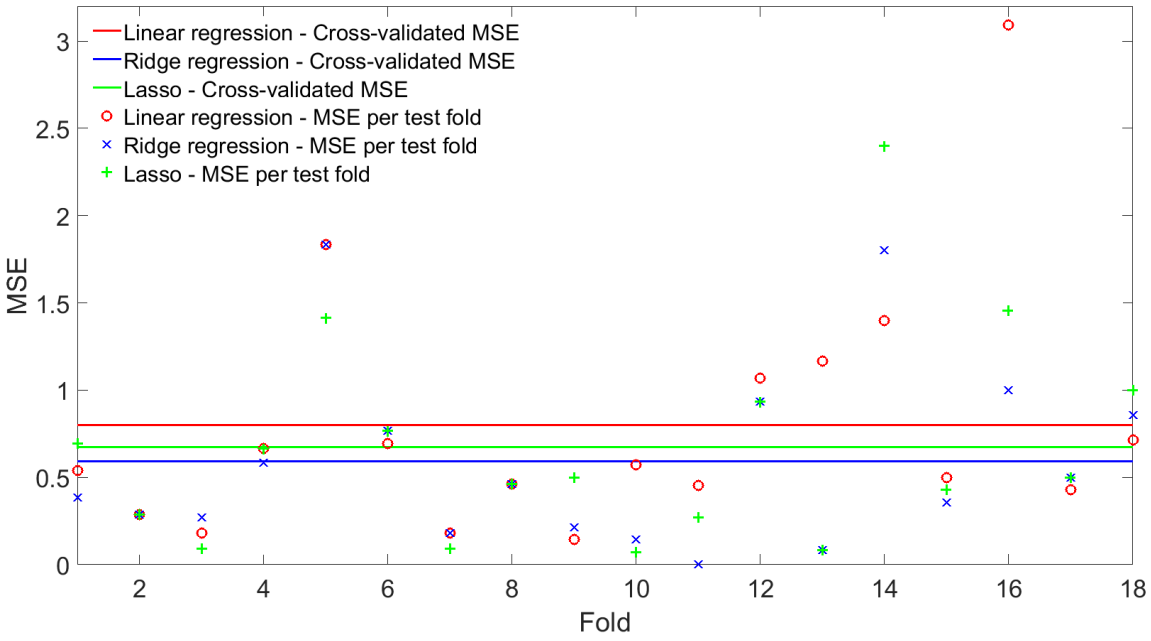

Figure 6.3: MSE corresponding to each outer loop test fold and MSE_{cv} for linear regression (red), ridge regression (blue) and the lasso (green) respectively

Furthermore, an illustration of the average of the MSE_{cv} of the inner cross-validation loop used for parameter selection in forward and backward selection is given in figure 6.4. Here, the average of the inner loop MSE_{cv} is related to the number of predictors considered for model building. The inner loop MSE_{cv} for 20 considered predictors is approximately equivalent to the MSE_{cv} of the linear regression model. Comparison of the average of the MSE_{cv} shows that backward selection results in a larger MSE_{cv} than forward selection for most numbers of predictors, a difference that is caused by the greedy approach of both methods.

Equivalent figures as 6.4 depicting the average of the inner loop MSE_{cv} for all other methods when using the first predictor set hypothesis are provided in figures B.1 to B.18 in section B of the appendix. Note that these figures do not depict the inner loop MSE_{cv} that was used for selection of the best tuning parameter value in each iteration. Instead, they show the average of all the inner loop MSE_{cv} that were used for this purpose. The inner loop MSE_{cv} for some folds as obtained using forward selection is given in figure 6.5. Similar variations as the ones shown here occur across the different test folds for other modeling methods as well.

Out of the predictor selection methods, forward selection provided the best results even though the differences in MSE_{cv} are not significant. Nonetheless, the first three predictors chosen by forward selection for each outside training set, as shown in table 6.2, clearly indicate which of the 20 predictors may be most useful for the development of a linear model of the data.

Here, the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$ of the gyroscope around the y-axis was chosen as the most important predictor sixteen out of eighteen times. For the second predictor, the choice is less conclusive with nine times the greatest angular velocity x_{max} around the y-axis, six times the energy $E_{t,c}$ of the same gyroscope measurement and three times the range r_x of the angular velocity around the y-axis.

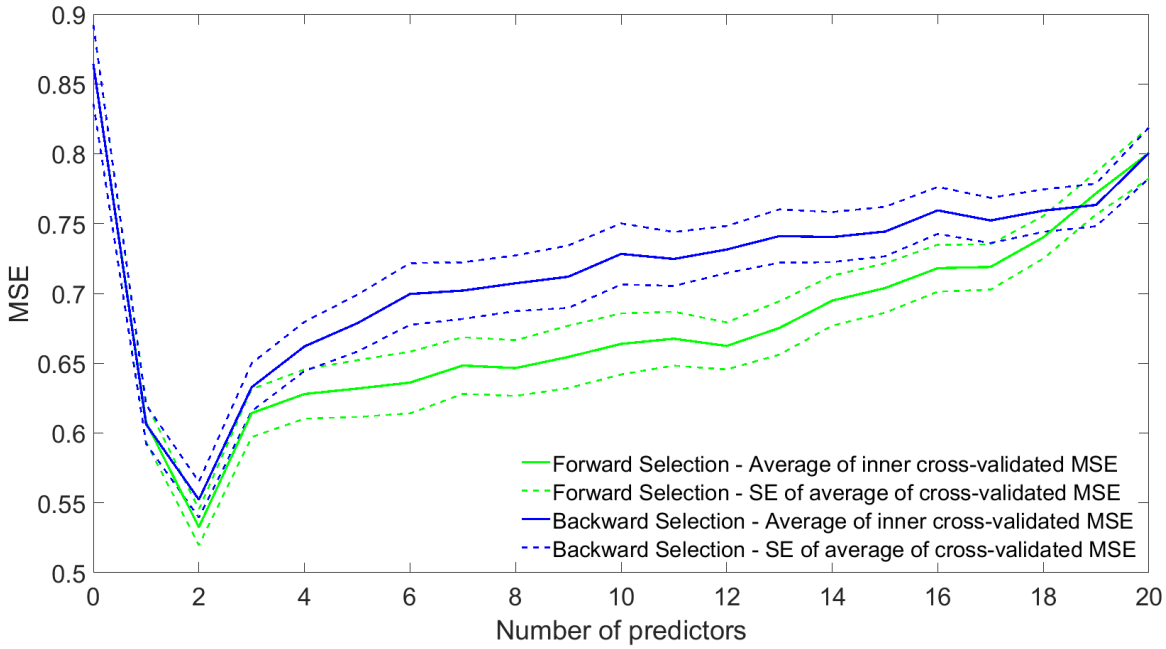

Figure 6.4: Average of the inner loop MSE_{cv} for forward selection models (green) and backward selection models (blue) with respect to the number of predictors used

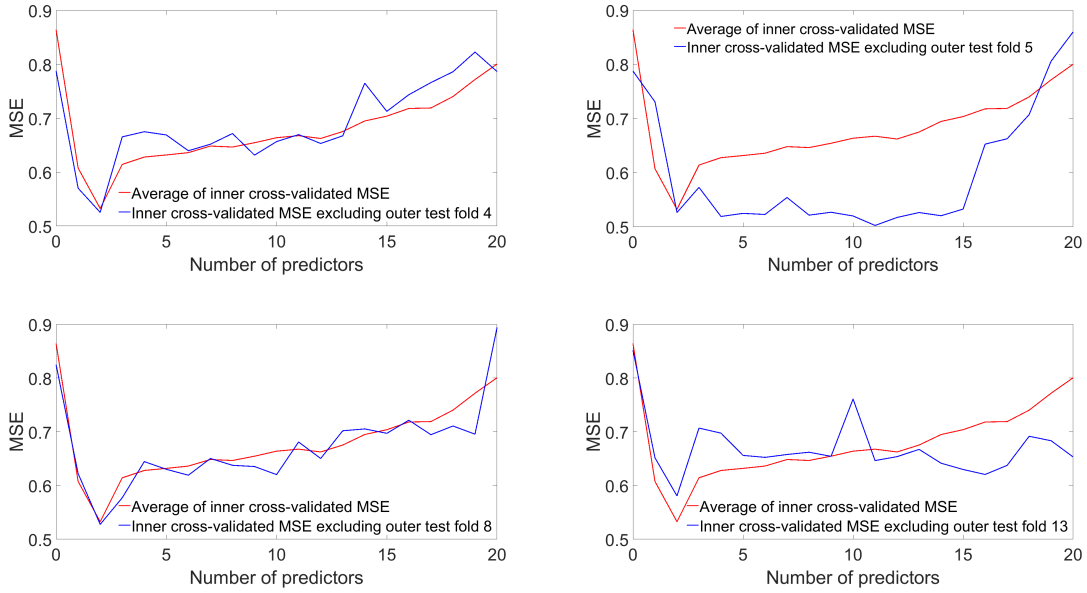

Figure 6.5: Average of all inner loop MSE_{cv} (red) as well as the MSE_{cv} for the individual inner cross-validation procedures (blue) when folds 4 (top left), 5 (top right), 8 (bottom left) and 13 (bottom right) where used as outside test set for forward selection models with respect to the number of predictors used

Table 6.2: First three predictors chosen by forward selection per training set using first predictor set hypothesis. Here, the indicated test fold denotes the fold that was not used during model development.

Test fold	1st predictor	2nd predictor	3rd predictor
1	$\sigma^r_{E_{t,c}}$	x_{max}	$\sigma^r_{EC_{b_2^l - b_2^u}}$
2	$\sigma^r_{E_{t,c}}$	x_{max}	$\sigma^r_{EC_{b_2^l - b_2^u}}$
3	$\sigma^r_{E_{t,c}}$	x_{max}	r_x
4	$\sigma^r_{E_{t,c}}$	x_{max}	r_x
5	$\sigma^r_{E_{t,c}}$	$E_{t,c}$	$\hat{\sigma}^r_{E_{dom}}$
6	$\sigma^r_{E_{t,c}}$	x_{max}	$\sigma^r_{EC_{b_2^l - b_2^u}}$
7	$\sigma^r_{E_{t,c}}$	$E_{t,c}$	$\hat{\sigma}^r_{E_{dom}}$
8	$\sigma^r_{E_{t,c}}$	r_x	x_{max}
9	$\sigma^r_{E_{t,c}}$	$E_{t,c}$	$\hat{\sigma}^r_{E_{dom}}$
10	$\sigma^r_{E_{t,c}}$	$E_{t,c}$	$\hat{\sigma}^r_{r_E}$
11	$\sigma^r_{x_{max}}$	x_{max}	f_{dom}
12	$\sigma^r_{E_{t,c}}$	x_{max}	r_x
13	$\sigma^r_{E_{t,c}}$	x_{max}	r_x
14	$\sigma^r_{E_{t,c}}$	$E_{t,c}$	$\hat{\sigma}^r_{E_{dom}}$
15	$\sigma^r_{E_{t,c}}$	r_x	$\sigma^r_{EC_{b_1^l - b_1^u}}$
16	$\sigma^r_{r_x}$	r_x	$\sigma^r_{EC_{b_1^l - b_1^u}}$
17	$\sigma^r_{E_{t,c}}$	$E_{t,c}$	f_{dom}
18	$\sigma^r_{E_{t,c}}$	x_{max}	$\sigma^r_{EC_{b_1^l - b_1^u}}$

Simultaneously, one may observe that twelve out of twenty defined predictors were chosen as one of the three most important predictors at least once. The only measures that neither appear directly nor in terms of their relative standard deviation are the signal entropy and the energy content of the highest frequency band ($2.25 - 3$ Hz).

The abundance with which the relative standard deviation of the signal energy was chosen as a first predictor suggests that this measure may carry a great deal of information. Therefore, it was decided to evaluate the performance of some methods when utilizing only two predictors as well, namely the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$ and the greatest angular velocity x_{max} around the y-axis. The results of these settings are provided in section 6.1.3.

Returning to the results obtained considering all 20 predictors, one may observe that principal component regression performed better than linear regression on all 20 predictors. The reason for this performance difference could lie in linear regression's consideration of all predictors. When some of the information of the 20 predictors is discarded during analysis of the principal components, this may remove unrelated information, i.e. noise, from the model. Here, the number of principal components best suited for the linear

regression model was determined with help of nested cross-validation.

The more flexible linear multivariate adaptive regression splines did not perform better than the linear models. Even more so than for the linear predictor selection and shrinkage methods, this is caused by the differences among results obtained for different folds.

Local regression and KNN are known to not handle large dimensions well, thus their poor performance is not surprising. Decision trees are known for inaccurate predictions and tend to only outperform parametric models when these are based on incorrect assumptions regarding the mathematical relationship between observations and responses.

6.1.2 Consideration of Two Principal Components

As described in section 3.1.12, principal component analysis aims to reduce the predictor space's dimension by considering each observation's projection onto the axes of largest spread instead of using the original predictor. The results for models utilizing the first two principal components of the first predictor set hypothesis are given in table 6.3. The first of those two principal components is on average influenced most by the relative standard deviation of the greatest angular velocity $\sigma^r_{x_{max}}$, the relative standard deviation of the range of the angular velocity $\sigma^r_{r_x}$, the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$, and the relative standard deviation of the dominant frequency energy $\hat{\sigma}^r_{E_{dom}}$.

However, all other predictors, with exception of the relative standard deviation of the ratio of the dominant frequency energy to the total energy denoted by $\hat{\sigma}^r_{r_E}$, are considered with at least half the weight of the previously mentioned predictors as well, i.e. neither of their transformation factors ϕ_{jm} approaches zero.

Table 6.3: Results of modeling methods applied to the first two principal components of predictors considered in the first predictor set hypothesis

Method	Cross-validated MSE ± SE	Tuning parameter ± SE
PC linear regression	0.51 ± 0.11	-
PC local regression	0.53 ± 0.11	0.62 ± 0.05
PC smoothing splines	0.53 ± 0.11	$(2 \pm 1) \times 10^{-3}$
PC k-nearest neighbors	0.71 ± 0.15	38.6 ± 5.7
PC decision tree	0.61 ± 0.14	$(40 \pm 3) \times 10^{-3}$
Physicians' ratings	0.35 ± 0.30	-

Considering table 6.3, one may observe that the performance of all methods improved with the application of the first two principal components instead of all 20 predictors.

6.1.3 Consideration of Two Predictors

Forward selection chooses predictors based on their contribution to a least square model fit. Therefore, it may give some indication of predictors of interest depending on the true underlying mathematical relationship between observations and responses. As shown in table 6.2, the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$ and the greatest angular velocity x_{max} around the y-axis were chosen by forward selection as the first two predictors, i.e. the predictors causing the greatest decrease of the RSS, most often. The results obtained from application of linear regression, local regression, smoothing splines,

KNN and decision trees to these two predictors are given in table 6.4. These results were better than the results found utilizing any other predictor set.

