



Swiss Federal Institute of Technology Zurich

Seminar for  
Statistics

1   **Department of Mathematics**

2

3

---

4

5   Master Thesis

Spring 2022

6

---

7

**Lukas Graz**

8

9                   **Interpolation and Correction**

10                  of

11                 **Multispectral Satellite Image Time Series**

11

---

12

Submission Date: September 18th 2022

13

---

14

Co-Adviser: Gregor Perich  
Adviser: Prof. Dr. Nicolai Meinshausen

# 15 Preface

## 16 Complementary Material

17 Github: <https://github.com/LGraz/MasterThesis-Code>

18 R package: <https://github.com/LGraz/CorrectTimeSeries>

## 19 Acknowledgements

20 First of all, I wish to express my sincere gratitude to my supervisor Prof. Dr. Nicolai  
21 Meinshausen who took the responsibility for my work and happily took the time to discuss  
22 conceptual and guiding questions and to inspire me with new ideas.

23 It is necessary to highlight that without Gregor Perich this project would not have been  
24 possible. His high personal commitment, reliability as well as the weekly instructive su-  
25 pervision meetings were without question essential for this work.

26 It was a real pleasure for me to be part of the *Crop Science* group for this time. Enjoying  
27 everyday company, a two-day excursion, and harvesting wheat together have made this  
28 time truly remarkable. In particular, I would like to thank WALTER, who supported this  
29 collaboration at its core.

30 Last but not least, I would like to express my gratitude to the *Seminar for Statistics*,  
31 which created the framework conditions for this work and did everything to help me with  
32 concetpional and administrative questions. I should also mention the computing resources  
33 provided by them, without which my computations would not have been feasible.

# **Abstract**

- 34
- 35 Kurze problemläuterung (NDVI-ts im Zentrum)
- 36 NDVI Interpolation gewinner
- 37 erforscht Robusification
- 38 NDVI Correction + yield-based evaluation

39 **Contents**

40	<b>Notation</b>	vi
41	<b>1 Introduction</b>	1
42	1.1 XXX motivation - why is it important . . . . .	1
43	1.2 XXX problembaum / fragestellungen . . . . .	1
44	1.3 XXX State-of-the-art . . . . .	1
45	1.4 Roadmap . . . . .	1
46	<b>2 Data and Methods</b>	2
47	2.1 Available Data . . . . .	2
48	2.1.1 Sentinel 2 Satellite Image Data . . . . .	2
49	2.1.2 Harvest Yield Data . . . . .	4
50	2.1.3 Gather Data . . . . .	4
51	2.2 General Methods . . . . .	6
52	2.2.1 MATLAB Matrix Notation . . . . .	6
53	2.2.2 (Relative) RMSE . . . . .	6
54	2.2.3 Out-Of-Bag ( <i>OOB</i> ) and Leave-One-Out-Cross-Validation ( <i>LOOCV</i> ) . . . . .	6
55	2.2.4 Generalized Cross Validation (GCV) . . . . .	6
56	<b>3 Interpolation Methods</b>	7
57	3.1 DAS vs. GDD . . . . .	7
58	3.2 Setting . . . . .	7
59	3.3 Parametric Regression . . . . .	9
60	3.3.1 Double Logistic . . . . .	9
61	3.3.2 Fourier Approximation . . . . .	10
62	3.4 Non-Parametric Regression . . . . .	10
63	3.4.1 Kernel Regression . . . . .	10
64	3.4.2 Kriging . . . . .	11
65	3.4.3 Savitzky-Golay Filter (SG Filter) . . . . .	13
66	3.4.4 Locally Weighted Regression (LOESS) . . . . .	14
67	3.4.5 B-splines . . . . .	15
68	3.4.6 Natural Smoothing Splines . . . . .	16
69	3.5 Tuning parameter estimation . . . . .	16
70	3.6 Robustify . . . . .	17
71	3.6.1 Our Adjustment: . . . . .	18
72	3.6.2 Examples and Conclusions . . . . .	18
73	3.6.3 Upper Envelope Approach - Penalty for Negative Residuals . . . . .	19
74	3.7 Performance Assessment . . . . .	19
75	3.8 XXX Evaluation . . . . .	19
76	<b>4 NDVI Correction</b>	20
77	4.1 Considering other SCL Classes . . . . .	20
78	4.2 Correction . . . . .	22
79	4.2.1 Response and Covariates . . . . .	22
80	4.2.2 Correction Methods . . . . .	22
81	4.2.3 Uncertainty Estimation . . . . .	26
82	4.2.4 Interpolation . . . . .	26

83	4.3 Resulting Interpolation Strategies . . . . .	26
84	4.4 Evaluation Method . . . . .	27
85	4.4.1 Idea . . . . .	27
86	4.4.2 Yield Estimation . . . . .	27
87	<b>5 Results</b>	<b>29</b>
88	5.1 XXX small recap from “Interpolation Methods” . . . . .	29
89	5.2 Robustification and NDVI-Correction . . . . .	29
90	<b>6 Discussion</b>	<b>30</b>
91	6.1 Interpolation Methods . . . . .	30
92	6.2 NDVI Correction . . . . .	30
93	6.2.1 Do we need to separate test and training data strictly by year? . . . . .	30
94	6.2.2 Shall We Use Additional Covariates? . . . . .	30
95	6.2.3 High RMSE in Yield Prediction . . . . .	31
96	<b>7 Summary</b>	<b>32</b>
97	7.1 Future Work . . . . .	32
98	7.1.1 Time Series Correction-Interpolation as a General Method . . . . .	32
99	7.1.2 Minor Improvements . . . . .	33
100	<b>Bibliography</b>	<b>34</b>
101	<b>A Reproducibility</b>	<b>35</b>
102	A.1 Reproduce Results . . . . .	35
103	A.2 R-Package . . . . .	35
104	<b>B Further Material</b>	<b>37</b>
105	B.1 Interpolation . . . . .	37
106	B.2 NDVI correction . . . . .	37

107 **Todo list**

108	definithion here and also relative . . . . .	6
109	definithion here and explenation why (computational cheap) . . . . .	6
110	TODO: include Weighted versions . . . . .	10
111	write out keywords, after final results . . . . .	19
112	shoud w write 1:1 the sam es in the end of section 3 . . . . .	29
113	where does this section belong to? Chapter ‘NDVI Correction’ or ‘Further Work’? . . . . .	30

# 114 Notation

## 115 Conventions for Variables

- 116  $c$ : a (vector of) constant(s)
- 117  $\lambda \in \mathbb{R}$ : a scalar
- 118  $n \in \mathcal{N}$ : sample size
- 119  $i, j$  are indices in  $\{1, \dots, n\}$
- 120  $x \in \mathbb{R}^n$ : covariate in 1-dim interpolation setting
- 121  $w \in \mathbb{R}^n$ : a vector of weights for each location  $x$
- 122  $y \in \mathbb{R}^n$ : response in 1-dim interpolation setting
- 123  $\hat{y} \in \mathbb{R}^n$ : estimate of  $y$
- 124  $\bar{y} \in \mathbb{R}$ : mean of  $y$
- 125  $r \in \mathbb{R}^n$ : residuals given by  $y - \hat{y}$

## 126 Abbreviations and Objects

- 127 Pixel: A pixel describes a specific location in a field. It has the size of 10 x 10 meters and coincides with the resolution (and location) of the sentinel-2 pixels. Such pixels are illustrated in figure 2.2b. Additional information like yield is also attached.
- 130  $P_t$ : this describes the observed data (weather and spectral bands) at time  $t$  and the location of one pixel.
- 132  $P$ : a pixel. We see it as a collection of all the observations at the specified location within one season. More formally,  $P := \{P_t | t \text{ is a valid sample time within a defined season}\}$
- 134 SCL: scene classification layer. This indicates what one can expect at a pixel at a sampled time. For an overview cf. table 2.2
- 136  $P^{SCL45}$ : similar to  $P$  but we only consider observations which belong to the classes 4 and 5. This is used done to get a subset of observations which are less contaminated by clouds and shadows.
- 139 NDVI: normalized vegetation difference index
- 140 DAS: days after sowing
- 141 GDD: growing degree days – cumulative sum of  $(\text{temperature} - \text{threshold})^+$

<sup>142</sup> XXX ML models and their shortnames

<sup>143</sup> RYEA : relative yield-estimation-accuracy. Definition [4.4.0.1](#)

<sup>144</sup> OOB : out-of-box. Describes the procedure if we estimate the value for a point but not  
<sup>145</sup> consider the point itself (cf. section [2.2.3](#))

146 **Chapter 1**

147 **Introduction**

148 **1.1 XXX motivation - why is it important**

- 149 - NDVI-timeseries is very simple and widely used. Examples are: - Plant Models REF  
150 - Season Start (start of spring) (community name: land-surface-plant-phenology) - Yield  
151 prediction - crop classification  
152 - NDVI is not only of interest to researchers but also public agents and insurance companies  
153 Since satellite images are “for free” researchers extract it

154 **1.2 XXX problembaum / fragestellungen**

155 problem schilderung anhand referenzen und evtl. eines bileds:

156 **1.3 XXX State-of-the-art**

- 157 zusammenfassung mit literaturrecherche hier (jetzige antowrt auf problemstellung):  
158 — Doublelogistic (winter-ndvi)  
159 — parametric / non-parametric approaches  
160 — spatio-temporal approaches

161 **1.4 Roadmap**

162 XXX

163 **Chapter 2**

164 **Data and Methods**

165 **2.1 Available Data**

166 Our study region is a farm of over 800ha, which is located in western Switzerland. From  
167 REF-gregor we acquire satellite image data (section 2.1.1), yield maps of several cereals  
168 from 2017 to 2021 (section 2.1.2), and meteorological data (section 2.1.3).

169 **2.1.1 Sentinel 2 Satellite Image Data**

170 **General Information**

171 The European Space Agency (ESA)<sup>1</sup> freely distributes the high-quality images of the two  
172 Sentinel satellites 2 (S2). Together, both satellites have a revisit time of 5 days at the  
173 equator and 2-3 at mid-latitudes. However, in our study region, we only receive an image  
174 every 5 days. In order to decrease the effect of atmospheric conditions like reflections  
175 and scattering, we will not work with the raw data but with the results of the Level-2A  
176 processing<sup>23</sup>.

177 **Data Description**

178 The Level-2A processed images we use contain 12 spectral bands with local resolutions up  
179 to 10 meters (see 2.1). Bands which have a lower resolution (20 and 60 meters) will be  
180 scaled up to 10 meters using cubic interpolation (REF gregor perich). Additional to the  
181 spectral bands, the ESA also supplies a Scene Classification Layer (*SCL*) where for each  
182 location the observed subject is assigned to an *SCL-class* (cf. table 2.2). In chapter 3 we  
183 will use this classification to filter out unreliable data points, considering only SCL-classes  
184 4 and 5.

185 **Data Illustration**

186 The figure 2.1 shows a selection of 6 satellite images of a field, which display our challenges.  
187 In February (image(a)), as expected, we see no vegetation but bare soil. At the beginning

---

<sup>1</sup>REF: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2>

<sup>2</sup>REF <https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithms>

<sup>3</sup>XXXREF gregor perich “Data prior to March 2018 was only available in the top-of-atmosphere L1C format and was downloaded as such [...] L1C data was processed to L2A product level using the ‘Sen2Cor’ processor provided by ESA”

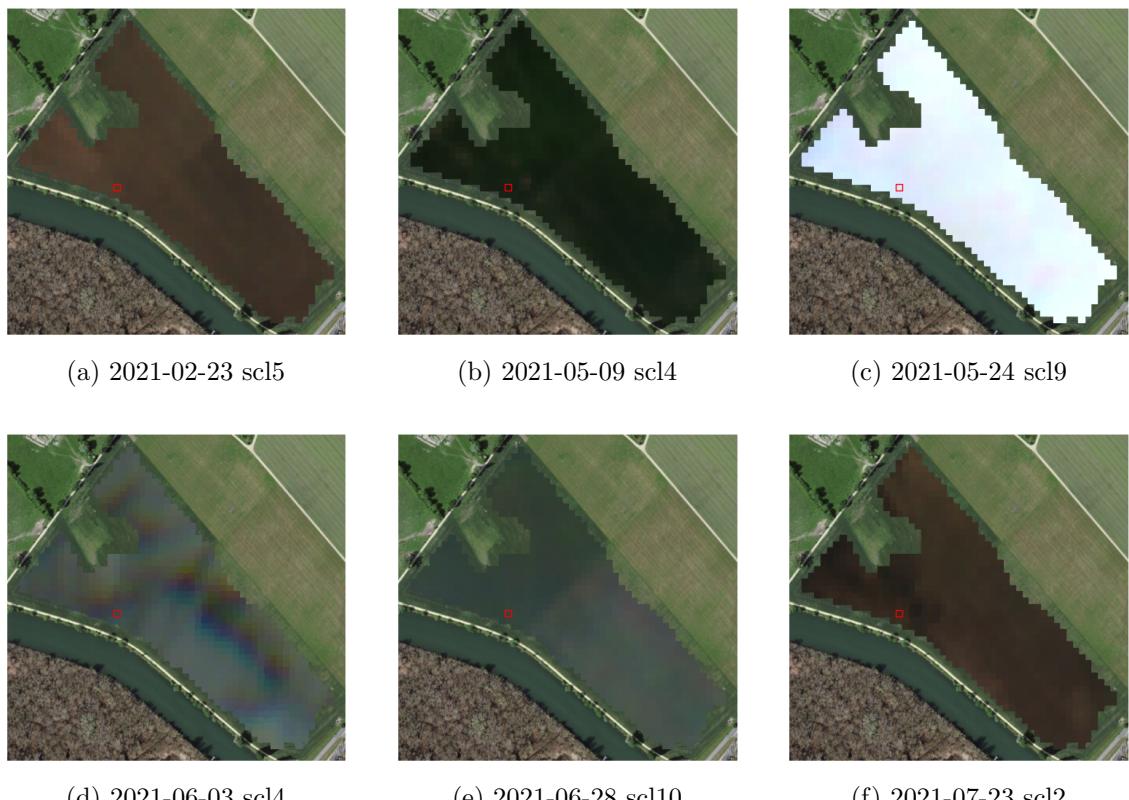


Figure 2.1: Satellite images of a field at selected times with a static background for orientation. The SCL-class of the highlighted pixel is provided in the respective subtitle. (???xxx include scl legend?)

Table 2.1: Jaramaz, Perović, Belanovic Simic, Saljnikov, Cakmak, Mrvić, and Zivotic (Jaramaz et al.) List of spectral bands of the S2-satellites. Each band has its center at the wavelength  $\lambda$  in nm with the spectral width  $\Delta\lambda$  in nm with a spatial resolution  $SR$  in m.

Band	$\lambda$	$\Delta\lambda$	$SR$	Purpose
1	443	20	60	Atmospheric correction (aerosol scattering)
2	490	65	10	Sensitive to vegetation senescing, carotenoid, browning and soil background; atmospheric correction (aerosol scattering)
3	560	35	10	Green peak, sensitive to total chlorophyll in vegetation
4	665	30	10	Maximum chlorophyll absorption
5	705	15	20	Position of red edge; consolidation of atmospheric corrections / fluorescence baseline.
6	740	15	20	Position of red edge, atmospheric correction, retrieval of aerosol load.
7	783	20	20	Leaf Area Index (LAI), edge of the Near-Infrared (NIR) plateau.
8	842	115	10	LAI
8a	865	20	20	NIR plateau, sensitive to total chlorophyll, biomass, LAI and protein; water vapor absorption reference; retrieval of aerosol load and type.
9	945	20	60	Water vapor absorption, atmospheric correction.
10	1375	30	60	Detection of thin cirrus for atmospheric correction.
11	1610	90	20	Sensitive to lignin, starch and forest above ground biomass. Snow/ice-/cloud separation.
12	2190	180	20	Assessment of Mediterranean vegetation conditions. Distinction of clay soils for the monitoring of soil erosion. Distinction between live biomass, dead biomass and soil, e.g. for burn scars mapping.

188 of May, we observe a cloudless dark green field. In (c) it is obvious that we have no chance  
 189 to get useful information when there is a heavy cloud cover. Figure (d) shows that the  
 190 SCL classification is not reliable, since we evidently observe clouds. In (e) we see a pale  
 191 green. This likely shimmers through cirrus clouds.

### 192 2.1.2 Harvest Yield Data

193 The crop yield data were collected using a combine harvester. Equipped with GPS, the  
 194 harvester drives over the fields and continuously estimates the crop density in t/ha (see fig.  
 195 2.2a). We take the data set derived from this in REF-Gregor-Perich, where error-prone  
 196 measurement points (such as during an egen curve) were removed and then the yield map  
 197 was rasterized using linear interpolation (cf. fig. 2.2b).

198 Comparing the manually weighted yield and the sum of estimated raster (per field per  
 199 year) we note a discrepancy of about 10% (cf. REF-gregor). Since the relative estimation  
 200 error is rather constant and we do not aim to estimate the absolute yield we will not  
 201 consider this deviation.

### 202 2.1.3 Gather Data

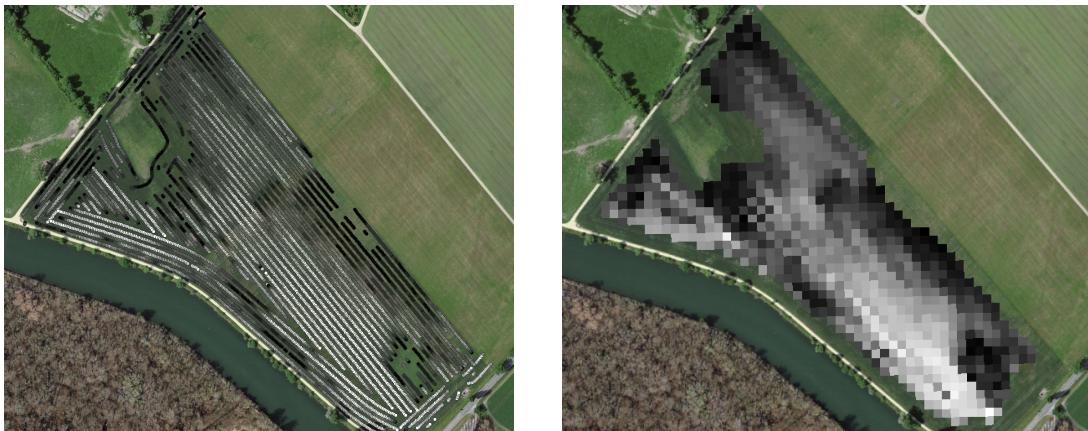
203 Before we join all the data, we define a few concepts.

204 Using bands  $B4$  and  $B8$ , we calculate the well-known Normalized Difference Vegetation  
 205 Index ( $NDVI$ ) using the formula: (???REF nötig?)

$$NDVI = \frac{B8 - B4}{B8 + B4}$$

Table 2.2: Overview: Scene Classification Layers (SCL)

No.	Class	Color
0	No Data (Missing data on projected tiles) (black)	
1	Saturated or defective pixel (red)	
2	Dark features / Shadows (very dark gray)	
3	Cloud shadows (dark brown)	
4	Vegetation (green)	
5	Bare soils / deserts (dark yellow)	
6	Water (dark and bright) (blue)	
7	Cloud low probability (dark gray)	
8	Cloud medium probability (gray)	
9	Cloud high probability (white)	
10	Thin cirrus (very bright blue)	
11	Snow or ice (very bright pink)	



(a) obtained by a combine harvester (cleaned)

(b) rasterized to Sentinel 2 resolution.

Figure 2.2: Crop yield density map of a field. Ranges from 0.1 t/ha (black) to 5.35 t/ha (white)

206 Note that we call the calculated values merely the *observed NDVI*, as we must be aware  
 207 of imprecisions due to clouds and shadows.