Table 6.4: Results of modeling methods applied to only two predictors chosen by forward selection out of predictors considered in the first predictor set hypothesis

Method	Cross-validated MSE ± SE	Tuning parameter ± SE
Linear regression	0.50 ± 0.10	-
Local regression	0.49 ± 0.09	0.60 ± 0.03
Smoothing splines	0.54 ± 0.10	$(20 \pm 9) \times 10^{-3}$
K-nearest neighbors	0.51 ± 0.10	7.72 ± 0.36
Decision tree	0.55 ± 0.07	$(9 \pm 2) \times 10^{-3}$
Physicians' ratings	0.35 ± 0.30	-

The MSE_{cv} and the MSE for each fold obtained for linear regression and local regression, the two methods receiving the lowest MSE_{cv}, are depicted in figure 6.6. Both methods lead to approximately equivalent results for more than half of all folds.

For further comparison, the median of the UPDRS scores assigned by the three physicians and the UPDRS scores predicted by the linear regression and the local regression models for patient 8, 10 and 14 are given in figures 6.7 and 6.8. Both models received the smallest test MSE for patient 10 and the highest test MSE for patient 14 while the test MSE of patient 8 was closest to the MSE_{cv}. One may observe that linear regression and local regression returned identical predictions with the exception of the prediction for measurement 8 of patient 14.

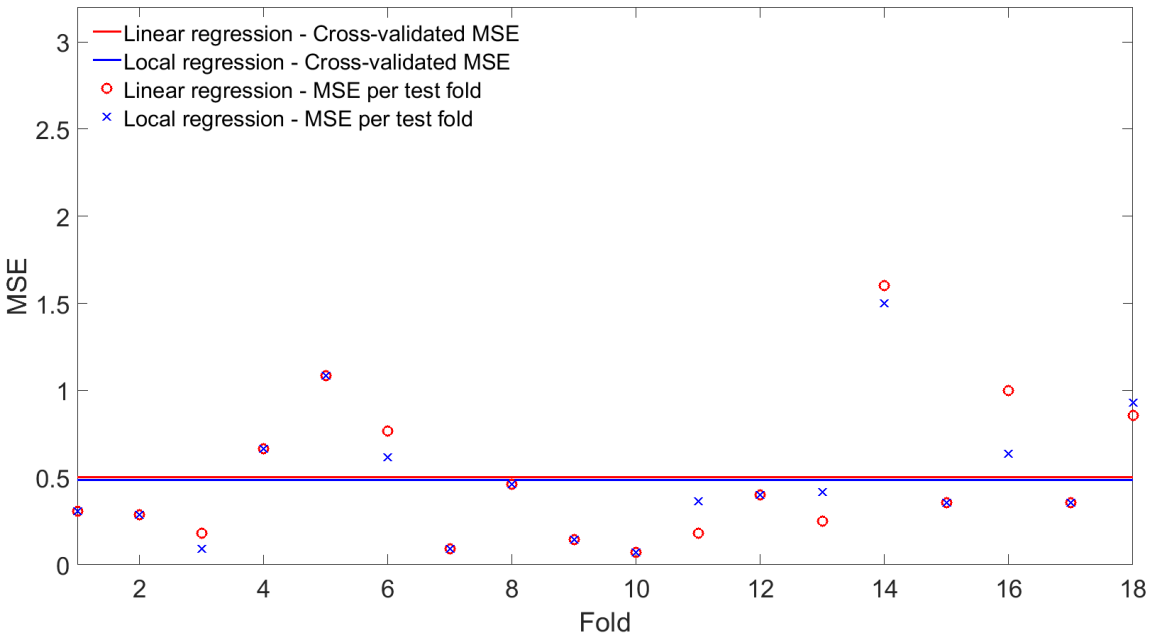

Figure 6.6: MSE for each outer loop test fold and MSE_{cv} for linear regression (red) and local regression (blue)

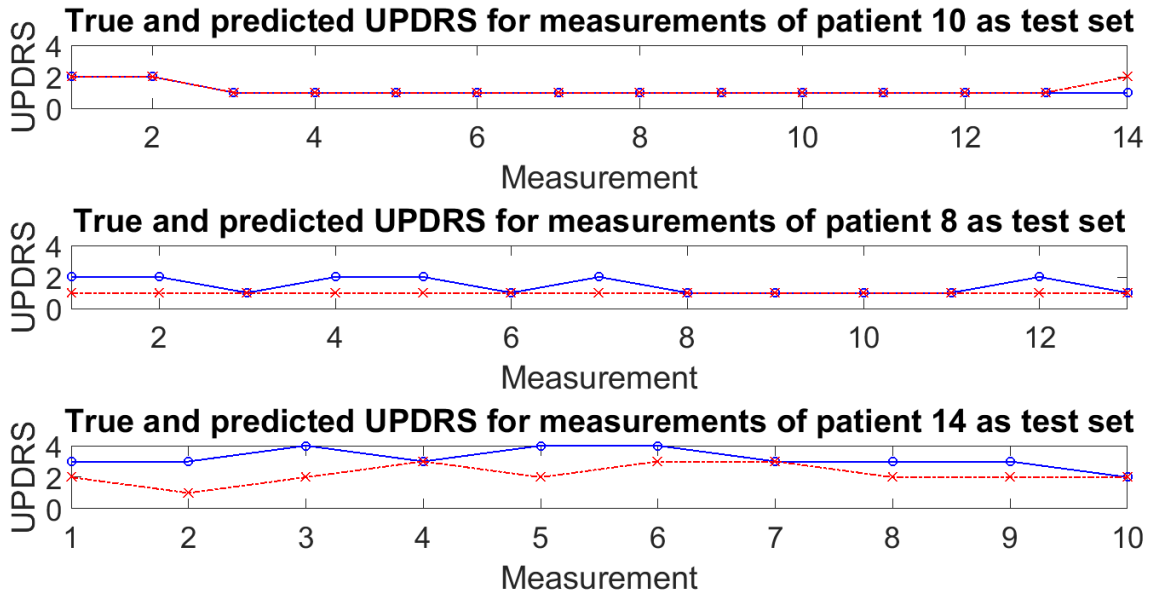

Figure 6.7: True UPDRS scores (blue) and UPDRS scores as predicted by linear regression models obtained using two predictors (red) for three patients, each used as test fold during model development

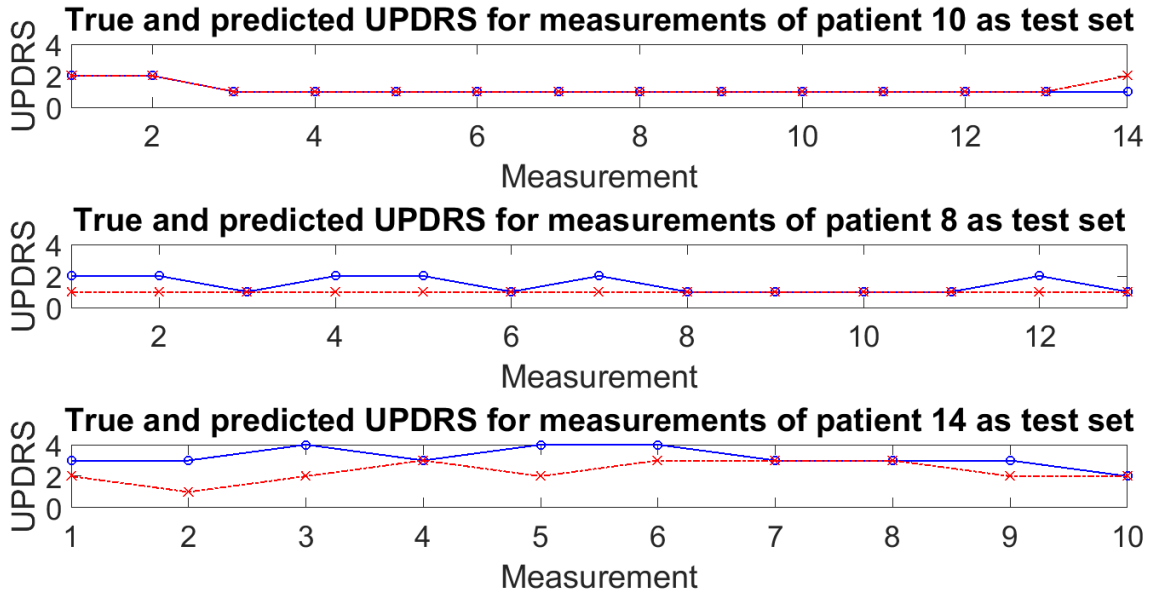

Figure 6.8: True UPDRS scores (blue) and UPDRS scores as predicted by local regression models obtained using two predictors (red) for three patients, each used as test fold during model development

Additionally, the confusion matrices for linear regression and local regression relating predicted to true UPDRS scores are depicted in figures 6.9 and 6.10. Here, the predicted UPDRS scores are the UPDRS scores obtained for all measurements of each outer cross-validation test fold.

The confusion matrices in figures 6.9 and 6.10 show that about 61% of all predicted responses for test observations were correct while the majority of mispredicted UPDRS scores deviated by one step from the median of the UPDRS scores assigned by the physicians. For both linear and local regression only six out of 228 observations, i.e. 2.6% of the available data, were misclassified by more than one step.

Considering the overall accuracy given in the depicted confusion matrices, one may observe that linear regression correctly predicted 139 out of 228 test observations, while local regression assigned the desired score to 138 out of 228 predictions. Nonetheless, the MSE_{cv} of local regression when applied to only two predictors was found to be marginally lower than the MSE_{cv} found using linear regression and the same two predictors. This discrepancy stems from the procedure for calculation of the MSE_{cv}.

For example, two methods may predict responses for two folds where one fold contains ten observations, while the other fold consists of 15 observations. If both methods misclassified six observations by one step but these were distributed differently over the two folds, this may lead to a situation where the MSE_{cv} for the first method is given by

$$MSE_{ave,1} = \frac{\frac{2}{10} + \frac{4}{15}}{2} = 0.23, \tag{6.1}$$

while the MSE_{cv} for the second method amounts to

$$MSE_{ave,2} = \frac{\frac{4}{10} + \frac{2}{15}}{2} = 0.27. \tag{6.2}$$

Although these effects are less pronounced for 18 folds, one may conclude to not attach importance to small differences in the obtained MSE_{cv}.

Neither smoothing splines, KNN or decision trees utilizing the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$ and the greatest angular velocity x_{max} around the y-axis misclassified observations by a large margin as may be seen in figures B.22 to B.24 in appendix B. However, both KNN and decision trees did not predict an extreme UPDRS score (0 or 4) for any of the test observations. For comparison, a confusion matrix for the method which received the largest MSE_{cv} overall (local regression for the second predictor set hypothesis) is depicted in figure B.25.

One advantage of linear regression compared to local regression and KNN is that once a function describing the mathematical relationship between observations and responses has been found, the training data is no longer necessary for the prediction of new data. Instead, the derived function contains sufficient information for the prediction of responses for new observations. Additionally, a linear model is easily interpretable.

For the linear regression model obtained utilizing only the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$ and the greatest angular velocity x_{max} around the y-axis, averaging of the coefficients derived for all folds results in the relationship

$$\hat{y}_i = 1.614 + 0.468\ \sigma^r_{E_{t,c}} - 0.137\ x_{max}, \tag{6.3}$$

where $\hat{y}_i$ denotes a predicted response.

This result indicates that greater variations in the signal energy imply a larger UPDRS score while an increasing greatest angular velocity implies a decrease of the expected UPDRS score, indications that concur with the initial assumptions used for predictor formulation in chapter 4.

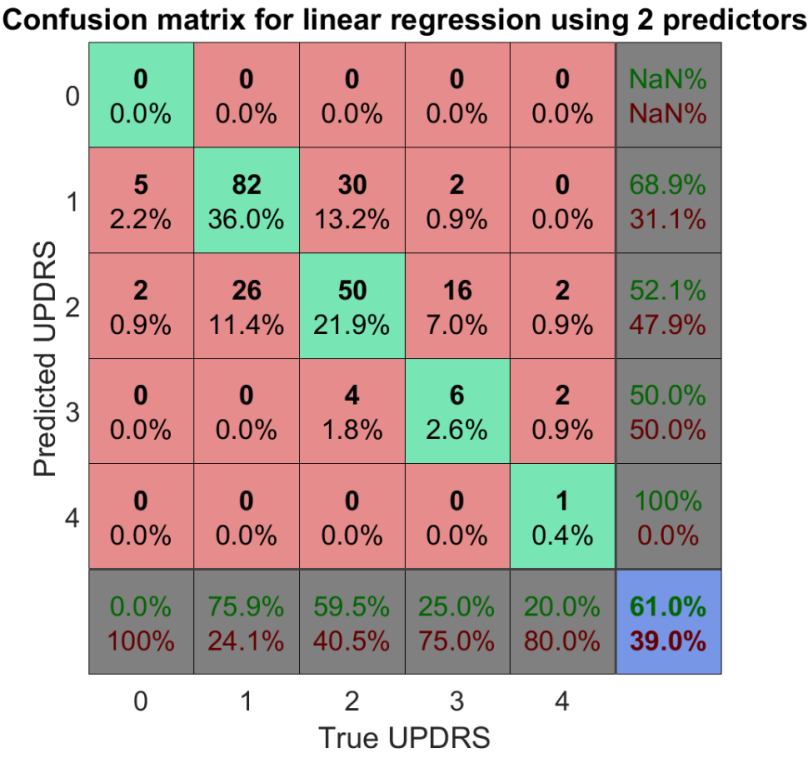

Figure 6.9: Confusion matrix for linear regression with two predictors, where the UPDRS score for each measurement as predicted by linear regression is related to the median of the physicians' UPDRS scores for the same measurement

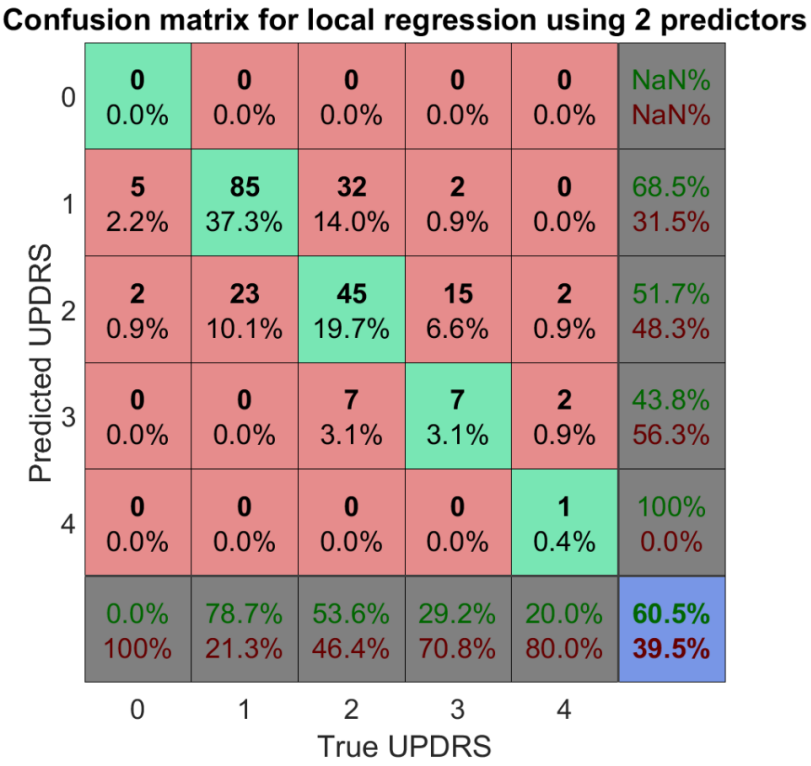

Figure 6.10: Confusion matrix for local regression with two predictors, where the UPDRS score for each measurement as predicted by local regression is related to the median of the physicians' UPDRS scores for the same measurement

6.2 Results for Second Predictor Set Hypothesis

The second predictor set hypothesis includes all 60 predictors per measurement derived from gyroscope measurements around the x-, y- and z-axis.

6.2.1 Consideration of 60 Predictors

Application of the previously described statistical machine learning methods employing all gyroscope measurement predictors directly yields the results shown in table 6.5.

Table 6.5: Results of modeling methods utilizing second predictor set hypothesis

Method	Cross-validated MSE ± SE	Tuning parameter ± SE
Linear regression	1.38 ± 0.56	-
Forward selection	0.61 ± 0.45	2.89 ± 1.31
Backward selection	0.84 ± 0.13	2.56 ± 0.70
Ridge regression	0.61 ± 0.12	920 ± 99
Lasso	0.59 ± 0.13	0.15 ± 0.01
Principal component regression	0.55 ± 0.12	2.73 ± 0.39
Local regression	1.61 ± 0.81	0.63 ± 0.03
Linear MARS	0.82 ± 0.15	4.11 ± 1.00
K-nearest neighbors	0.67 ± 0.11	78 ± 4.2
Decision tree	0.68 ± 0.14	$(59 \pm 4) \times 10^{-3}$
SVM linear kernel	0.91 ± 0.17	$(32 \pm 10) \times 10^{-3}$
SVM quadratic kernel	1.34 ± 0.35	$(17 \pm 3) \times 10^{-3}$
SVM 3rd degree polynomial kernel	0.88 ± 0.14	$0.01 \pm 4.2 \times 10^{-19}$
Physicians' ratings	0.35 ± 0.30	-

Comparison of the results obtained utilizing predictors considered in the first and second predictor set hypotheses shows that only the lasso, KNN, decision trees and SVMs with 3rd degree polynomial kernels performed better when applied to predictors derived from all three gyroscopes. In other words, the addition of predictors did not enhance model performance generally.

In the previous section, the first two predictors chosen by forward selection were found to be beneficial for the results obtained using other modeling methods as well. The first three predictors chosen by forward selection in each outer cross-validation fold considering predictors for measurements of all three gyroscopes are given in table 6.6.

Here, as for the first predictor set hypothesis, the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$ of the gyroscope around the y-axis was chosen as the most important predictor sixteen out of eighteen times. For the second predictor, the greatest angular velocity x_{max} around the y-axis was chosen eight times, the energy $E_{t,c}$ of the same gyroscope measurement four times and the range r_x of the angular velocity around the y-axis three times. Additionally, some features derived from gyroscope measurements around the x- and z-axis appear once each.

These choices indicate that the additional predictors carry relevant information, thus they appear within the first three most relevant predictors from a least squares approach standpoint. Simultaneously, the increase in MSE_{cv} implies that the supplemented predictors contribute more noise than useful information to the models.

Table 6.6: First three predictors chosen by forward selection per training set using second predictor set hypothesis. Here, the indicated test fold denotes the fold that was not used during model development.

Test fold	1st predictor		2nd predictor		3rd predictor	
1	$\sigma^r_{E_{t,c}}$	y-axis	x_{max}	y-axis	$\sigma^r_{EC_{b_2^l - b_2^u}}$	y-axis
2	$\sigma^r_{E_{t,c}}$	y-axis	x_{max}	y-axis	$\sigma^r_{EC_{b_2^l - b_2^u}}$	y-axis
3	$\sigma^r_{E_{t,c}}$	y-axis	x_{max}	y-axis	r_x	y-axis
4	$\sigma^r_{E_{t,c}}$	y-axis	x_{max}	y-axis	r_x	y-axis
5	$\sigma^r_{E_{t,c}}$	y-axis	$E_{t,c}$	y-axis	$\hat{\sigma}^r_{E_{dom}}$	y-axis
6	$\sigma^r_{E_{t,c}}$	y-axis	x_{max}	y-axis	$\sigma^r_{EC_{b_2^l - b_2^u}}$	y-axis
7	$\sigma^r_{E_{t,c}}$	y-axis	$E_{t,c}$	y-axis	$\hat{\sigma}^r_{E_{dom}}$	x-axis
8	$\sigma^r_{E_{t,c}}$	y-axis	r_x	y-axis	x_{max}	y-axis
9	$\sigma^r_{E_{t,c}}$	y-axis	$E_{t,c}$	y-axis	$\hat{\sigma}^r_{E_{dom}}$	y-axis
10	$\sigma^r_{E_{t,c}}$	y-axis	$E_{t,c}$	y-axis	r_x	x-axis
11	$\sigma^r_{x_{max}}$	y-axis	x_{max}	y-axis	$\hat{\sigma}^r_{E_{dom}}$	z-axis
12	$\sigma^r_{E_{t,c}}$	y-axis	$\sigma^r_{EC_{b_2^l - b_2^u}}$	z-axis	$EC_{b_2^l - b_2^u}$	z-axis
13	$\sigma^r_{E_{t,c}}$	y-axis	x_{max}	y-axis	r_x	y-axis
14	$\sigma^r_{E_{t,c}}$	y-axis	f_{dom}	x-axis	$\sigma^r_{EC_{b_1^l - b_1^u}}$	z-axis
15	$\sigma^r_{E_{t,c}}$	y-axis	r_x	y-axis	$\sigma^r_{EC_{b_1^l - b_1^u}}$	y-axis
16	$\sigma^r_{r_x}$	y-axis	r_x	y-axis	$\sigma^r_{EC_{b_1^l - b_1^u}}$	y-axis
17	$\sigma^r_{E_{t,c}}$	y-axis	r_x	x-axis	$\sigma^r_{x_{max}}$	y-axis
18	$\sigma^r_{E_{t,c}}$	y-axis	x_{max}	y-axis	$\sigma^r_{EC_{b_1^l - b_1^u}}$	y-axis

Consideration of the two predictors chosen most often as the first or second predictor that cause the greatest decrease in RSS leads to the same two predictors as chosen for the first predictor set hypothesis. The results of the application of these two predictors to several modeling methods have already been shown in section 6.1.3.

6.2.2 Consideration of Two Principal Components

The results obtained utilizing the first two principal components of the 60 gyroscope predictors are shown in table 6.7. With the exception of local regression and KNN, model performance did not improve compared to that using the first predictor set hypothesis. Additionally, as for the first predictor set hypothesis, the results obtained using KNN and decision trees were worse than those of the regression and spline models.

Table 6.7: Results of modeling methods applied to first two principal components of predictors considered in the second predictor set hypothesis

Method	Cross-validated MSE ± SE	Tuning parameter ± SE
PC linear regression	0.55 ± 0.11	-
PC local regression	0.52 ± 0.12	0.60 ± 0.05
PC smoothing splines	0.57 ± 0.12	$(11 \pm 6) \times 10^{-5}$
PC k-nearest neighbors	0.66 ± 0.14	23.5 ± 5.6
PC decision tree	0.67 ± 0.17	$(49 \pm 5) \times 10^{-3}$
Physicians' ratings	0.35 ± 0.30	-

6.3 Results for Third Predictor Set Hypothesis

The third predictor set hypothesis proposes that the usage of accelerometer measurements may be sufficient for the quantification of bradykinesia.

6.3.1 Consideration of 60 Predictors

The MSE_{cv} as well as the standard error of the MSE_{cv} for models obtained considering the 60 predictors derived using accelerometers along all three axes are summarized in table 6.8.

Table 6.8: Results of modeling methods utilizing third predictor set hypothesis

Method	Cross-validated MSE ± SE	Tuning parameter ± SE
Linear regression	0.84 ± 0.16	-
Forward selection	0.59 ± 0.13	17.4 ± 3.1
Backward selection	0.87 ± 0.17	4.50 ± 0.85
Ridge regression	0.64 ± 0.10	323 ± 45
Lasso	0.65 ± 0.12	$(59 \pm 6) \times 10^{-3}$
Principal component regression	0.71 ± 0.11	8.60 ± 1.50
Local regression	1.00 ± 0.21	0.56 ± 0.01
Linear MARS	0.92 ± 0.21	5.00 ± 0.74
K-nearest neighbors	0.83 ± 0.19	66.9 ± 6.1
Decision tree	0.96 ± 0.16	$(36 \pm 6) \times 10^{-3}$
SVM linear kernel	0.78 ± 0.13	$(44 \pm 6) \times 10^{-3}$
SVM quadratic kernel	0.95 ± 0.18	$(13 \pm 2) \times 10^{-3}$
SVM 3rd degree polynomial kernel	0.84 ± 0.13	$0.01 \pm 4.2 \times 10^{-19}$
Physicians' ratings	0.35 ± 0.30	-

The obtained results did not improve compared to the results derived using the first predictor set hypothesis, with exception of improvements for the lasso and SVMs with quadratic kernels. The predictors chosen by forward selection were not considered further since the results for all 60 predictors indicate that about 17 out of the 60 accelerometer predictors are relevant for model development, a number too large to promise improved performance for methods suffering from the curse of dimensionality. Backward selection performed worse than forward selection, thus its chosen number of 4.5 predictors does not appear promising.

6.3.2 Consideration of Two Principal Components

The results obtained considering the first two principal components of the 60 accelerometer predictors are shown in table 6.9. Here, as for the other two predictor set hypotheses when utilizing principal component analysis, local regression obtained the lowest MSE_{cv}.

Table 6.9: Results of modeling methods applied to first two principal components of predictors considered in the third predictor set hypothesis

Method	Cross-validated MSE ± SE	Tuning parameter ± SE
PC linear regression	0.61 ± 0.11	-
PC local regression	0.60 ± 0.12	0.60 ± 0.07
PC smoothing splines	0.72 ± 0.15	$(12 \pm 10) \times 10^{-4}$
PC k-nearest neighbors	0.81 ± 0.18	80.7 ± 7.9
PC decision tree	0.63 ± 0.12	$(45 \pm 5) \times 10^{-3}$
Physicians' ratings	0.35 ± 0.30	-

Chapter 7

Discussion of Results

As expected, the predictors defined in the first predictor set hypothesis, including only predictors derived from gyroscope measurements around the axis of the forearm, were sufficient for the prediction of UPDRS scores for the considered rotational motion. The addition of predictors derived from measurements of angular velocities around the x- and z-axis did not enhance the models, although these predictors appeared to contain some relevant information as well.

The results obtained utilizing only accelerometer measurements as included in the third predictor set hypothesis were worse than those of the first predictor set hypothesis. However, the results obtained for methods applied to the first two principal components of predictors included in the third predictor set hypothesis may be sufficient for usage in the home environment, if their performance deficiency is outweighed by large increases in battery life.

Across all hypotheses, the results obtained considering only two principal components were superior to the results obtained utilizing all predictors directly. This suggests that not all predictors are relevant. Indeed, model performance increased most when only two predictors, chosen by forward selection as the predictors reducing the RSS of a linear model most, were utilized for model development.

Results for both forward selection and principal component analysis indicate that the addition of predictors reflecting the standard deviation of a predictor's value over time was beneficial for the quantification of bradykinesia for the considered movement. One predictor from the literature that did not seem to have a great impact on the derived models was signal entropy. However, this predictor was previously only utilized for accelerometer measurements for which it may have greater importance.

Additionally, other studies mostly relied on the application of KNN, decision trees and support vector machines. Here, although not performing significantly worse than the other methods under consideration, the mentioned modeling methods did most often not achieve prediction accuracies equivalent to those of linear models.

In theory, all linear predictor selection and shrinkage methods should yield improved results compared to linear regression, the simplest linear modeling method, unless all predictors are equally important for the prediction of the UPDRS score. Forward selection, backward selection, ridge regression and the lasso all lead to improved results. However, the differences are not statistically significant because, as shown in figure 6.5, the MSEs obtained for the various test folds vary too much to offer a clear indication of the optimal number of predictors or the optimal value for the tuning parameter λ to choose. These large variations across the different folds explain why more advanced modeling methods as e.g. MARS and SVM did not perform better than the less sophisticated methods.

In this work, the cross-validation folds were defined in such a way that all measurements of one patient were assigned to the same fold. This appears reasonable to avoid optimistic test MSEs due to correlations between each patient's measurements. However, for the small number of measurements available, these patient-dependent characteristics may be one reason for the large variations observed across different test folds. One might argue that these correlation effects are harmless since a future application could rely on a larger database in which each patient is likely to find a counterpart displaying similar characteristics. However, it is unclear whether sufficient data will be available in the future. In any case, the obtained results represent the worst case scenario, thus application to new data should always yield similar or better results than the ones presented in chapter 6.

Furthermore, the estimation of the optimal tuning parameter λ^* as the average of the optimal tuning parameters found for each fold may not behave as intended for some complex methods. For example, when the folds can be divided clearly into two groups favoring two distinct intervals of the range of the tuning parameter, the average of the chosen tuning parameter values for all folds may be located nowhere near the actual optimal tuning parameter value.

However, neither linear nor local regression were affected by this issue. The optimal window size for local regression considering two predictors was estimated to span 60% of the largest distance between training observations. In other words, predictors in more than half of the occupied predictor space were considered for the development of a linear model around each test observation. This and the good results obtained for linear regression suggest that either the underlying mathematical relationship is effectively a linear one, or that advantages of more complex models are outweighed by their sensitivity to the variability across patient folds.

The confusion matrices given in figures 6.9 and 6.10 may not seem too promising since only about 61% of all test observations were quantified correctly. However, considering the large deviations among human physicians as illustrated in figure 2.2, one may conclude that at least deviations by one score on the UPDRS are not compromising the overall model quality. Additionally, the standard errors of the cross-validated MSEs for most modeling methods were much smaller than those of the physicians' ratings.

Linear and local regression applied to two predictors misclassified only 2.6% of all test observations by more than one step on the UPDRS. However, better prediction rates for extreme UPDRS scores would be desirable. These should be attainable by training the models on a more evenly distributed data set containing more training measurements with UPDRS scores 0 and 4.

For the motion considered in this work, the choice of predictors utilized for model development proved more significant than the selection of a modeling method. The best results were obtained when considering only the relative standard deviation of the signal energy $\sigma^r_{E_{t,c}}$ and the greatest angular velocity x_{max} around the y-axis. Furthermore, close to linear models performed slightly better than models obtained for other structural assumptions. The simplest of those linear models was linear regression and since it received some of the best results, linear regression appears to be the most recommendable modeling method for quantification of bradykinesia for item 25 of the UPDRS protocol.

In the future, this work could be extended to include other motions, movements of other limbs and the quantification of additional symptoms as e.g. dyskinesia, tremor and freezing of gait. Furthermore, a movement recognition algorithm could be added in order to avoid the requirement that patients must repeat pre-defined motions in certain time intervals. Another approach could be the adaptation to individual patients in order to avoid the large deviations observed in the present data.

Chapter 8

Conclusion

Recent studies have shown the feasibility of quantification of several symptoms present in Parkinson's disease from various motions utilizing accelerometer and gyroscope measurements. However, most studies employed non-parametric modeling methods as e.g. KNN and decision trees, hereby avoiding the need to assume an underlying model structure.