208 To define a timescale, we consider Days After Sowing (*DAS*) and a transformed timescale,  
 209 Growing Degree Days (*GDD*) ([McMaster and Wilhelm](#) ([McMaster and Wilhelm](#))). The  
 210 latter are defined as the cumulative sum (since sowing) of temperature above a given base  
 211 temperature  $T_{base}$ <sup>4</sup>. Thus, the GGD for  $n$  days after sowing will be equal to:

$$GDD_n := \sum_{i=0}^n \max(T_i - T_{base}, 0).$$

212 Now we create a data set, which will contain all the necessary information. Given that we  
 213 have the spectral data at a  $10m \times 10m$  resolution, we introduce the concept of a Pixel. A  
 214 *Pixel P* is associated with a  $10m \times 10m$  square defined by the S2 satellites and contains  
 215 all relevant information for a season and this location. More precisely,  $P$  is a collection  
 216 of general information (like yield and coordinates) and all associated  $P_t$  of a given season.  
 217 Where  $P_t$  represents a tuple of the spectral data for time  $t$ , the NDVI calculated from it,

<sup>4</sup>XXX For cereals we use  $T_{base} = 0$

218 and the associated GDD. We will call the resulting data set *PIXELS* as it is the collection  
 219 of all Pixels (over all seasons).

220 Finally, we split *PIXELS* randomly into a train (80%) and test (20%) set.

## 221 2.2 General Methods

222 We will only introduce general methods within this section, whereas more specific methods  
 223 will be introduced in their context. We discuss interpolation methods in sections 3.3 and  
 224 3.4, a robustification strategy in section 3.6, a method how we can objectively determine  
 225 the quality of an interpolation in section 3.5, and in section 4.2 we present the NDVI  
 226 correction with an adapted interpolation strategy.

### 227 2.2.1 MATLAB Matrix Notation

228 We will use the MATLAB ‘:’ notation to indicate rows and columns of a matrix. That is  
 229 if  $X \in \mathbb{R}^{n \times p}$  is a matrix, then  $X_{[:,3]}$  is the 3rd column of  $X$  and  $X_{[2,:]}$  is the second row of  
 230  $X$ .

### 231 2.2.2 (Relative) RMSE

232 definitition here and also relative

### 233 2.2.3 Out-Of-Bag (*OOB*) and Leave-One-Out-Cross-Validation (*LOOCV*)

Let

$$D = \{(X_{[j,:]}, y_j) \mid X \in \mathbb{R}^{n \times p}, y \in \mathbb{R}^n, j = 1, \dots, n\}$$

234 be a dataset,  $i \in \{1, \dots, n\}$  and  $M^{(-i)}$  a model fitted on a subset of  $D \setminus \{(X_{[i,:]}, y_i)\}$ . Then  
 235 we call  $\hat{y}_i := M^{(-i)}(X_{[i,:]})$  an *OOB* estimator of  $y_i$ . If we do this for all  $i \in \{1, \dots, n\}$ , we  
 236 obtain  $\hat{y} := (\hat{y}_1, \dots, \hat{y}_n)$  the OOB estimator for  $y \in \mathbb{R}^n$ .

237 In the bootstrap (e.g., random forest) framework, we define  $\hat{y}_i$  to be the average of all  
 238 computed and admissible  $M^{(-i)}$ .

239 In the case that  $M^{(-i)}$  was fitted on the set  $D \setminus \{(X_i, y_i)\}$  (i.e., not a true subset), we call  
 240 the corresponding  $\hat{y}_i$  also the LOOCV estimator.

241 If we optimize some parameter via OOB (or LOOCV) this means that we compute

### 242 2.2.4 Generalized Cross Validation (GCV)

243 definitition here and explenation why (computational cheap)

244 **Chapter 3**

245 **Interpolation Methods**

246 In this section, we take a closer look at several interpolation methods, which will be  
247 used to interpolate and smooth the NDVI time series, while considering only SCL45 in  
248 this chapter. A brief overview of the considered interpolation methods can be found in  
249 table 3.1.

250 First, we define the general setting and discuss a general approach to make the interpola-  
251 tion more robust (i.e. reduce the impact of outliers).

252 Afterwards, we introduce and discuss each method.

253 Then, we try to extract the main ingredients of each method to forge our own one.

254 Finally, using leave-one-out cross-validation, we tune the parameters (where necessary)  
255 and get a first idea of the performance of each method.

256 **3.1 DAS vs. GDD**

257 Prior to interpolating the NDVI time series, we should decide on a timescale. We can  
258 choose between DAS and GDD (cf. section 2.1.3 and equation 2.1.3). In figure 3.1 we see  
259 an example for comparison of the two. Here we see that the first 120 DAS are compressed  
260 to just 500 GDD. This has several advantages. First, it makes the scales comparable (in  
261 terms of plant growth) because the plants are not concerned with the month of the year but  
262 the current temperature. Second, in winter we tend to have higher cloud cover and thus  
263 fewer SCL45 observations. Hence, this gap in observations is compressed. Consequently,  
264 we will only use GDD in the subsequent.

265 **3.2 Setting**

We are given data in the form of  $(x_i, Y_i)$  for  $i = 1, \dots, n$ . Assume that it can be represented by

$$y_i = m(x_i) + \varepsilon_i,$$

where  $\varepsilon_i$  is some noise and  $m : \mathbb{R} \rightarrow \mathbb{R}$  is some (parametric or non-parametric) function.  
If we assume that  $\varepsilon_1, \dots, \varepsilon_n$  i.i.d. with  $\mathbb{E}[\varepsilon_i] = 0$  then

$$m(x) = \mathbb{E}[y | x]$$

Table 3.1: A short summary of the studied interpolation methods. Important assumptions are stated, pros/cons are listed and it is indicated whether the method supports weighted observations (w) and if the resulting interpolation is bounded w.r.t. a fixed interval (b).

	<b>assumptions</b>	<b>pros</b>	<b>cons</b>	<b>w</b>	<b>b</b>
Savitzky-Golay filter	<ul style="list-style-type: none"> <li>- High frequencies are noise (Low-Pass-Filter)</li> <li>- Equidistant points</li> <li>- Local polynomials</li> </ul>	<ul style="list-style-type: none"> <li>- Computationally very fast</li> </ul>	<ul style="list-style-type: none"> <li>- Cannot deal natively with missing data (need some interpolation)</li> </ul>	No	(Yes)
SG + NDVI	<ul style="list-style-type: none"> <li>- Upper envelope</li> <li>- Vegetation cannot grow faster than some slope</li> </ul>	<ul style="list-style-type: none"> <li>- Biological edge</li> </ul>	<ul style="list-style-type: none"> <li>- Bad “upper envelope” since weights are not used for the estimation itself</li> </ul>	(No)	(Yes)
Loess	<ul style="list-style-type: none"> <li>- Local polynomial with points closer to the estimated point are more important</li> </ul>	<ul style="list-style-type: none"> <li>- Flexible</li> <li>- Generalization of SG</li> <li>- Weighting function makes intuitive sense</li> </ul>	<ul style="list-style-type: none"> <li>- Computationally expensive</li> </ul>	Yes	(Yes)
Smoothing Splines	<ul style="list-style-type: none"> <li>- 2cd derivative of function is integrable</li> </ul>	<ul style="list-style-type: none"> <li>- Intuitive meaning of penalty</li> <li>- General assumptions</li> <li>- Flexible shape</li> </ul>	<ul style="list-style-type: none"> <li>- Unbounded</li> </ul>	Yes	No
B-Splines (Smoothed)	<ul style="list-style-type: none"> <li>- Function can be approximated by a linear combination of B-splines basis functions</li> </ul>	<ul style="list-style-type: none"> <li>- General assumption</li> <li>- Flexible shape</li> </ul>	<ul style="list-style-type: none"> <li>- Unbounded</li> <li>- No intuitive meaning for smoothing</li> </ul>	Yes	No
(Gaussian) Kernel Smoothing	<ul style="list-style-type: none"> <li>- Close points are related to each other via a kernel funtion</li> </ul>	<ul style="list-style-type: none"> <li>- Simple</li> <li>- General assumptions</li> </ul>	<ul style="list-style-type: none"> <li>- Bandwidth: fails if there are big data-gaps</li> </ul>	Yes	Yes
Double-Logistic	<ul style="list-style-type: none"> <li>- Function first increases then decreases</li> <li>- Ndvi has a minimal value</li> </ul>	<ul style="list-style-type: none"> <li>- Good for evergreen plants (if snow masks ndvi)</li> <li>- Upper envelope</li> </ul>	<ul style="list-style-type: none"> <li>- Parameterestimation can go seriously wrong</li> <li>- Strange behaviour for long data-gaps</li> </ul>	Yes	(Yes)
Universal Kriging	<ul style="list-style-type: none"> <li>- Function is a realization of a stationary gaussian process</li> </ul>	<ul style="list-style-type: none"> <li>- Informative parameters</li> <li>- Flexible</li> </ul>	<ul style="list-style-type: none"> <li>- Regression to the mean</li> <li>- Assumptions clearly not met</li> </ul>	Yes	(Yes)

266 We will introduce some approaches to estimate  $m$  in section 3.3 and 3.4.

267 Furthermore, in the subsequent, we denote  $w \in \mathbb{R}^n$  as the vector of weights such that  $w_i$  corresponds to the weight that  $(x_i, Y_i)$  should have in the interpolation.

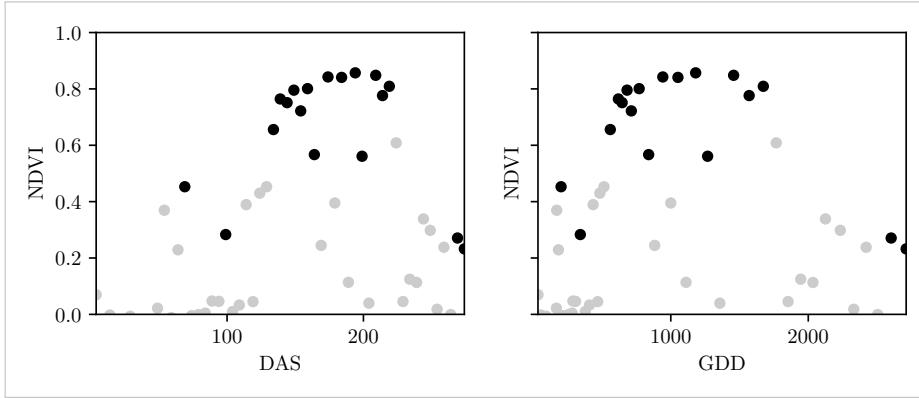


Figure 3.1: The same NDVI time-series, on the left with DAS as the timescale, on the right GDD is the timescale. SCL45 are colored black. Non-SCL45 (clouds and shadows) are colored in gray.

### 3.3 Parametric Regression

Parametric Curve estimation tries to fit a parametric function (e.g. a Gaussian function with parameter  $\mu$  and  $\sigma$ ) to a dataset. In the following, we introduce 2 such parametric approaches.

#### Optimization Issues

We shall mention some optimization issues we countered during implementation. Since we aim to minimize the residuals sum of squares over 5 (or 6) parameters, we try to solve a non-convex optimization problem. Thus, the algorithm<sup>1</sup> either struggles to find the global minimum or fails to converge. This was fixed by providing for each parameter reasonable initial values and generous bounds (which match our experience).

##### 3.3.1 Double Logistic

The Double Logistic smoothing as described in [Beck, Atzberger, Høgda, Johansen, and Skidmore \(Beck et al.\)](#) heavily relies on shape assumptions of the fitted curve (i.e. the NDVI time series).

Assumptions:

- There is a minimum NDVI level  $y_{\min}$  in the winter (e.g. due to evergreen plants), which might be masked by snow. This can be estimated beforehand, taking into several years into account.
- The growth cycle can be divided into an increase and a decrease period, where the time series follows a logistic function. The maximum increase (or decrease) is observed at  $t_0$  (or  $t_1$ ) with a slope of  $d_0$  (or  $d_1$ ).

The equation of the double-logistic fit is given by:

$$y(t) = y_{\min} + (y_{\max} - y_{\min}) \left( \frac{1}{1 + e^{-d_0(t-t_0)}} + \frac{1}{1 + e^{-d_1(t-t_1)}} - 1 \right)$$

Where the five free parameters:  $y_{\max}$ ,  $d_0$ ,  $d_1$ ,  $t_0$ ,  $t_1$  are initially estimated by least squares. Such fit can be seen in figure 3.2.

<sup>1</sup>We used the python function `scipy.optimize.curve_fit`

<sup>292</sup> Similar as for the Savitzky-Golay Filter (cf. section 3.4.3) we reestimate (only once) the  
<sup>293</sup> parameters by giving less weight to the overestimated observations and more weight to  
<sup>294</sup> the underestimated observations<sup>2</sup>.

Pros	Cons
<ul style="list-style-type: none"> <li>— Incorporates subject specific knowledge in the case of evergreen plants covered in snow.</li> <li>— Optimized parameters have an intuitive meaning.</li> </ul>	<ul style="list-style-type: none"> <li>— Strong shape assumptions on the NDVI curve.</li> <li>— Parameter optimization might go wrong. This can be mitigated to some extent to provide bounds for the parameters</li> <li>— Strange behavior in regions with little observations. (cf. figure 3.2)</li> </ul>

<sup>295</sup> **3.3.2 Fourier Approximation**

Similar as in section 3.3.1 we fit a parametric curve to the data by least squares. Here we take the second order Fourier series:

$$\text{NDVI}(t) = \sum_{j=0}^2 a_j \times \cos(j \times \Phi_t) + b_j \times \sin(j \times \Phi_t)$$

<sup>296</sup> where  $\Phi = 2\pi \times (t - 1)/n$ .

Pros	Cons
<ul style="list-style-type: none"> <li>— Assumption of periodicity can be helpful if we are modelling multiyear grow cycles</li> <li>— Flexible curve shape</li> </ul>	<ul style="list-style-type: none"> <li>— Bad behavior in regions with little data (cf. figure 3.2)</li> <li>— Hard to interpret estimated parameters</li> <li>— Parameter estimation can go wrong. Introducing bounds can help.</li> </ul>

<sup>297</sup> **3.4 Non-Parametric Regression**

<sup>299</sup> In non-parametric curve estimation, we no longer demand our curve to be fully determined  
<sup>300</sup> by several parameters, but we allow it to also dependent on the data. That said, we might  
<sup>301</sup> still use some tuning-parameters sometimes.

TODO:  
include  
Weighted  
versions

<sup>302</sup> **3.4.1 Kernel Regression**

<sup>303</sup> As described previously, we would like to estimate

$$\mathbb{E}[Y | X = x] = \int_{\mathbb{R}} y f_{Y|X}(y | x) dy = \frac{\int_{\mathbb{R}} y f_{X,Y}(x, y) dy}{f_X(x)}, \quad (3.4.1.1)$$

<sup>2</sup>For the details on the weights we refer to Beck, Atzberger, Høgda, Johansen, and Skidmore (Beck et al.)

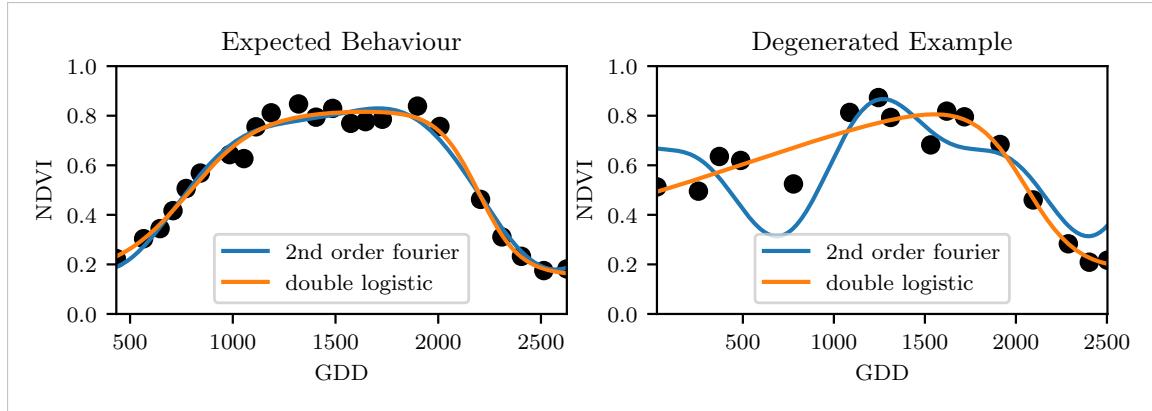


Figure 3.2: Here we observe the nice fitting possibilities of the two parametric methods but notice also some misbehavior

where  $f_{Y|X}, f_{X,Y}, f_X$  denote the conditional, joint and marginal densities. This can be done with a kernel  $K$ :

$$\hat{f}_X(x) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)}{nh}, \hat{f}_{X,Y}(x, y) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) K\left(\frac{y-Y_i}{h}\right)}{nh^2}$$

By plugging the above into equation (3.4.1.1) we arrive at the *Nadaraya-Watson* kernel estimator:

$$\hat{m}(x) = \frac{\sum_{i=1}^n K\left((x - x_i)/h\right) Y_i}{\sum_{i=1}^n K\left((x - x_i)/h\right)}$$

304 Common choices for the kernel are the normal function or a uniform function (also called  
 305 “box”function.). Note that we still need to choose the bandwidth of the function (in the  
 306 case of the normalfunction this is  $\sigma$  the standarddeviation). For local adaptive bandwith-  
 307 selection we refer to [Brockmann, Gasser, and Herrmann \(Brockmann et al.\)](#).

### Pros

- flexible due to different possible kernels
- can be assigned degrees of freedom (trace of the hat-matrix)
- estimation of the noise variance  $\hat{\sigma}_\varepsilon^2$  (REF cf. CompStat 3.2.2)

### Cons

- if the  $x \mapsto K(x)$  is not continuous,  $\hat{m}$  isn’t either
- choice of bandwidth, especially if  $x_i$  are not equidistant.

308 **3.4.2 Kriging**

309 Kriging was developed in geostatistics to deal with autocorrelation of the response variable  
 310 at nearby points. By applying the notion that two spectral indices which are (timewise)  
 311 close should also take similar values, we justify the application of Kriging. In the end, we  
 312 would like to fit a smooth Gaussian process to the data. For this subsection, we will follow  
 313 [Diggle and Ribeiro \(dig\)](#).