In this work, the usefulness of a number of different statistical machine learning methods in conjunction with several predictor sets was evaluated for the example of a predefined motion from the UPDRS protocol, namely repeated forearm rotation which is used for the quantification of bradykinesia. It was found that, for the motion under consideration, the choice of predictors has a greater impact on prediction performance than the model structure. In fact, the results for most methods under consideration did not differ significantly on the one standard error level when the same predictor set was used for model development. However, prediction accuracy generally increased when principal component analysis was employed as a pre-processing step before model development.

The best results were obtained when the relative standard deviation of the signal energy and the greatest angular velocity around the forearm were used as predictors. Utilizing these predictors, linear regression and local regression performed significantly better than smoothing splines or decision trees applied to the same two predictors, while the difference between the cross-validated MSEs for linear regression and KNN was not significant. Consequently, linear regression is not only a valid alternative to the much employed non-parametric models, but it offers greater simplicity and interpretability without any decrease in performance accuracy.

Obviously, the validity of these results is limited to the considered motion and to the quantification of bradykinesia. Additionally, all obtained results are influenced by the definition of patient folds for cross-validation and the assumption of integer responses as well as the imbalance of the utilized data set. Therefore, testing of the obtained models on a new and balanced data set would be desirable. Additionally, further investigation of subsets of the considered predictors derived from accelerometer measurements may lead to better results than those found using all predictors derived from accelerometer measurements. Apart from the motion regarded in this work, similar considerations for different movements including motions of other limbs may prove interesting as well. Furthermore, the application of several statistical machine learning methods for quantification of other symptoms, as e.g. dyskinesia, may lead to a better understanding of the underlying relationship between movement characteristics caused by those symptoms and UPDRS scores.

Once a greater understanding of the underlying characteristics and implications has been gained, this knowledge may be combined with a movement recognition algorithm to allow for unintrusive symptom quantification while patients follow their daily routines.

Bibliography

[1] Paolo Bonato, Delsey M. Sherrill, David G. Standaert, Sara S. Salles, and Metin Akay, Data Mining Techniques to Detect Motor Fluctuations in Parkinson's Disease, *26th Annual International Conference of the IEEE EMBS*, San Francisco, CA, USA, pp. 4766-4769, September 2004.

[2] Pierre R. Burkhard, Heidi Shale, J. William Langston, and James W. Tetrud, Quantification of Dyskinesia in Parkinson's Disease: Validation of a Novel Instrumental Method, *Movement Disorders*, vol. 14, no. 5, pp. 754-763, 1999.

[3] J. Cancela, M. Pansera, M. T. Arredondo, J. J. Estrada, M. Pastorino, L. Pastor-Sanz, and J. L. Villalar, A Comprehensive Motor Symptom Monitoring and Management System: The Bradykinesia Case, *32nd Annual International Conference of the IEEE EMBS*, Buenos Aires, Argentina, pp. 1008-1011, 2010.

[4] Madeleine Czarnecki, and Niclas Gustafsson, *Machine Learning for Detection of Epileptic Seizures*, Master's Thesis EX028/2015, Chalmers University of Technology, pp. 13-16, 2015.

[5] Robert I. Griffiths, Katya Kotschet, Sian Arfon, Zheng Ming Xu, William Johnson, John Drago, Andrew Evans, Peter Kempster, Sanjay Raghav, and Malcolm K. Horne, Automated Assessment of Bradykinesia and Dyskinesia in Parkinson's Disease, *Journal of Parkinson's Disease*, vol. 2, pp. 47-55, 2012.

[6] Ground Up Strength, Supination, February 2016, retrieved from `http://www.gustrength.com/glossary:supination`, June 2nd, 2016.

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning*, Data Mining, Inference and Prediction, 2nd edition Springer Sciene+Business Media, New York, USA, ISBN 978-0-387-84858-7, 2009.

[8] Simon Haykin, and Barry Van Veen, *Signals and Systems*, 2nd edition, John Wiley & Sons, Inc., ISBN 0-471-16474-7, p. 23-24, 202, 304, 2003.

[9] J. I. Hoff, A. A. v/d Plas, E. A. H. Wagemans, and J. J. van Hilten, Accelerometric Assessment of Levodopa-Induced Dyskinesias in Parkinson's Disease, *Movement Disorders*, vol. 16, no. 1, pp. 58-61, 2001.

[10] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An Introduction to Statistical Learning*, Springer Sciene+Business Media, New York, USA, ISBN 978-1-4614-7138-7, 2013.

[11] Gints Jekabsons, Locally Weighted Polynomials Toolbox for MATLAB/Octave, retrieved from `http://www.cs.rtu.lv/jekabsons/regression.html`, May 12th, 2016.

[12] Gints Jekabsons, ARESLab: Adaptive Regression Splines Toolbox for MATLAB/Octave, retrieved from `http://www.cs.rtu.lv/jekabsons/regression.html`, May 10th, 2016.

[13] Katie Kompoliti, Cynthia L. Comella, and Christopher G. Goetz, Clinical Rating Scales in Movement Disorders, Joseph Jankovic, and Eduardo Tolosa (editors), *Parkinson's Disease and Movement Disorders*, 5th edition, Lippincott Williams & Wilkins, ISBN 0-7817-7881-6, p. 693, 2007.

[14] Damjan Krstajic, Ljubomir J. Buturovic, David E. Leahy, and Simon Thomas, Cross-Validation Pitfalls when Selecting and Assessing Regression and Classification Models, *Journal of Cheminformatics*, vol. 6, no. 10, 2014.

[15] Jonathan Law, and Richard Rennie (editors), *A Dictionary of Physics*, 7th edition, Oxford University Press ISBN-13 9780198714743, 2015.

[16] Walter Maetzler, Josefa Domingos, Karin Srulijes, Joaquim Ferreira, and Bastiaan R. Bloem, Quantitative Wearable Sensors for Objective Assessment of Parkinson's Disease, *Movement Disorders*, vol. 28, no. 12, pp. 1628-1637, 2013.

[17] C. D. Marsden, and M. Schachter, Assessment of Extrapyramidal Disorders, M. Lader, and A. Richens (editors), *Central Nervous System*, ISBN 978-1-349-06040-5, p. 92-99, 1981.

[18] Kevin St. P. McNaught, Peter Jenner, and C. Warren Olanow, Protein Mishandling: Role of the Ubiquitin Proteasome System in the Pathogenesis of Parkinson's Disease, Joseph Jankovic, and Eduardo Tolosa (editors), *Parkinson's Disease and Movement Disorders*, 5th edition, Lippincott Williams & Wilkins, ISBN 0-7817-7881-6, p. 33, 2007.

[19] Alan V. Oppenheim, and Ronald W. Schafer, *Discrete-Time Signal Processing*, 3rd edition, Pearson Higher Education, Inc., Upper Saddle River, NJ, USA, ISBN 0-13-206709-9, pp. 38, 649, 668-669, 749-757, 820, 836, 2010.

[20] Parkinson Förbundet, Vanliga frågor och svar, Juni 2015, retrieved from `http://parkinsonguiden.se/fragor-och-svar/fragor-om-lakemedel/`, June 2nd, 2016.

[21] Shyamal Patel, Konrad Lorincz, Richard Hughes, Nancy Huggins, John Growdon, David Standaert, Metin Akay, Jennifer Dy, Matt Welsh, and Paolo Bonato, Monitoring Motor Fluctuations in Patients With Parkinson's Disease Using Wearable Sensors, *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 864-873, November 2009.

[22] Werner Poewe, Nonmotor Symptoms in Parkinson's Disease, Joseph Jankovic, and Eduardo Tolosa (editors), *Parkinson's Disease and Movement Disorders*, 5th edition, Lippincott Williams & Wilkins, ISBN 0-7817-7881-6, p. 68-74, 2007.

[23] John G. Proakis, and Dimitris G. Manolakis, *Digital Signal Processing*, 4th edition, Pearson Higher Education, Inc., Upper Saddle River, NJ, USA, ISBN 0-13-187374-1, pp. 464, 480, 2007.

[24] Serge Przedborski, Etiology and Pathogenesis of Parkinson's Disease, Joseph Jankovic, and Eduardo Tolosa (editors), *Parkinson's Disease and Movement Disorders*, 5th edition, Lippincott Williams & Wilkins, ISBN 0-7817-7881-6, p. 77, 2007.

[25] Daniel P. Redmond, and Frederick W. Hegge, Observations on the Design and Specification of a Wrist-Worn Human Activity Monitoring System, *Behavior Research Methods, Instruments, & Computers*, vol. 17, no. 6, pp. 659-669, 1985.

[26] Arash Salarian, Heike Russmann, Christian Wider, Pierre R. Burkhard, Françios J. G. Vingerhoets, and Kamiar Aminian, Quantification of Tremor and Bradykinesia in Parkinson's Disease Using a Novel Ambulatory Monitoring System, *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 2, pp. 313-322, February 2007.

[27] Annete Schrag, Epidemiology of Movement Disorders, Joseph Jankovic, and Eduardo Tolosa (editors), *Parkinson's Disease and Movement Disorders*, 5th edition, Lippincott Williams & Wilkins, ISBN 0-7817-7881-6, p. 51-52, 2007.

[28] C. E. Shannon, A Mathematical Theory of Communication, *The Bell System Technical Journal*, vol. 27, pp. 623-656, October 1948.

[29] Shimmer3, August 2013, retrieved from `http://www.shimmersensing.com/shop/shimmer3` and `http://www.shimmersensing.com/news/shimmer3-the-newest-wearable-sensor-platform`, May 2nd, 2016.

[30] Eduardo Tolosa, and Regina Katzenschlager, Pharmacological Management of Parkinson's Disease, Joseph Jankovic, and Eduardo Tolosa (editors), *Parkinson's Disease and Movement Disorders*, 5th edition, Lippincott Williams & Wilkins, ISBN 0-7817-7881-6, p. 110-137, 2007.

[31] Markos G. Tsipouras, Alexandros T. Tzallas, George Rigas, Sofia Tsouli, Dimitrios I. Fotiadis, and Spiros Konitsiotis, An Automated Methodology for Levodopa-Induced Dyskinesia: Assessment Based on Gyroscope and Accelerometer Signals, *Artificial Intelligence in Medicine*, vol. 55, pp. 127-135, 2012.

[32] Josep Valls-Solé, Neurophysiology of Motor Control and Movement Disorders, Joseph Jankovic, and Eduardo Tolosa (editors), *Parkinson's Disease and Movement Disorders*, 5th edition, Lippincott Williams & Wilkins, ISBN 0-7817-7881-6, p. 14, 2007.

[33] ´Multimodal motor symptoms quantification platform for individualized Parkinson´s disease treatments´ MuSyQ, January 2016, retrieved from `http://www.vinnova.se/sv/Resultat/Projekt/Effekta/2009-02187/Multimodal-motor-symptoms-quantification-platform-for-individualized-Parkinsons-disease-treatments--MuSyQ/`, May 2nd, 2016.

[34] Jan Wipenmyr, and Filip Bergquist, Rörelseanalys över tid för bättre anpassad medicinering av Parkinsonsjuka, *Proceedings of Vitalis 2014*, Göteborg, April 2014.

Appendices

Appendix A

Sampling and the Fourier Transform

A similar representation of the original continuous signal $x(t)$ as the sampled sequence $x[n]$ as defined in equation (4.1) is given by the impulse train $x_s(t)$ described by Oppenheim and Schafer [19] as

$$\begin{aligned} x_s(t) &= \sum_{n=-\infty}^{\infty} x\left(t\right)\delta\left(t-nT_s\right) \\ &= \sum_{n=-\infty}^{\infty} x\left(nT_s\right)\delta\left(t-nT_s\right). \end{aligned} \tag{A.1}$$

where the continuous signal $x_s(t)$ is zero at all times other than $t = nT_s$.

The continuous time Fourier Transform (FT) as described by Haykin and van Veen [8] where $\omega = 2\pi f$ is defined by

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-j\omega t}dt. \tag{A.2}$$

Therefore, one may determine the FT of the continuous signal $x_s(t)$ as follows:

$$X(j\omega) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x\left(nT_s\right)\delta\left(t-nT_s\right)\mathrm{e}^{-j\omega t}dt \tag{A.3}$$

$$= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x\left(nT_s\right)\delta\left(t-nT_s\right)\mathrm{e}^{-j\omega nT_s}dt \tag{A.4}$$

$$= \sum_{n=-\infty}^{\infty} x\left(nT_s\right)\mathrm{e}^{-j\omega nT_s}\int_{-\infty}^{\infty}\delta\left(t-nT_s\right)dt \tag{A.5}$$

$$= \sum_{n=-\infty}^{\infty} x\left(nT_s\right)\mathrm{e}^{-j\omega nT_s} \tag{A.6}$$

$$= \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-j\omega nT_s}. \tag{A.7}$$

The obtained spectrum $X(j\omega)$ is periodic with a period of $2\pi/T_s$. When sampling the frequency spectrum, the change of frequency between two samples $X[k]$ and $X[k+1]$ is

$\Delta\omega = 2\pi/(KT_s)$ where K denotes the number of frequency samples, i.e. $k = 0, ..., K/2$. Usually, one assumes $N = K$ for notational convenience.

$$X(k\Delta\omega) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-j\frac{2\pi knT_s}{KT_s}}, \tag{A.8}$$

$$\Rightarrow \quad X[k] = \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-j\frac{2\pi kn}{K}} \tag{A.9}$$

$$= \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-j\frac{2\pi kn}{N}}. \tag{A.10}$$

Consequently, through comparison of equations (4.2) and (A.10) one may observe that sampling does not effect the definition of the DFT.

Appendix B

Complementary Figures

Here, some figures complementing the results obtained using the first predictor set hypothesis are provided.

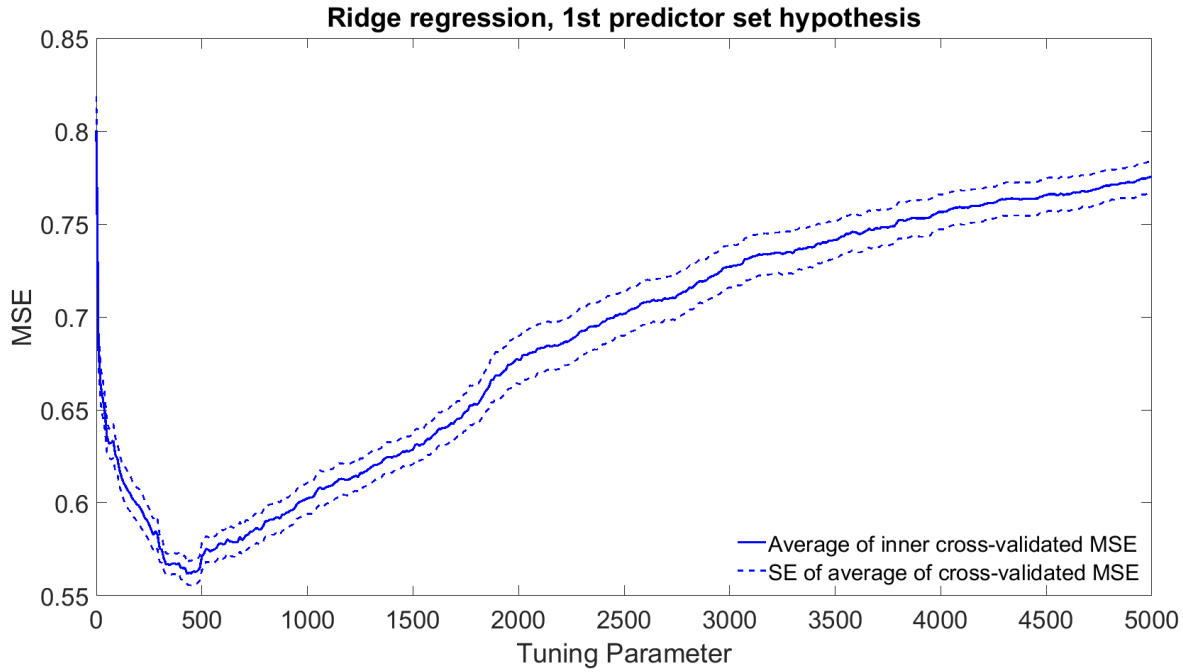

Figure B.1: Average of inner loop MSE_{cv} for ridge regression models with respect to the value of the tuning parameter λ used

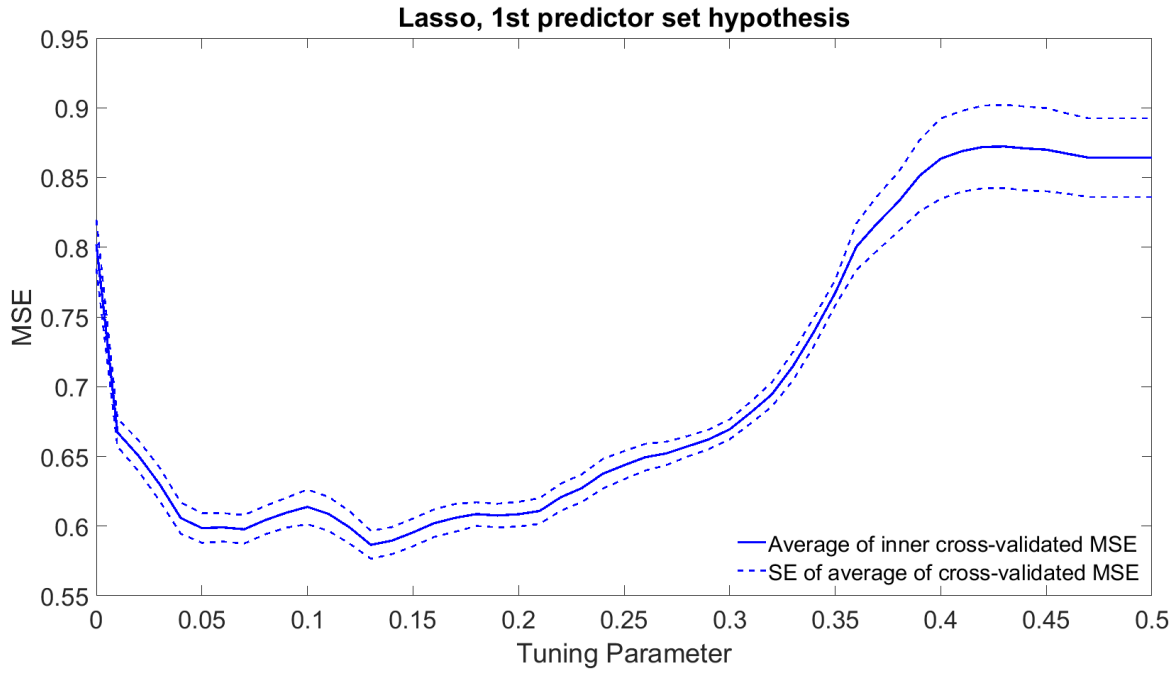

Figure B.2: Average of inner loop MSE_{cv} for lasso models with respect to the value of the tuning parameter λ used

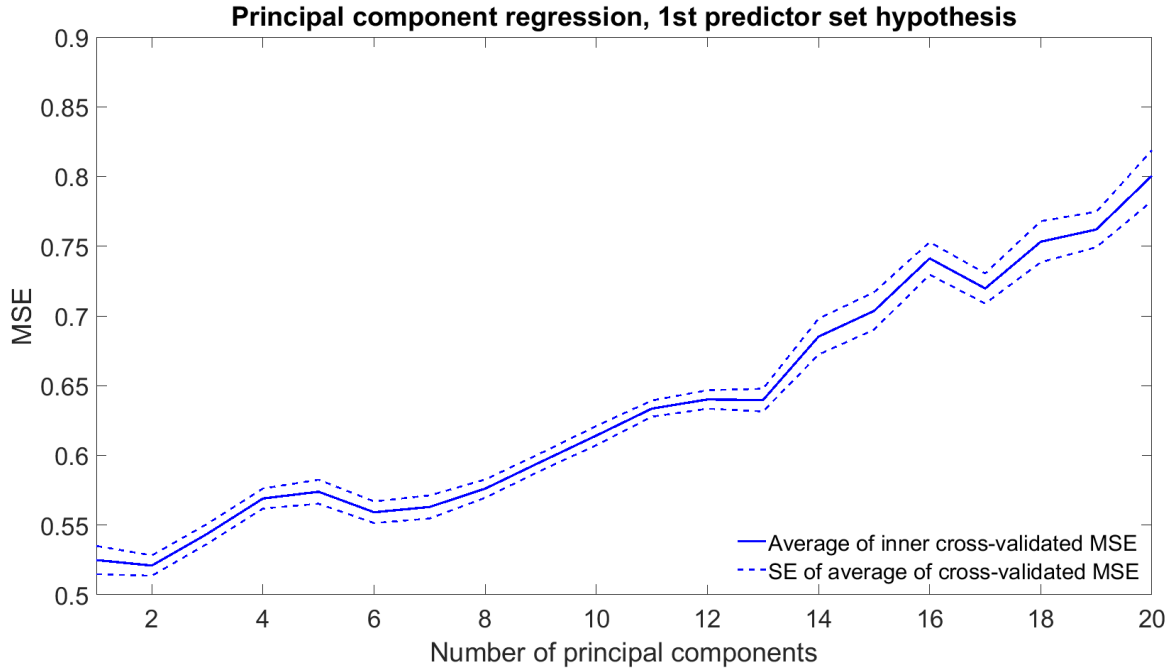

Figure B.3: Average of inner loop MSE_{cv} for principal component regression models with respect to the number of principal components used

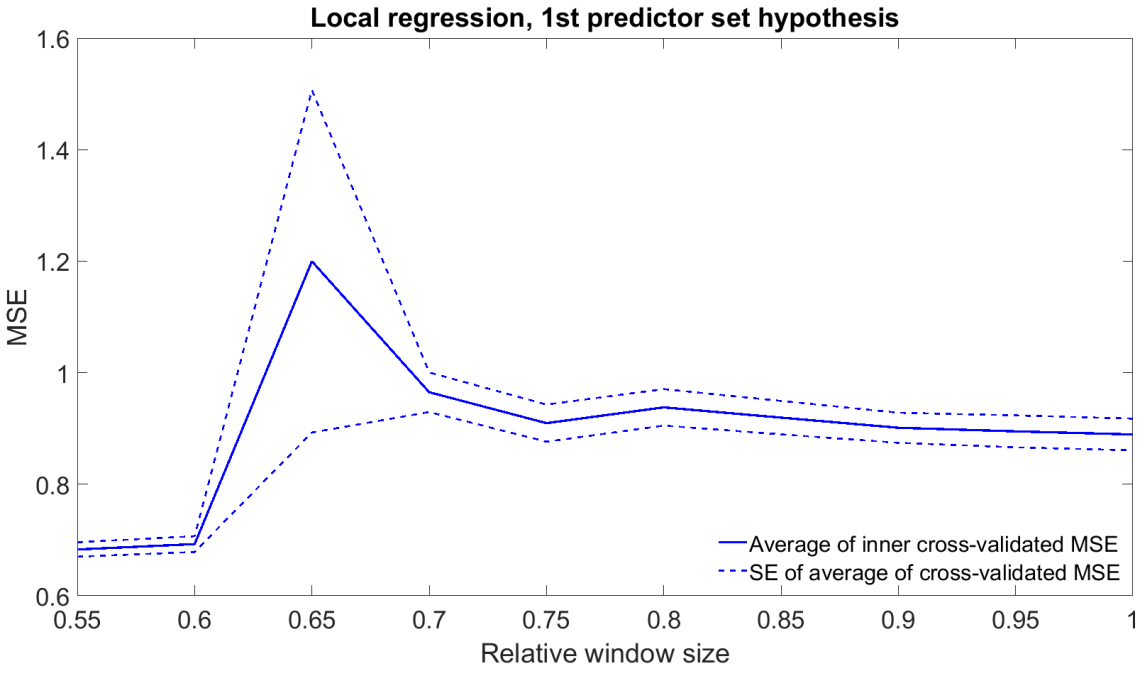

Figure B.4: Average of inner loop MSE_{cv} for local regression models with respect to the relative window size used

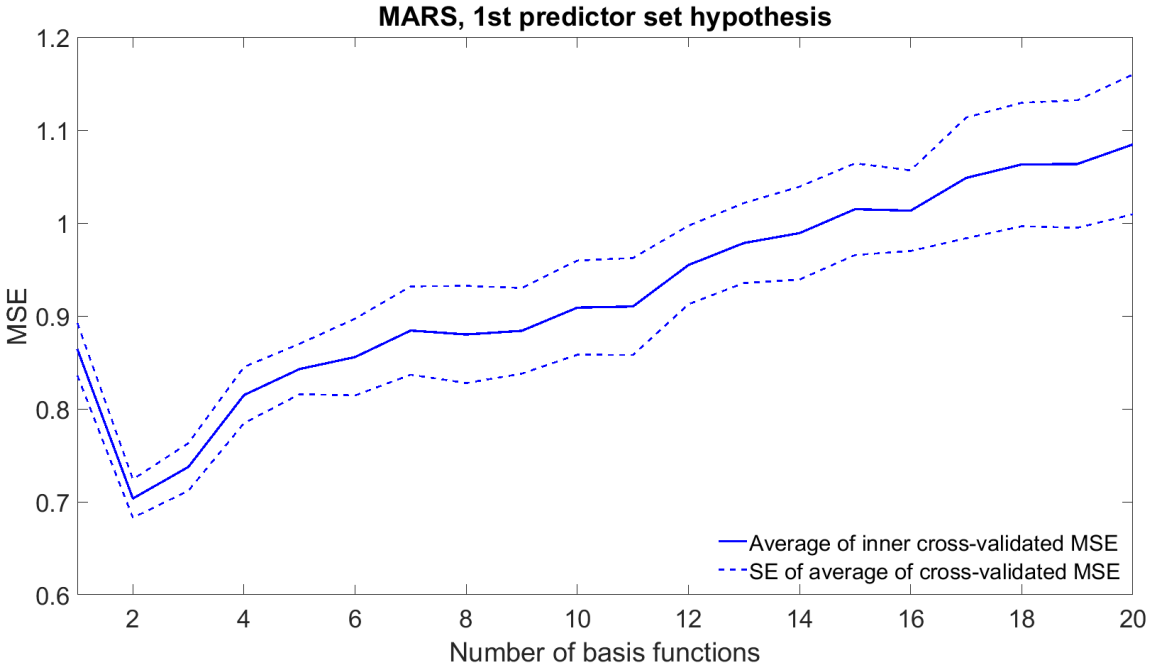

Figure B.5: Average of inner loop MSE_{cv} for MARS models with respect to the number of basis functions used

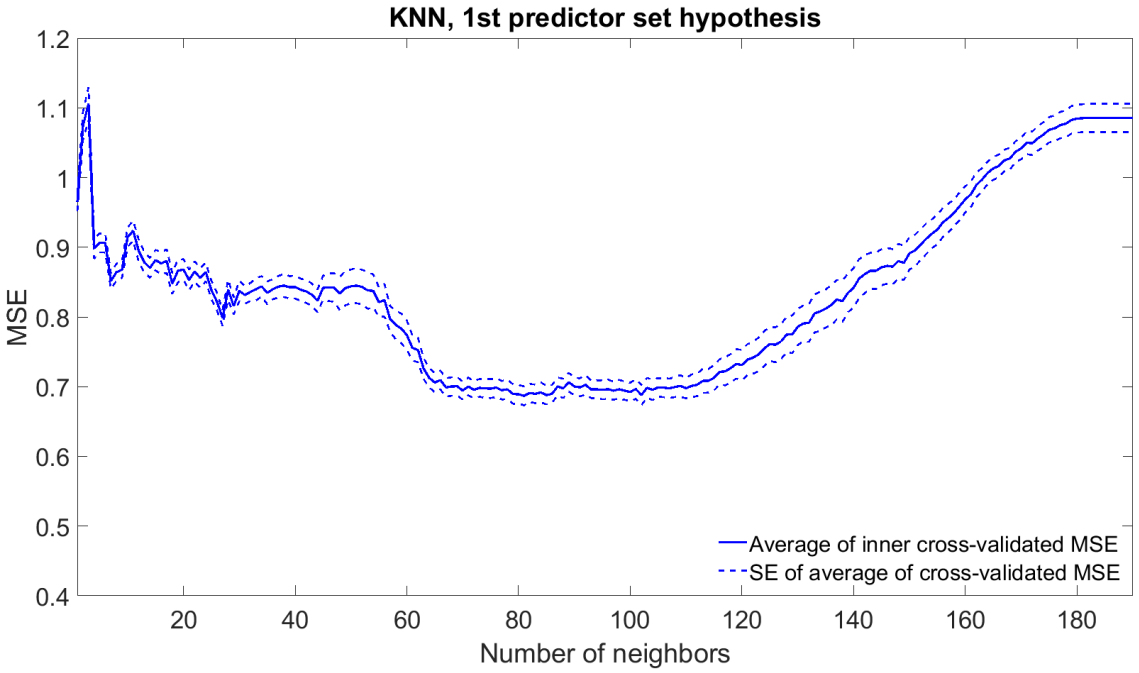

Figure B.6: Average of inner loop MSE_{cv} for KNN models with respect to the number of neighbors used