314 **Definitions and Assumptions**

315 **Definition 3.4.2.1. (Gaussian Process)** A Gaussian Process  $\{S(t) : t \in \mathbb{R}\}$  is a stochastic  
 316 process if  $(S(t_1), \dots, S(t_k))$  has a multivariate Gaussian distribution for every collection of

317 times  $t_1, \dots, t_k$ .  $S$  can be fully characterized by the mean  $\mu(t) := E[S(t)]$  and its covariance  
 318 function  $\gamma(t, t') = \text{Cov}(S(t), S(t'))$

319 **Assumption 1.** We will assume the Gaussian process to be stationary. That is for  $\mu(t)$   
 320 to be constant in  $t$  and  $\gamma(t, t')$  to depend only on  $h = t - t'$ . Thus, we will write in the  
 321 following only  $\gamma(h)$ .<sup>3</sup>

**Definition 3.4.2.2.** (*Variogram*) We also define the variogram of a Gaussian process as

$$V(h) := V(t, t+h) := \frac{1}{2} \text{Var}(S(t) - S(t+h)) = (\gamma(0))^2(1 - \text{corr}(S(t), S(t+h)))$$

And decide to use a Gaussian Variogram defined by

$$V(h) = p \cdot \left( 1 - e^{-\frac{h^2}{(\frac{4}{7}r)^2}} \right) + n,$$

322 where  $h$  is the distance,  $n$  is the nugget,  $r$  is the range and  $p$  is the partial sill visualized  
 323 in figure 3.3.<sup>4</sup>

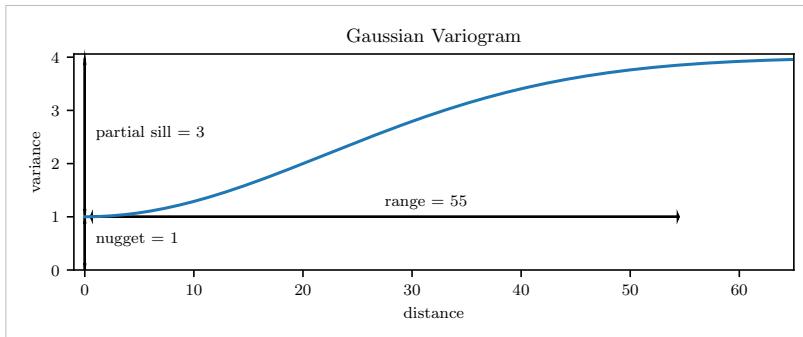


Figure 3.3: Gaussian Variogram with nugget=1, partial sill=3, range=55

324 Next, we consider a one-dimensional Gaussian process  $G_\gamma$  with variogram  $\gamma$ . We tune the  
 325 variogram parameters using maximum likelihood<sup>5</sup>. Let  $z$  be a vector with the new values  
 326 to extrapolate, then we can determine the values  $m(z) = \mathbb{E}[G_\gamma(z)|(x, y)]$  using bayes rule<sup>6</sup>.  
 327 For an example fit we refere to figure 3.4.

328 Since we obsere a clear pattern of a growth period in spring and harvest in the end of  
 329 summer, we have to admit that assumption 1 with the constant mean is clearly violated.  
 330 This is also the reason why we observe (for every variogram parameter) a tendency to the  
 331 mean as indicated in figure 3.4.

<sup>3</sup>Note that the process is also *isotropic* (i.e.  $\gamma(h) = \gamma(\|h\|)$ ) since we are in a one-dimensional setting and the covariance is symmetric.

<sup>4</sup>Strictly speaking we use a scaled version of the variogram. Thus, only the ratio of  $p/n$  matters.

<sup>5</sup>As illustrated in figure 3.4 maximum likelihood estimation can lead to overfitting. Thus, we will in practice sample several such optimized parameters and use their median in the end.

<sup>6</sup>Bayes rule generally claims, that for two random variables  $A$  and  $B$  we have that  $P(A|B) = P(B|A)/P(B)$

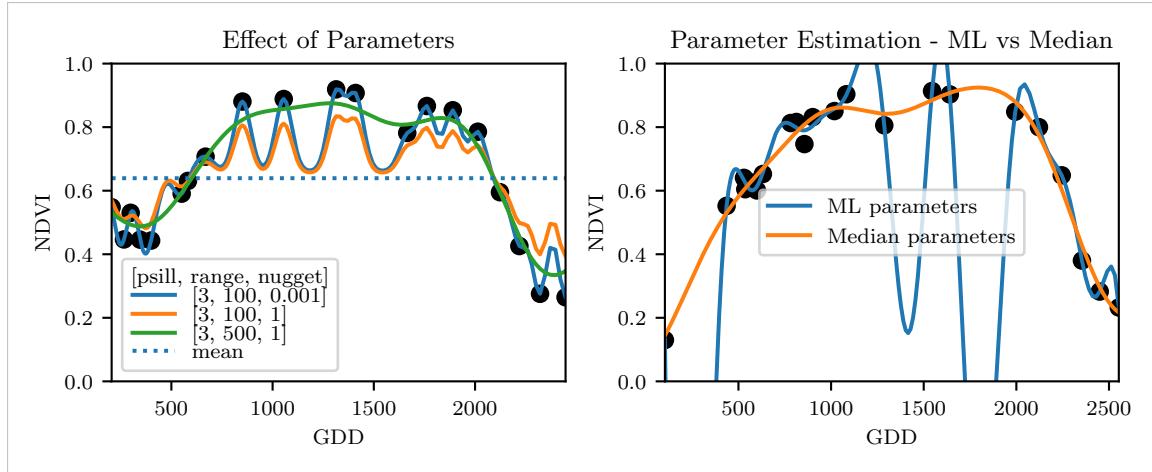


Figure 3.4: On the left, we see how the interpolation change if we increase the nugget and the range parameter. On the right we compare two kriging interpolations, where one takes parameters by numerically maximizing the (which results in a very small nugget) and the other takes the median of many such numerical optimizations.

Pros	Cons
<ul style="list-style-type: none"> <li>— It is a well-studied method.</li> <li>— Variogram parameters have an intuitive meaning.</li> <li>— Flexible covariance structure.</li> </ul>	<ul style="list-style-type: none"> <li>— Regression to the mean.</li> <li>— Violated assumption of constant mean and constant variance. Thus, the NDVI is not a stationary process.</li> <li>— Skewness of errors is not taken into account.</li> </ul>

### 332 3.4.3 Savitzky-Golay Filter (SG Filter)

The *Savitzky-Golay Filter*, introduced in [Savitzky and Golay](#) ([Savitzky and Golay](#)) is a technique in signal processing and can be used to filter out high frequencies (low-pass filter) as argued in [Schafer](#) ([Schafer](#)). Furthermore, it also can be used for smoothing by filtering high frequency noise while keeping the low frequency signal. First, we choose a window size  $m$ . Then, for each point,  $j \in \{m, m+1, \dots, n-m\}$  we fit a polynomial of degree  $k$  by:

$$\hat{y}_j = \min_{p \in P_k} \sum_{i=-m}^m (p(x_{j+i}) - y_{j+i})^2,$$

333 where  $P_k$  denotes the Polynomials of degree  $k$  over  $\mathbb{R}$ .

For equidistant points this can efficiently be calculated by

$$\hat{y}_j = \sum_{i=-m}^m c_i y_{j+i},$$

334 where the  $c_i$  are only dependent on the  $m$  and  $k$  and are tabulated in the original paper.

### 335 Adaptation to the NDVI

336 In the rather famous paper of [Chen, Jönsson, Tamura, Gu, Matsushita, and Eklundh](#) ([Chen et al.](#)) a “robust” method based on the Savitzky-Golay has been used. The method

338 is based on the assumption that due to atmospheric effects the observed NDVI tends to  
 339 be underestimated and that it cannot increase too quickly<sup>7</sup>. Their proposed algorithm is:

- 340 i.) Remove points which are labeled as cloudy.
- 341 ii.) Remove points which would indicate an increase greater than 0.4 within 20 days.
- 342 iii.) Linearly interpolate to obtain an equidistant time series  $X^0$ .
- 343 iv.) Apply the Savitzky-Golay Filter to obtain a new time series  $X^1$ .
- 344 v.) Update  $X^1$  by applying again a Savitzky-Golay Filter. Repeat this until  $w^T |X^1 - X^0|$   
 345 stops decreasing, where  $w$  is a weight vector with  $w_i = \min\left(1, 1 - \frac{X_i^1 - X_i^0}{\max_i \|X_i^1 - X_i^0\|}\right)$ .  
 346 This reduces the penalty introduced by outliers<sup>8</sup> and by repeating this step we approach the “upper NDVI envelope”.

Pros	Cons
<ul style="list-style-type: none"> <li>— Popular technique in signal processing.</li> <li>— Efficient calculation for equidistant points.</li> <li>— Upper envelope matches intuition for the NDVI. Therefore, it is robust against outliers with small values.</li> </ul>	<ul style="list-style-type: none"> <li>— No natural way of how to estimate points which are not in the data.</li> <li>— Not generalizable to other spectral indices.</li> <li>— Linear interpolation to account for missing data might be not appropriate.</li> <li>— No smooth interpolation between two measurements.</li> </ul>

348 **Extension: Spatial-Temporal-Savitzky-Golay Filter**

349 One notable adaptation of the Savitzky-Golay is the presented by Cao, Chen, Shen, Chen,  
 350 Zhou, Wang, and Yang (Cao et al.). The key difference is the additional assumption of the  
 351 cloud cover being discontinuous and that we can improve by looking at adjacent pixels<sup>9</sup>.  
 352 Because we are working with rather high resolution satellite data, and we need the variance  
 353 in the predictors, we will waive this extension.

354 **3.4.4 Locally Weighted Regression (LOESS)**

355 The Locally Weighted Regression (LOESS) introduced by Cleveland (Cleveland) can be  
 356 understood as a generalization of the Savitzky-Golay Filter (cf. sec. 3.4.3).

Given a proportion  $\alpha \in (0, 1]$ , we estimate each  $y_i$  separately by fitting a polynomial of order  $d$  by weighted least squares. The weights are (usually) defined by

$$w_i(x_j) = \begin{cases} \left(1 - \left(\frac{x_j}{h_i}\right)^3\right)^3, & \text{for } |x_j| < h_i, \\ 0, & \text{for } |x_j| \geq h_i \end{cases}$$

357 where  $h_i$  is the minimal distance such that  $\lceil \alpha n \rceil$  observations are in the ball  $B_{h_i}(x_i)$ .<sup>10</sup> So  
 358 for each  $y_i$  we only consider a proportion  $\alpha$  of the observations.

<sup>7</sup>The latter is argued by the biological impossibility of such fast vegetation changes

<sup>8</sup>Here we call a point  $i$  an outlier if  $X_i^0 < X_i^1$ .

<sup>9</sup>Here, we say that a pixel is adjacent if it is the same pixel but from a different year (keeping the same day of the year) or (if not enough of such temporal-adjacent pixel are found) it is spatially adjacent

<sup>10</sup>If too many weights are set to zero, we might end up considering not enough observations and thus

359 **How does the Robust LOESS differ from the SG Filter?**

360 The LOESS smoother takes a fraction of points instead of a fixed number and therefore  
 361 automatically adapts to the size of the data we wish to interpolate. However, we run  
 362 into the danger of considering too little observations, since the estimation breaks down if  
 363  $\lceil \alpha n \rceil < d + 1$ .<sup>10</sup> Furthermore, LOESS gives less weight to points further away. This yields  
 364 a "smoother" estimate, since when we slide the window (e.g. for estimating the next value)  
 365 an influential point at the border does not suddenly get zero weight from being weighted  
 366 equally before. Finally, the LOESS also can be used for non-equidistant data and allows  
 367 for arbitrary interpolation.