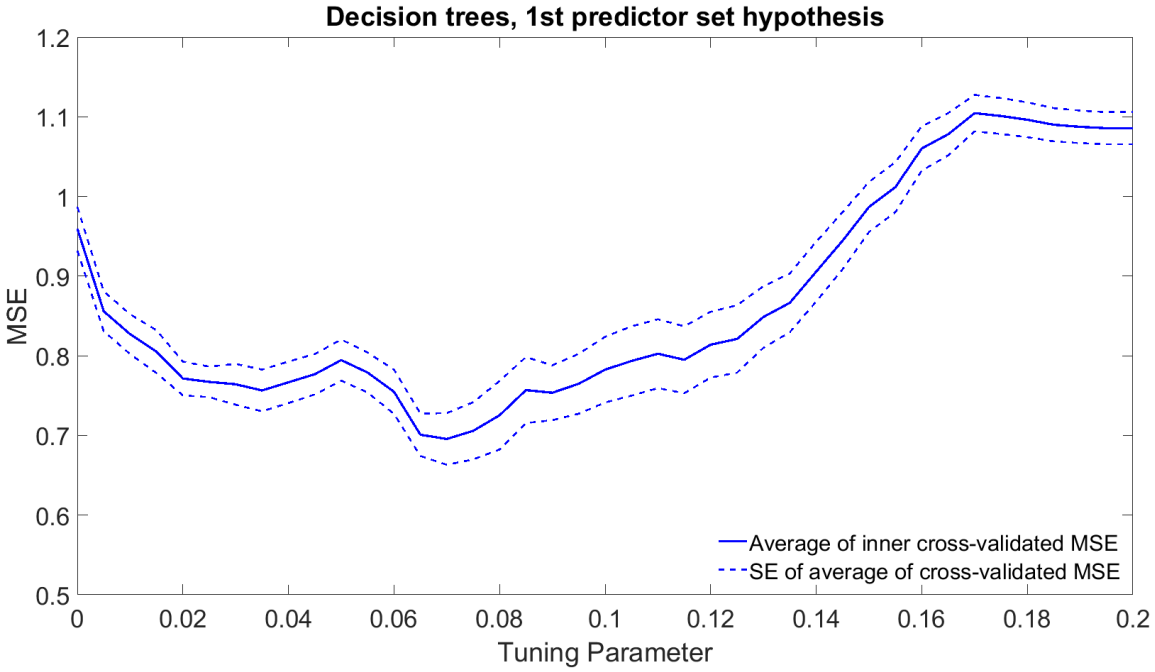

Figure B.7: Average of inner loop MSE_{cv} for decision tree models with respect to the value of the tuning parameter λ used

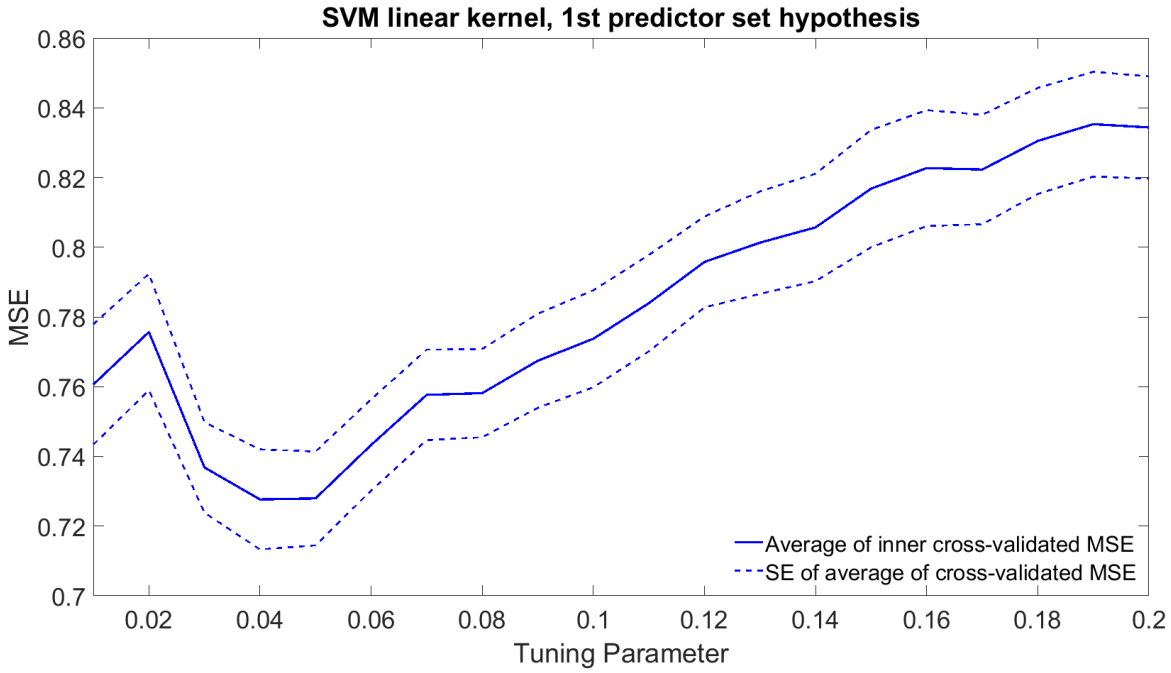

Figure B.8: Average of inner loop MSE_{cv} for SVMs with linear kernel with respect to the tuning parameter C used

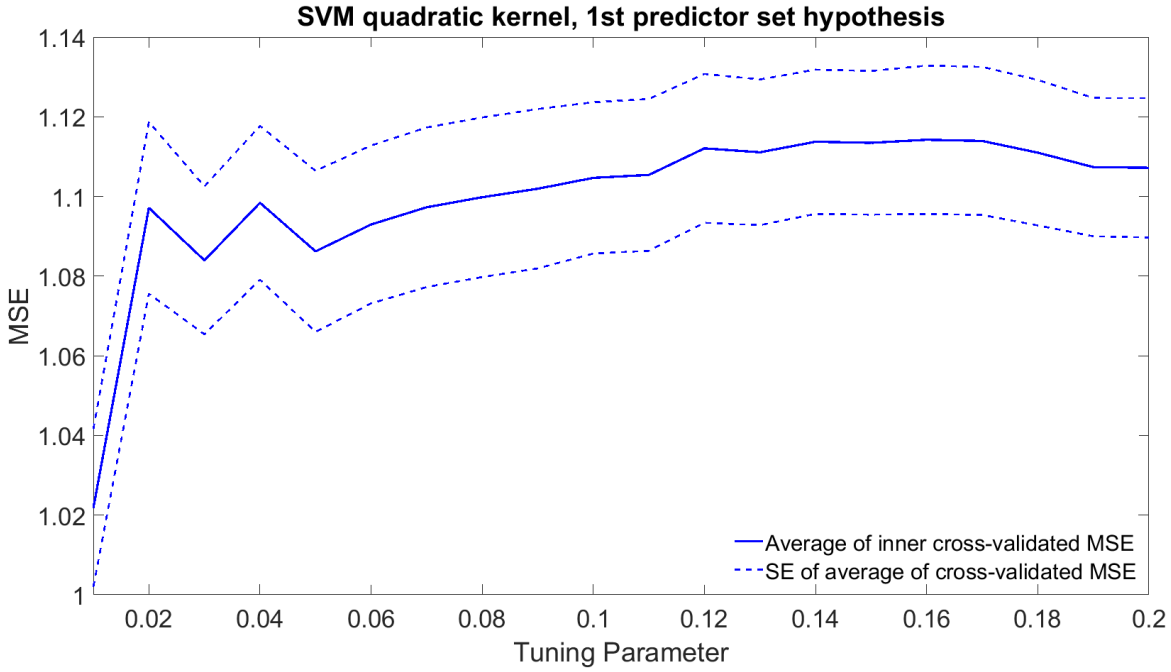

Figure B.9: Average of inner loop MSE_{cv} for SVMs with quadratic kernel with respect to the tuning parameter C used

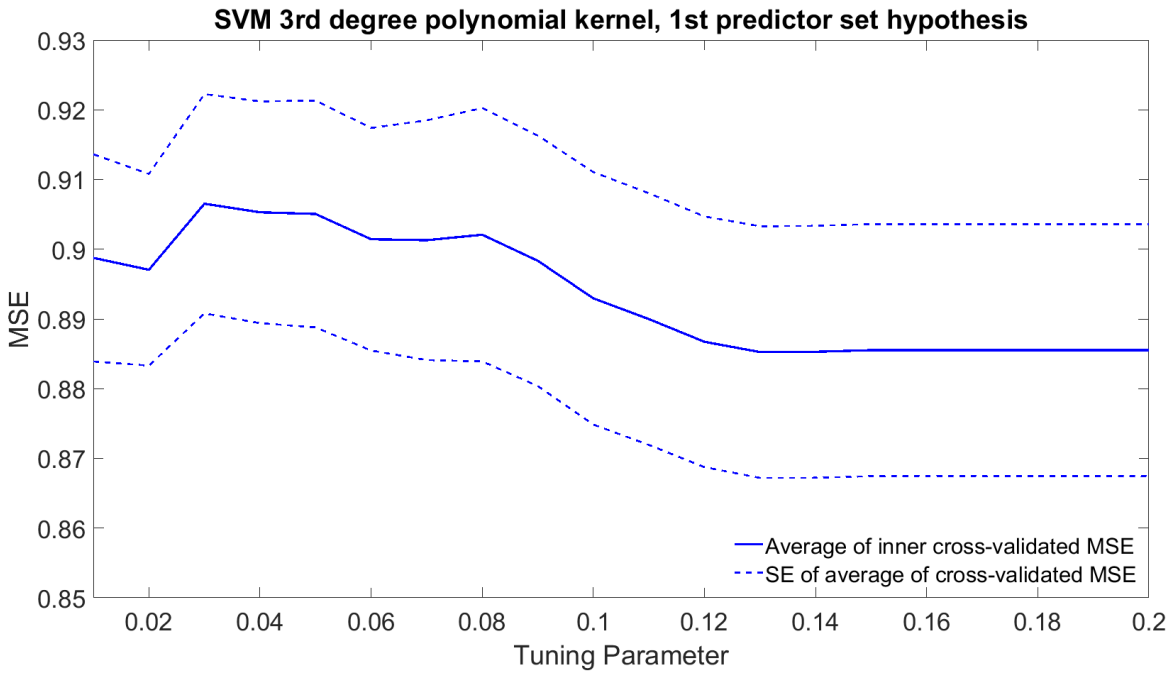

Figure B.10: Average of inner loop MSE_{cv} for SVMs with 3rd degree polynomial kernel with respect to the tuning parameter C used

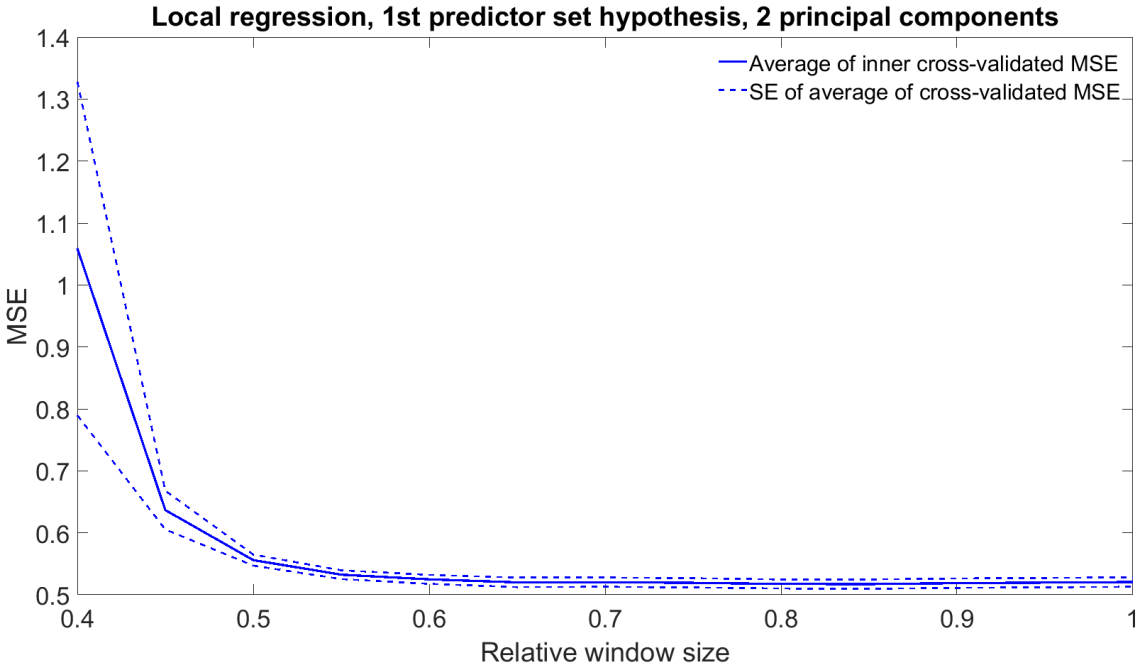

Figure B.11: Average of inner loop MSE_{cv} for local regression applied to first two principal components of first predictor set hypothesis with respect to the relative window size used

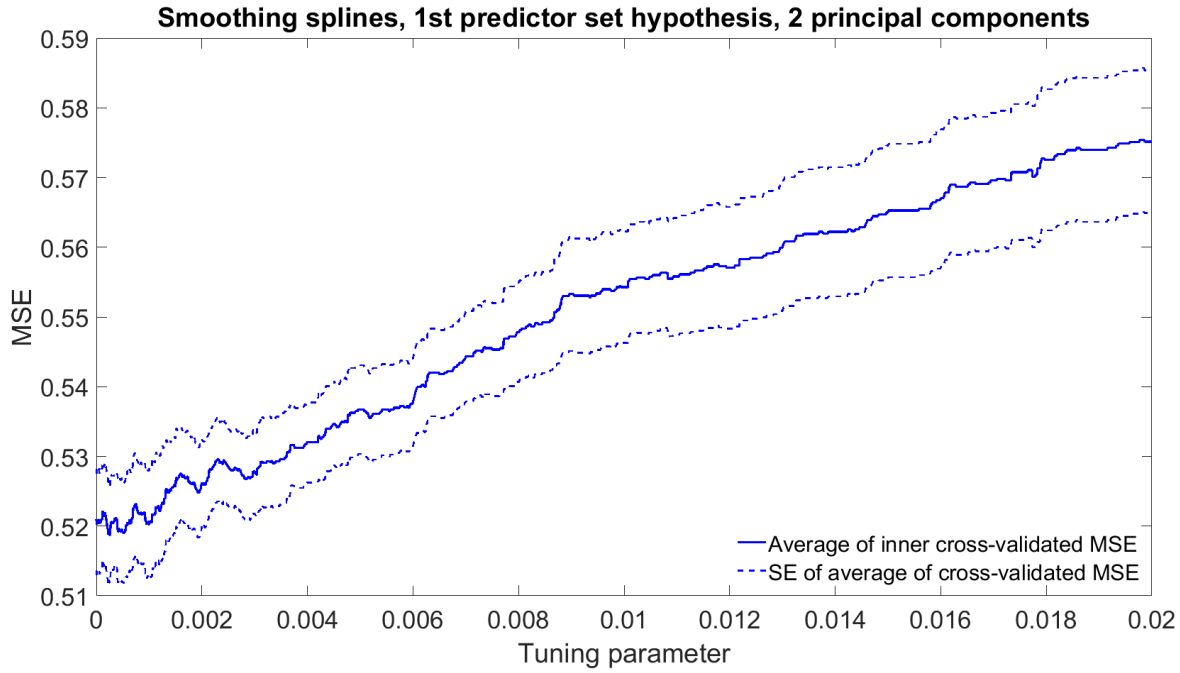

Figure B.12: Average of inner loop MSE_{cv} for smoothing splines applied to first two principal components of first predictor set hypothesis with respect to the value of the tuning parameter λ used

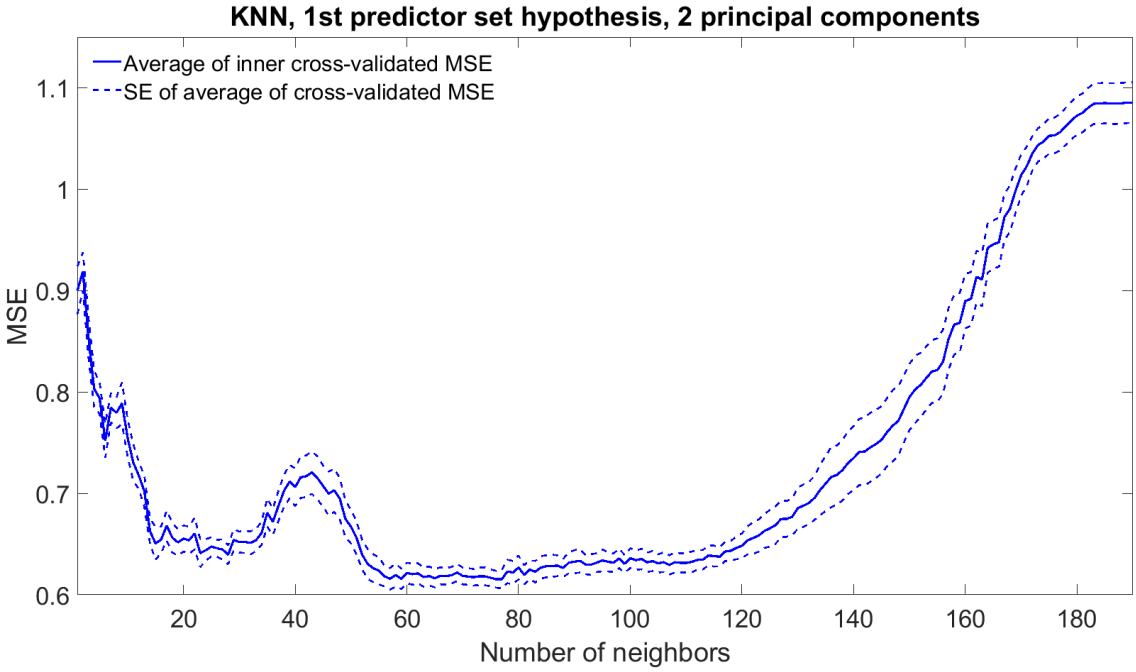

Figure B.13: Average of inner loop MSE_{cv} for KNN applied to first two principal components of first predictor set hypothesis with respect to the number of neighbors used

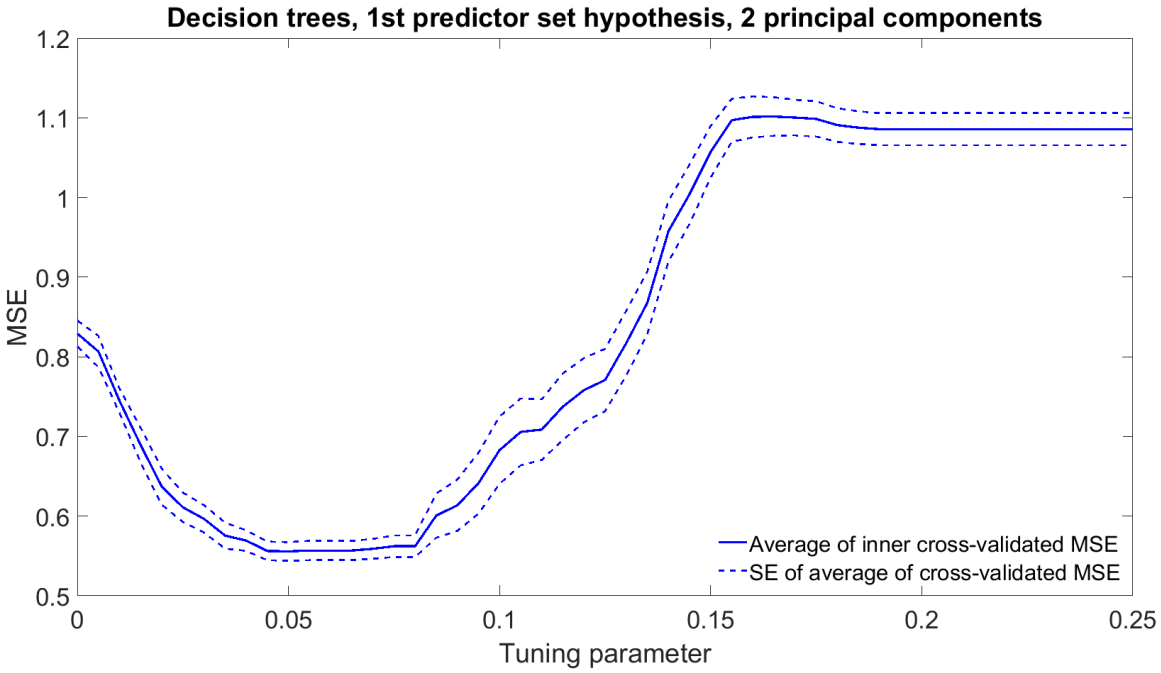

Figure B.14: Average of inner loop MSE_{cv} for decision trees applied to first two principal components of first predictor set hypothesis with respect to the value of the tuning parameter λ used

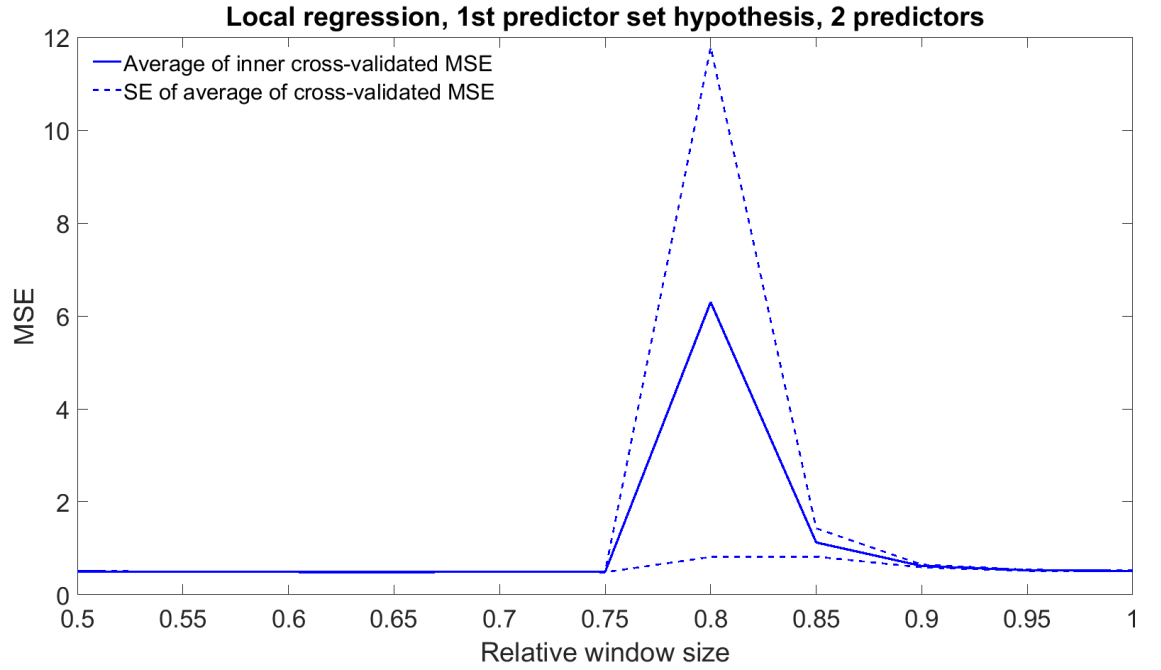

Figure B.15: Average of inner loop MSE_{cv} for local regression models applied to two predictors of first predictor set hypothesis with respect to the relative window size used

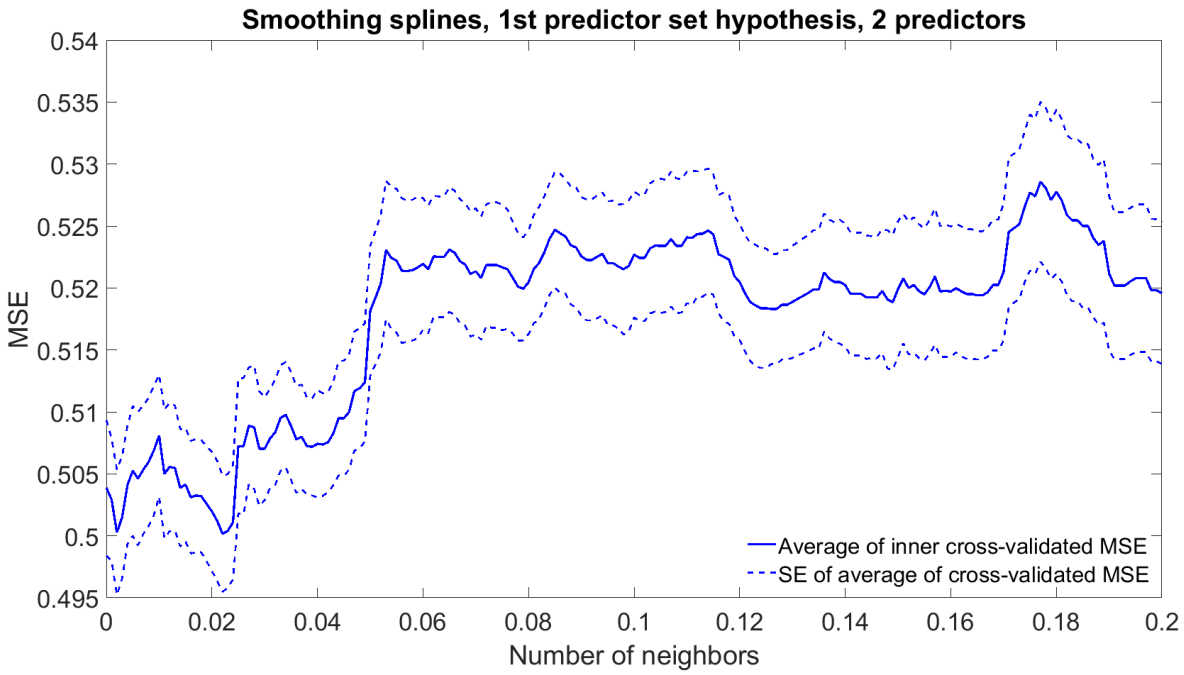

Figure B.16: Average of inner loop MSE_{cv} for smoothing splines applied to two predictors of first predictor set hypothesis with respect to the number of basis functions used

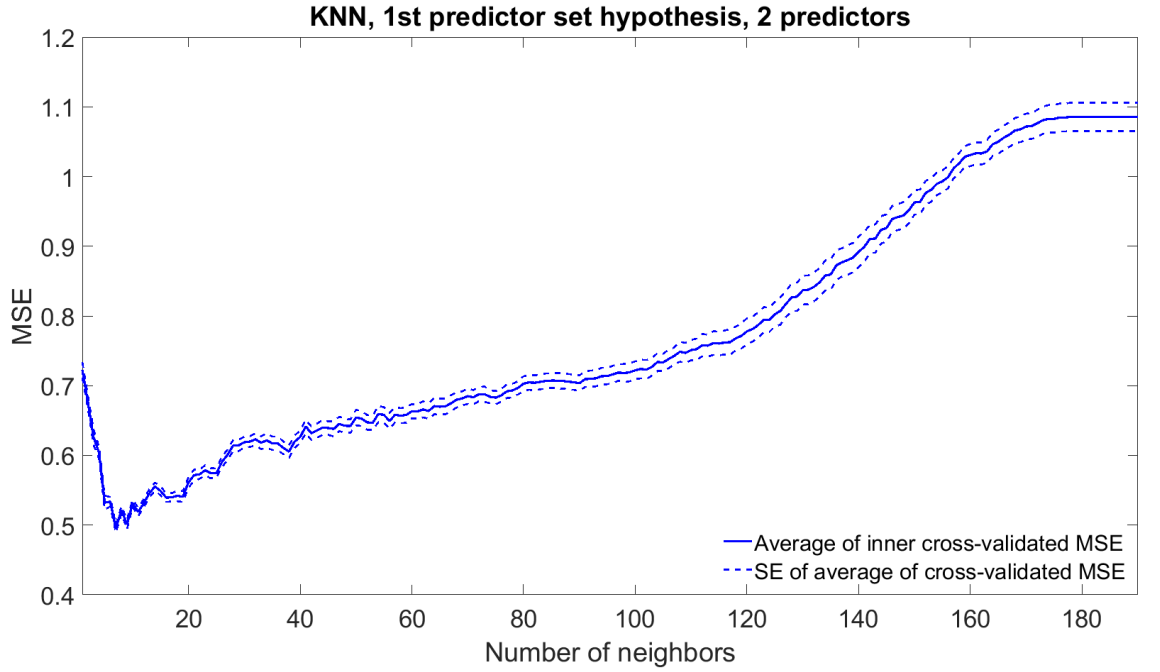

Figure B.17: Average of inner loop MSE_{cv} for KNN models applied to two predictors of first predictor set hypothesis with respect to the number of neighbors used

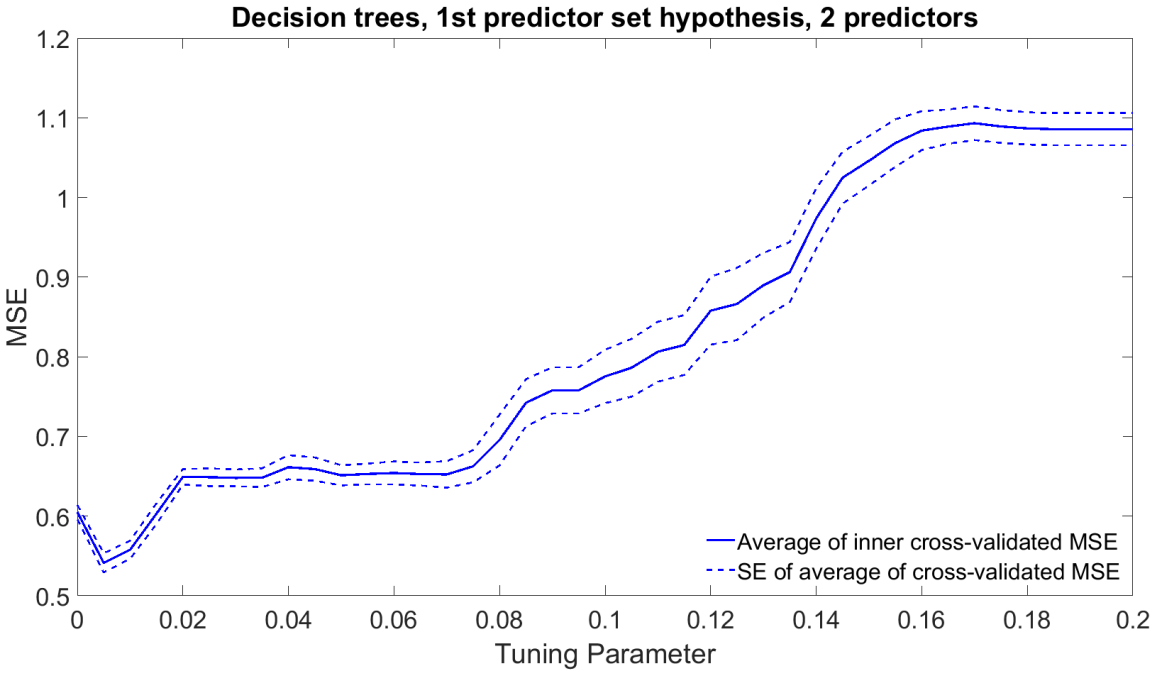

Figure B.18: Average of inner loop MSE_{cv} for decision tree models applied to two predictors of first predictor set hypothesis with respect to the value of the tuning parameter λ used