Pros	Cons
— Flexible generalization of Savitzky-Golay	— The nature of local regression might lead to surprising estimates (no smoothness guarantees for the second derivative)
— arbitrary interpolation possible	
— Intuitive parameters	

368 **3.4.5 B-splines**

B-splines as discussed in [Lyche and Mørken](#) ([Lyche and Mørken](#)) are piecewise cubic polynomials defined by

$$S(x) = \sum_{j=0}^{n-1} c_j B_{j,k;t}(x),$$

where  $B$  are basisfunctions and recursively defined by:

$$B_{i,0}(z) = 1, \text{ if } t_i \leq z < t_{i+1}, \text{ otherwise } 0 \\ B_{i,k}(z) = \frac{z-x_i}{x_{i+k}-x_i} B_{i,k-1}(z) + \frac{x_{i+k+1}-z}{x_{i+k+1}-x_{i+1}} B_{i+1,k-1}(z).$$

Assuming that all  $x_i$  are distinct this yields a interpolation which fits the data perfectly. To reduce the amount of overfitting and increase the smoothness we relax the constraint that we have to perfectly interpolate. Thus, we use the minimum number of basisfunction<sup>11</sup> such that:

$$\sum_{i=1}^n (w_i(y_i - \hat{y}_i))^2 \leq s$$

Pros	Cons
— can be assigned degrees of freedom	— smoothing process does not translate well to a interpretation (unlike smoothing splines)
— extendable to "smooth" version	
— performs also well if points are not equidistant	— choice of smoothing parameter $s$

get a singular design-matrix (for the least squares estimation). Therefore, we substitute  $h_i$  with  $1.01h_i$ , so that the observation on the boundary of  $B_{h_i}(x_i)$  does not get completely ignored. But we also have to assure that  $\alpha$  is big enough.

<sup>11</sup>So we do not require one basisfunction for each neighboring pair of notes. SciPy uses FITPACK and DFITPACK, the documentation suggests that smoothness is achieved by reducing the number knots used

369 **3.4.6 Natural Smoothing Splines**

370 Let  $\mathcal{F}$  be the Sobolev space (the space of functions of which the second derivative is  
 371 integrable) . Then the unique<sup>12</sup> minimizer

$$\hat{m} := \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n w_i (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

372 is a natural<sup>13</sup> cubic spline (i.e. a piecewise cubic polynomial function). The objective  
 373 function has an intuitive meaning, as to avoid lateral acceleration it is desirable to move  
 374 the steering wheel as little as possible, when driving a car.

Pros	Cons
<ul style="list-style-type: none"> <li>— Can be assigned degrees of freedom (trace of the hat-matrix).</li> <li>— Efficient estimation (closed form solution).</li> <li>— Intuitive penalty (we don't want the function to be too "wobbly" — change slopes).</li> <li>— Performs also well if points are not equidistant.</li> <li>— Fixes the Runge's phenomenon (fluctuation of high degree polynomial interpolation).</li> </ul>	<ul style="list-style-type: none"> <li>— Choose <math>\lambda</math>.</li> </ul>

375 **3.5 Tuning parameter estimation**

376 Many of the interpolation methods introduced in section 3.3 and 3.4 include a free parameter.  
 377 To determine this parameter for a specific interpolation method, we will estimate the  
 378 absolute residuals using OOB estimation and then optimize the parameter using statistics.  
 379 We clarify the procedure step by step:

- 380 i.) Construct a set  $\Lambda$  of candidate parameters that generously covers the parameter  
 381 space.
- 382 ii.) Consider  $\mathcal{P}$ , a set of Pixels.
- 383 iii.) For each parameter  $\lambda \in \Lambda$  consider the individual pixels and compute the LOOCV<sup>14</sup>  
 384 for the absolute residuals of the specific NDVI-interpolation method for all Pixels in  
 385  $\mathcal{P}$  and store them in the set  $R_\lambda$ .
- 386 iv.) Determine  $\lambda_{optimal} = \arg \min_{\lambda \in \Lambda} \text{quantile}(90)(R_\lambda)$ , where we describe the 90% quantile with  $\text{quantile}(90)$ .

388 We choose  $\text{quantile}(90)$  as our optimization function because we want to allow 10% of  
 389 outliers (corrupt points) but also aim for an accurate fit in 90% of the cases.

<sup>12</sup>Strictly speaking it is only unique for  $\lambda > 0$

<sup>13</sup>It is called natural since it is affine outside the data range ( $\forall x \notin [x_1, x_n] : \hat{m}''(x) = 0$ )

<sup>14</sup>For a definition of the leave-one-out-cross-validation we refer to section 2.2.3

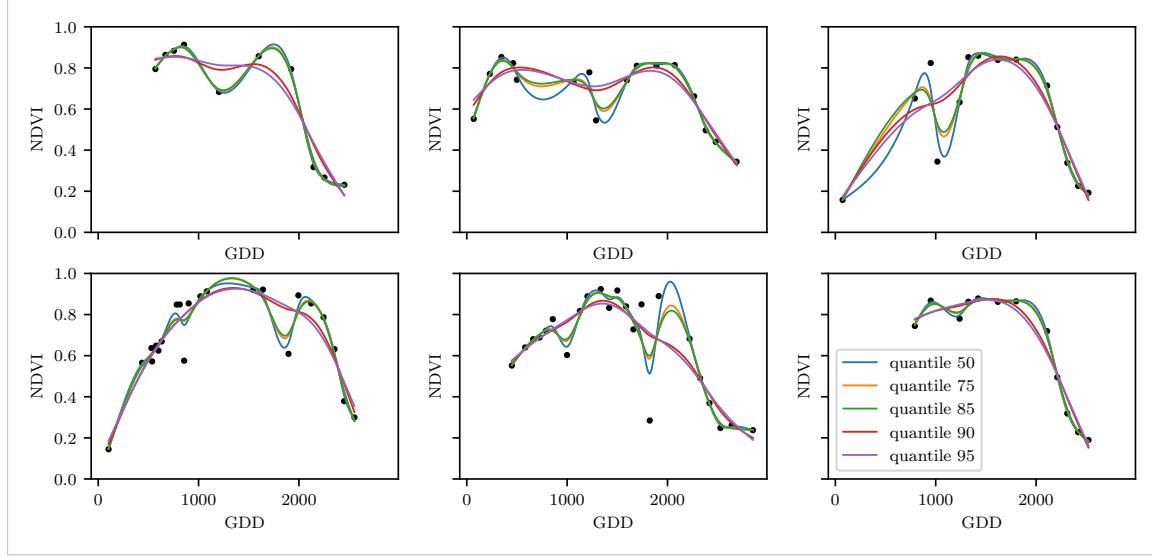


Figure 3.5: Smoothing splines fit with smoothing parameter optimized by minimizing the “...”-quantile of the absolute leave-one-out residuals. Note that the larger the considered quantile is, the smoother the resulting curve becomes.

390 The figure 3.5 exemplifies the effect of the optimization function (different quantiles). To  
391 summarize, we may say that the higher the quantile, the stronger the smoothing.

## 392 3.6 Robustify

393 Now we discuss a general approach of how to make an interpolation more robust against  
394 outliers. The main idea is to give less weight to observations that have high residuals after  
395 the initial (or if we reiterate, the last) fit.

396 Even though the procedure is taken from the robust version of the LOESS smoother (cf.  
397 section 3.4.4 and Cleveland (Cleveland)), we can apply it to every interpolation method  
398 that allows for prior weighting of observations.

Before we describe the procedure, we define a function that will determine the weight given to each observation, such that observations with large-scaled residuals will have less weight. That is the bisquare function  $B$ :

$$B(x) := \begin{cases} (1 - x^2)^2, & \text{if } |x| < 1 \\ 0, & \text{else} \end{cases}$$

399 Now, we do something similar to what is done in iteratively reweighted least squares. After  
400 an initial interpolation, update the weights of each observation with

$$w_i^{\text{new}} := w_i^{\text{old}} B\left(\frac{|r_i|}{6 \text{med}(|r_1|, \dots, |r_n|)}\right); \quad r_i := y_i - \hat{y}_i \quad (3.6.0.1)$$

401 and interpolate again using the new weights. We can iterate this reweighting and stop  
402 after several steps or when the change of the values is smaller than some tolerance.