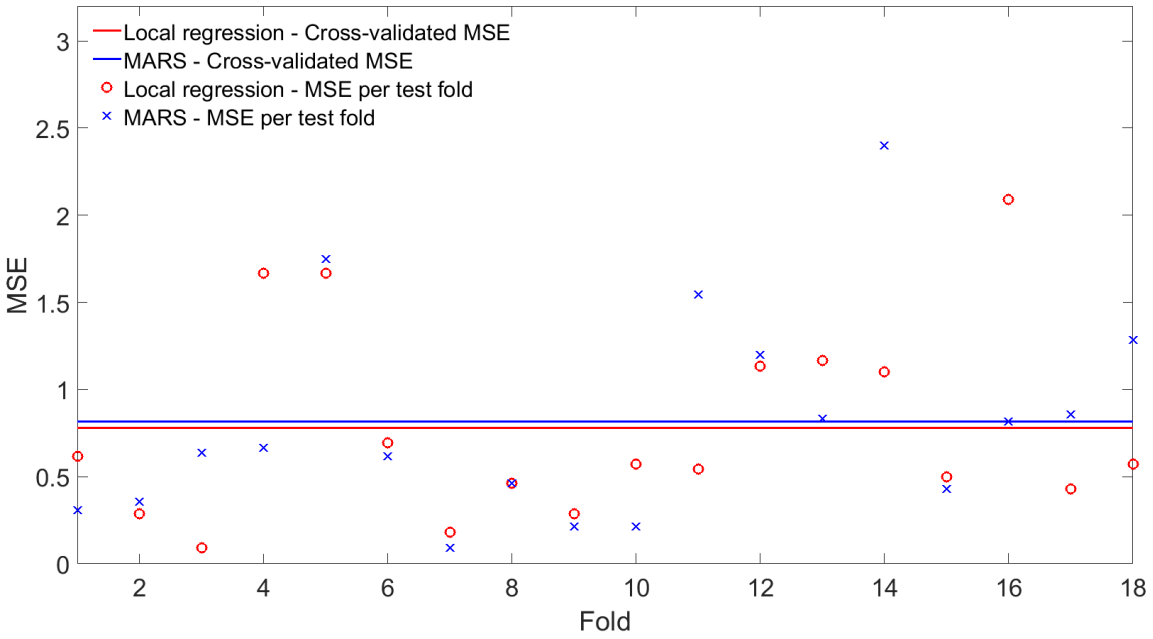

Figure B.19: MSE corresponding to each outer loop test fold and MSE_{cv} for local regression (red) and MARS (blue)

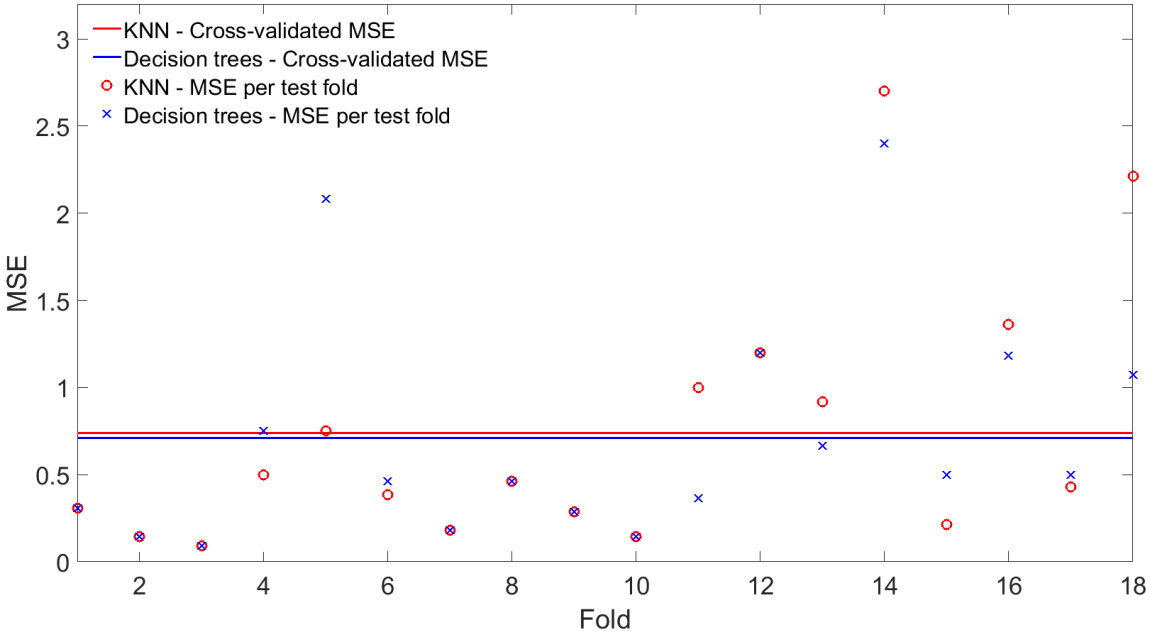

Figure B.20: MSE corresponding to each outer loop test fold and MSE_{cv} for KNN (red) and decision trees (blue)

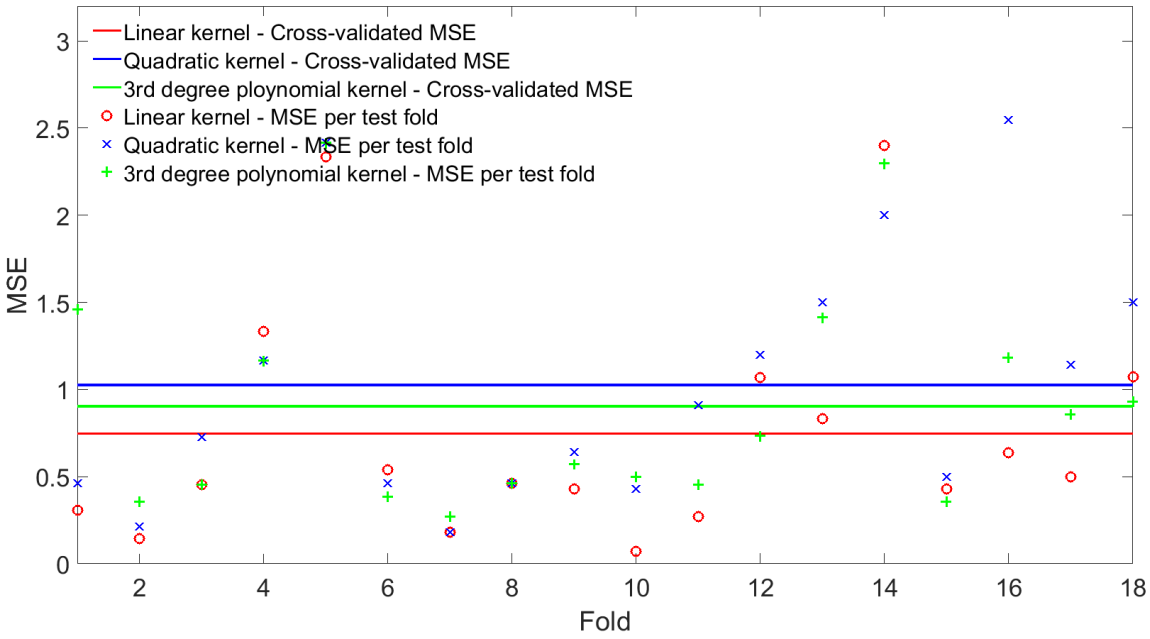

Figure B.21: MSE corresponding to each outer loop test fold and MSE_{cv} for SVMs with linear kernel (red), quadratic kernel (blue) and 3rd degree polynomial kernel (green)

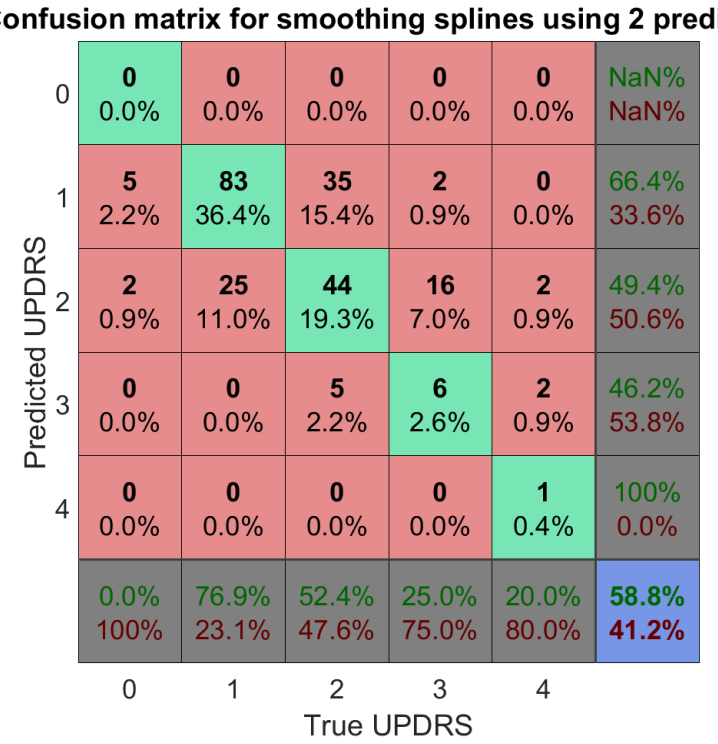

Figure B.22: Confusion matrix for smoothing splines with two predictors, where the UPDRS score for each measurement as predicted by the smoothing splines is related to the median of the physicians' UPDRS scores for the same measurement

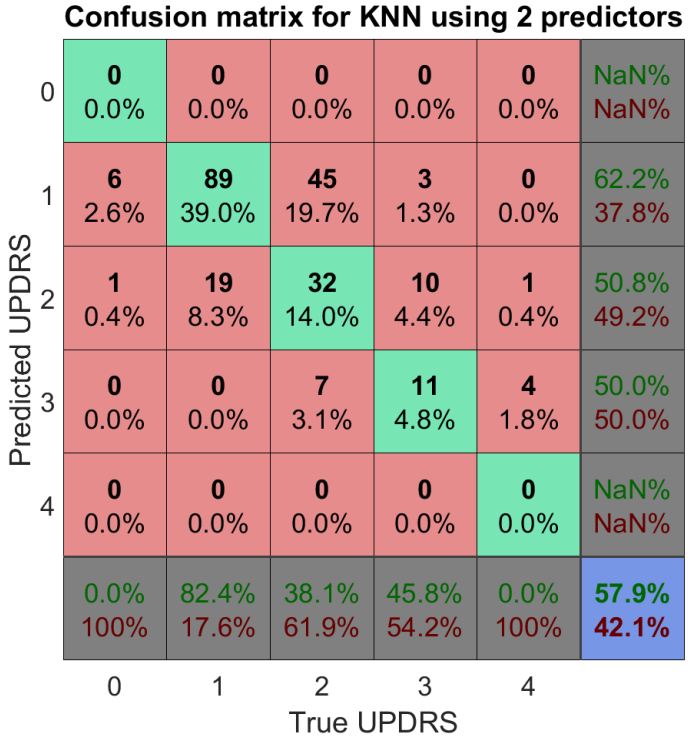

Figure B.23: Confusion matrix for KNN with two predictors, where the UPDRS score for each measurement as predicted by KNN is related to the median of the physicians' UPDRS scores for the same measurement

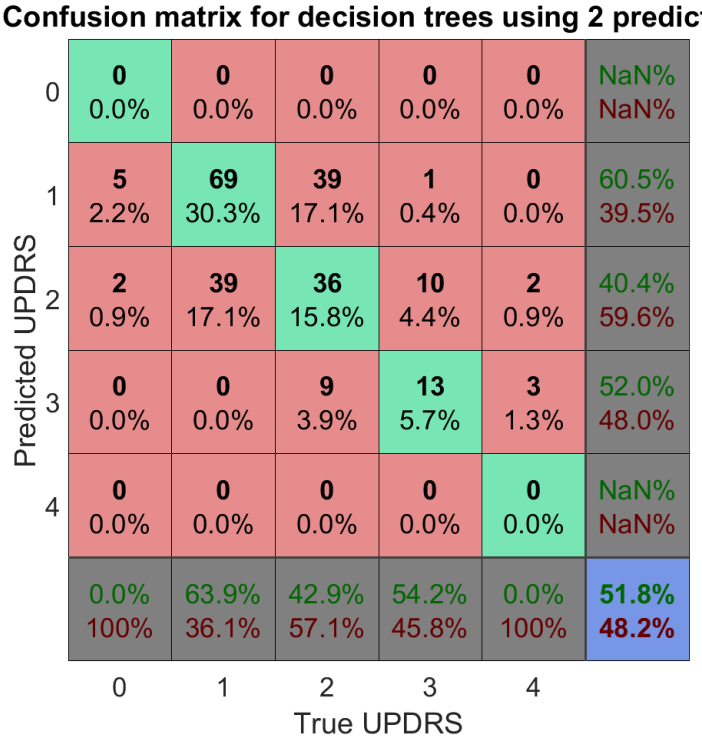

Figure B.24: Confusion matrix for decision trees with two predictors, where the UPDRS score for each measurement as predicted by the decision trees is related to the median of the physicians' UPDRS scores for the same measurement

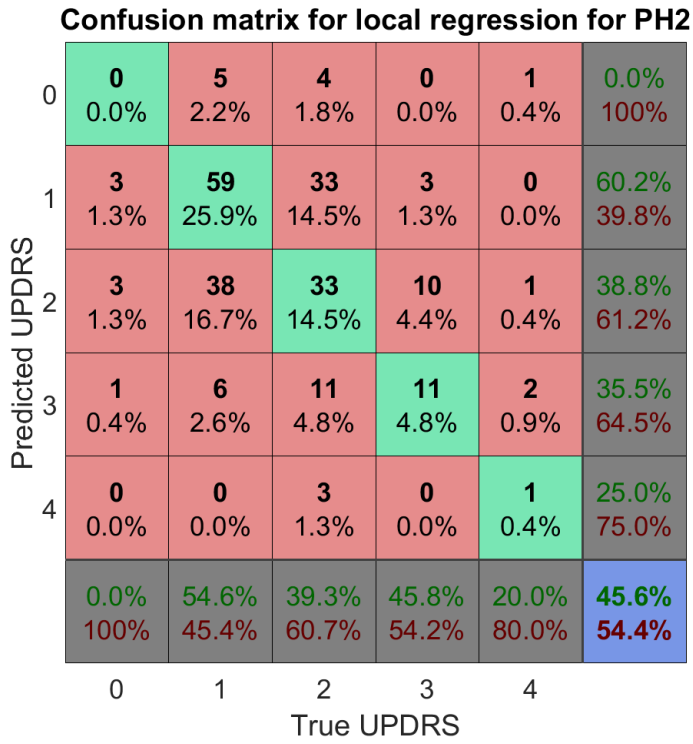

Figure B.25: Confusion matrix for local regression with all predictors of the second predictor set hypothesis, where the UPDRS score for each measurement as predicted by local regression is related to the median of the physicians' UPDRS scores for the same measurement