403 Note that this procedure is indeed robust since we use the median for the normalization  
 404 which has a breakdown point of 50%.<sup>15</sup>

405 **3.6.1 Our Adjustment:**

In the case that we would like to apply prior weights, we want to prevent low-weighted observations to corrupt our estimation of scale (the median) and thus we use the weighted median. This can be defined as

$$\text{med}_{\text{weighted}}(r, w) := \arg \min_{\lambda \in \mathbb{R}} \sum_{i=1}^n |r_i w_i - \lambda|$$

406 for  $r, w \in \mathbb{R}^n$ .<sup>16</sup>

407 **3.6.2 Examples and Conclusions**

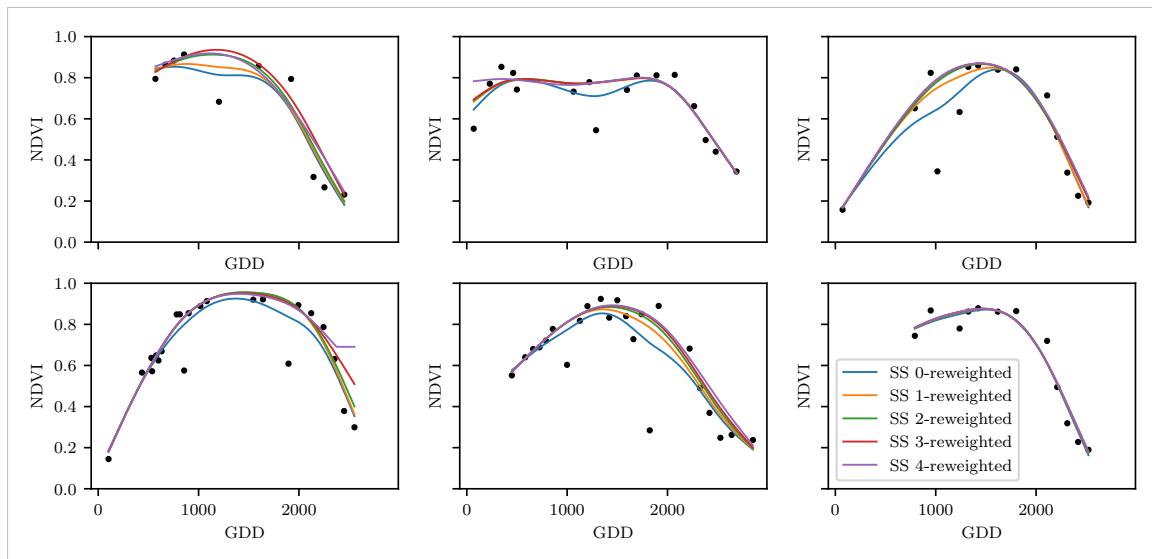


Figure 3.6: Smoothing Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.6) are also displayed

408 In figure 3.6 we observe for 6 pixels how the NDVI time series interpolated with smoothing  
 409 splines looks after 0, 1, 2, 3, 4 iterations ( we refer to the appendix for the analogous figures  
 410 of the other interpolation methods): B.1, B.2, B.3 and B.1).

411 Indeed, we observe how the interpolated time series is less affected by outliers after each  
 412 iteration. The biggest difference we notice in the first iteration. Furthermore, in the plot  
 413 at the bottom left we see how the interpolation “escapes” from the right endpoint with  
 414 each successive iteration, even though our intuition does not necessarily identify this point  
 415 as an outlier. Therefore, in the following, we will always perform only one iteration and  
 416 then stop.

<sup>15</sup>The breakdown point relates only to outliers in the  $y$  values. Note that we do not require the interpolation methods to be robust, since the residual for an outlier will still be larger than for non-outliers and thus will be down weighted more and more in each iteration (because for the next iteration the residual of the outlier will be even larger, since we gave less weight to it).

<sup>16</sup>This adjustment is also necessary to keep the scale estimation meaningful during the iterations.

Table 3.2: Performance comparison of different interpolation methods measured with various statistics. Considering only SCL45 points, we get the out-of-bag estimates using the given interpolation method. Consequently, we compute the absolute (value of the) residuals and apply the given statistic to it.

	ss	loess	dl	bspl	fourier	ss rob	loess rob	dl rob	bspl rob	fourier rob
rmse	0.063	0.061	0.061	0.074	0.075	0.070	0.065	0.065	0.079	0.208
qtile50	0.036	0.034	0.027	0.043	0.031	0.032	0.031	0.022	0.037	0.049
qtile75	0.063	0.061	0.051	0.077	0.058	0.061	0.057	0.044	0.070	0.099
qtile85	0.080	0.079	0.070	0.098	0.083	0.081	0.076	0.063	0.094	0.158
qtile90	0.092	0.092	0.088	0.112	0.108	0.097	0.090	0.082	0.113	0.226
qtile95	0.119	0.115	0.122	0.142	0.161	0.132	0.115	0.124	0.157	0.375

### 417 3.6.3 Upper Envelope Approach - Penalty for Negative Residuals

418 If we artificially increase the negative residuals in 3.6.0.1 by multiplying (e.g. factor 2),  
 419 the corresponding points will get less weight in the next iteration. This allows us to create  
 420 an interpolation that resembles an upper envelope. Intuitively, this upper envelope can be  
 421 thought of as a sheet that is laid on top of the points.

422 This approach is based on the premise that we tend to underestimate the NDVI (as in  
 423 REF-savitzky-golay). Since we want to develop a general method that is in principle not  
 424 related to the NDVI, we will not pursue this approach further.

## 425 3.7 Performance Assessment

426 Next, we will benchmark the different interpolation methods with and without robustification.  
 427 For this, we will use the same technique as we did for the parameter determination  
 428 in section 3.5. On  $B_\lambda$  we apply the RMSE and different quantiles and present the results  
 429 in table 3.2.

## 430 3.8 XXX Evaluation

- 432 – ss dominate (i.e. have better benchmark values w.r.t. all considered statistics) b-splines
- 433 (robustified and non-robustified)
- 434 – dl dominate Fourier (robustified and non-robustified)
- 435 – loess slightly dominates ss, but we prefer ss because of the smoothness guarantees (com-  
436 pare the figures B.1 and 3.6).
- 437 – use dl and ss in the following (keeping robustified and non-robustified variants)

write  
out key-  
words,  
after  
final re-  
sults

438 **Chapter 4**

439 **NDVI Correction**

440 Let's remind ourselves that the data from the Sentinel-2 is equipped with a scene classi-  
441 fication layer (*SCL*) and we therefore have some information about what is observed at  
442 each pixel for each sampled time (cf. table 2.2). So far, we have only considered cloud-free  
443 points (i.e., SCL-classes 4 and 5). In this chapter, we would like to improve the NDVI  
444 interpolation by inspecting also other SCL-classes and by using more information than  
445 just the two bands used to calculate the NDVI (B4 and B8).

446 **4.1 Considering other SCL Classes**

447 In figure 4.1 we notice that some blue points<sup>1</sup> follow the interpolated line closely and that  
448 they might be useful in improving an interpolation fit.

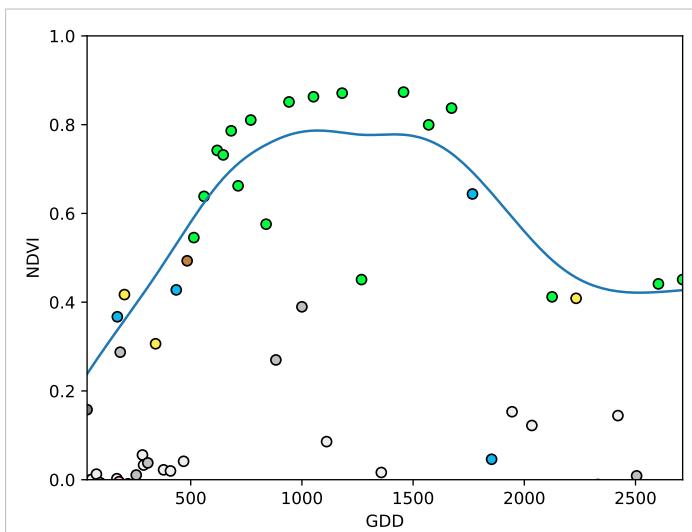


Figure 4.1: A smoothing splines fit considering green and yellow points (SCL45)

449 To get an impression of whether there is some useful information contained in the remaining  
450 SCL-classes (all except 4 and 5) we would like to compare the observed NDVI with the  
451 true NDVI. But since we do not have any ground truth data, we will make the following  
452 assumption:

---

<sup>1</sup>The blue points correspond to the SCL-class 10: Thin cirrus clouds

453 **Assumption 1.** The true NDVI value at time  $t$  can be successfully estimated by out-of-bag  
 454 interpolation using high-quality observations. That is the interpolated value (using an  
 455 interpolation method from chapter 3) considering the points  $P^{SCL45} \setminus P_t$ . In the following,  
 456 we will call this estimate the “true”-NDVI.

457 We would like to get an idea if there is any hope to recover information from SCL-classes  
 458 other than 4 and 5. For that, we will check for the other SCL-classes if there is a relation  
 459 between the “true”-NDVI<sup>2</sup> and the observed NDVI. Thus, we pair each “true”-NDVI with  
 460 its observed one, collect all pairs, and create a scatter plot for each SCL-class in fig 4.2.  
 461 As expected the “true” and the observed NDVI seem to be highly correlated for SCL45.  
 462 But we can also detect some patterns of correlation in the SCL-classes 2, 3, 7, 8 and 10.

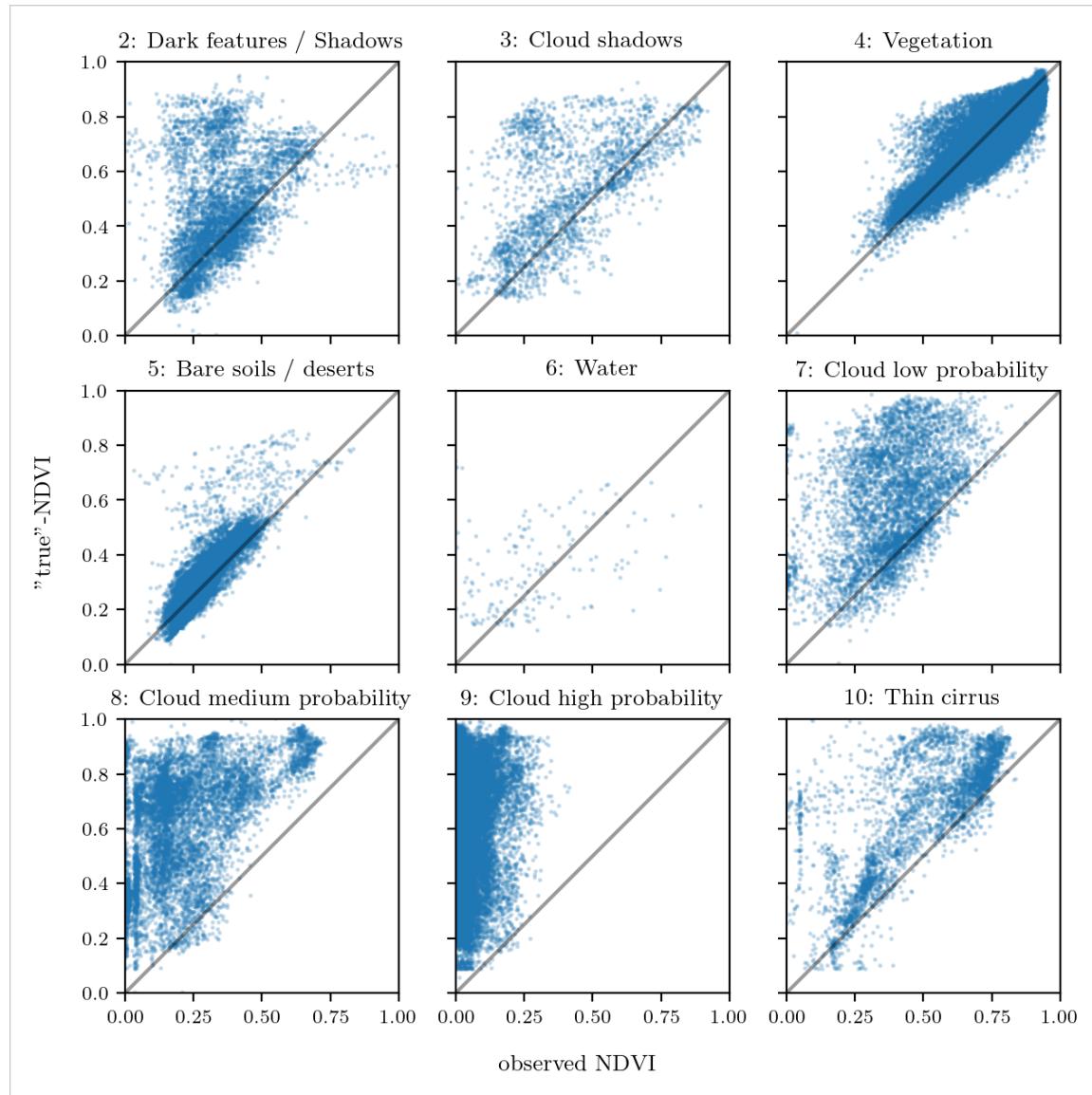


Figure 4.2: For each SCL class, we compare the true NDVI with the observed NDVI. (The true NDVI was estimated with OOB smoothing splines, and we used all observations of 10% of the total training pixels.)

<sup>2</sup> i.e. the out-of-bag (OOB) estimate using smoothing splines

463 It might be tempting to include some of the above SCL classes (for interpolation). But  
 464 on the one hand, the choice would not be objective and on the other hand, the correlation  
 465 seems to be weaker than for SCL45. Therefore, in the following section, we shall try to  
 466 correct the observed NDVI and estimate the uncertainty of each correction.

## 467 4.2 Correction

468 We recall the satellite images in figure 2.1d, where we had cloudy images despite SCL4  
 469 labeled and see fragments in figure 2.1e even though we are supposed to see clouds (SCL  
 470 10 - Cirrus clouds). The SCL classification is based only on a mixed model trained using  
 471 the s2 bands.

472 We will improve our NDVI interpolation by not relying on the existing SCL classifica-  
 473 tion, but by training our own model to estimate/correct NDVI using all S2 bands (see  
 474 sections 4.2.1 and 4.2.2). After we have corrected the observed NDVI, we will find out  
 475 how uncertain our corrections are and translate these uncertainties into weights (in sec-  
 476 tion 4.2.3). These we will use for the subsequent interpolation. This step-by-step procedure  
 477 is illustrated by the REF graph in the appendix.

478 Finally, in section 4.4 we will evaluate this correction procedure, considering different  
 479 interpolation methods and correction models.

### 480 4.2.1 Response and Covariates

481 For training an NDVI correction model, we need ground-truth (response) and informative  
 482 covariates. We organize those in a table, where each row corresponds to a  $P_t$  (i.e., a  
 483 pixel at a time  $t$ ). For the response, we will again use the assumption 1. There is no  
 484 canonical answer to the question of which covariates we should use. It is a tradeoff between  
 485 simplicity/generalizability and performance (with the danger of overfitting). Our desire  
 486 with the NDVI correction is to develop a product that is simple for others to understand  
 487 and use. Therefore, in the subsequent, we will only take the spectral data of the satellite  
 488 and the observed NDVI derived from it as covariates<sup>3</sup>.

### 489 4.2.2 Correction Methods

490 In the following, we will introduce different modelling approaches, which we will use to  
 491 model the relation between the response  $y = y_{\text{true OOB NDVI}} \in \mathbb{R}^n$  and the covariates  
 492 encoded in the design matrix<sup>4</sup>  $X \in \mathbb{R}^{n \times p}$ . Furthermore, we remind ourselves of the  
 493 MATLAB notation discussed in section 2.2.1

494 XXX Note that in order to reduce computation time, only 10% of the training data has  
 495 been used to fit the subsequent models.

#### 496 Ordinary Least Squares (OLS)

497 The OLS is a linear model which aims to minimize the sum of the squared residuals. Let  
 498  $y \in \mathbb{R}^n$  be the vector of responses and  $X \in \mathbb{R}^{n \times p}$  be the design matrix, where each row  
 499 corresponds to one pixel and each column consist of one covariate<sup>5</sup>. We assume a linear

<sup>3</sup>We do not mention the intercept explicitly, but it will also be included.

<sup>4</sup>This is the Matrix which contains all covariates.

<sup>5</sup>Strictly speaking since SCL-classes are dummy variables

500 relationship between  $y$  and  $X$  and allow for gaussian noise. That is:

$$y = X\beta + \epsilon \quad \text{where } \epsilon \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

501 Assuming that  $X$  is regular, we can estimate the regression coefficients  $\beta$  by

$$\hat{\beta} = (X^T X)^{-1} X^T y = \arg \min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$

502 We will train two models, one using only the SCL-classes as covariates and the other one  
503 using all covariates (which are discussed in section 4.2.1).

Pros	Cons
— Simple method with good interpretability of coefficients.	— Catches only linear relationships. — No integrated variable selection. <sup>6</sup>
— Computationally cheap.	

#### 504 LASSO

505 The Lasso can be similarly expressed than the OLS but adds a penalty to the minimization  
506 problem:

$$\hat{\beta}_\lambda = \arg \min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 = \arg \min_{\beta \in \mathbb{R}^p \text{ and } \|\beta\|_1 < \lambda} \|y - X\beta\|_2^2. \quad (4.2.2.1)$$

507 Even though we do not have a closed form solution for equation (4.2.2.1) we can solve  
508 it easily via optimization, since the function  $\beta \in \{\beta \in \mathbb{R}^p \mid \|\beta\|_1 < \lambda\} \mapsto \|y - X\beta\|_2^2$  is  
509 continuous and convex.

510 Tibshirani (Tibshirani) shows that the LASSO solution tends to be sparse (for not too big  
511  $\lambda$ ). That is  $\beta_i = 0$  for most  $i = 1, \dots, p$

512 In order to know which  $\lambda$  to choose we try a huge range of possible values. For each  $\beta_\lambda$  we  
513 calculate the cross-validated  $RMSE_\lambda$ <sup>8</sup> (and its standard deviation  $\sigma_\lambda$  using the  $k$  folds)  
514 and define the  $\lambda$  with the smallest corresponding  $RMSE_\lambda$  as  $\lambda_{min}$ . From here we choose  
515 the largest  $\lambda$  for which the  $RMSE_\lambda$  is smaller than  $RMSE_{\lambda_{min}} + \sigma_\lambda$ . This yields a simpler  
516 model while keeping the  $RMSE$  reasonable model.

517 We will apply the Lasso using the selected covariates in section 4.2.1 and their second  
518 degree of interactions.<sup>9</sup>

Pros	Cons
— Usually yields a sparse solution. This tends to give better generalizability (prediction performance on unseen data).	— Estimate is biased. — Computationally expensive.
— Successfully deals with correlation in covariates.	
— Interpretable results.	

<sup>7</sup>The last two terms are equivalent by lagrangian optimization

<sup>8</sup>The cross-validate Root Mean Square Error is the mean of the RMSE's obtained for each fold (using the model trained on the remaining folds). We use the following definition of the  $RMSE$ :

$\sqrt{\sum_{i=1}^n (y - \hat{y})^2 / n}$

<sup>9</sup>This is if our covariates are  $\{a, b\}$ , then we will now use  $\{a, b, ab, a^2, b^2\}$ .

519 **Random Forest (*RF*)**

520 To define a random Forest introduced by Breiman (Breiman) we will first define what a  
 521 Tree is. A (*decision*) Tree is a graph  $(V, E)$  without circles, a distinct root node, every  
 522 node has at most two children and every leaf has a value assigned to it. At each node there  
 523 is a boolean condition testing if one variable is greater than some value and a pointer to  
 524 one child depending on the boolean value. To evaluate a tree we start at the root node,  
 525 test the boolean expression and go to the node indicated by the resulting pointer. This  
 526 we repeat until we end up at a leaf-node where we return the value assigned to it.

527 To build such a Tree we will recursively partition the covariate space using greedy splits<sup>10</sup>  
 528 decreasing the RMSE<sup>11</sup> each time. If the set we want to split contains less then a certain  
 529 amount of training points we stop.

530 To build a *Random Forest* we will bootstrap-aggregate<sup>12</sup> many such Trees<sup>13</sup>. The prediction  
 531 of the Random Forest for a new point  $x$  is then the mean of the predictions from all  
 532 the Trees.

Pros	Cons
— Captures non-linear relationships.	— Resulting (prediction) function is non-continuous but locally constant.
— Captures all interactions and performs automatic variable selection.	— Computationally expensive.
— Can deal with missing data.	— No interpretability.

533 **Multivariate Adaptive Regression Splines (*MARS*)**

534 A MARS model as introduced in Friedman (Friedman) can be described by

$$g(x) = \sum_{m=0}^M \beta_m h_m(x),$$

535 where the  $h_m$  are simple functions (explained later) and the  $\beta_m$  are estimated via least  
 536 squares.

537 In the building procedure of a MARS model we first select many of those simple functions  
 538 and later drop some of them to avoid overfitting. For the construction of those simple  
 539 functions define  $\mathcal{B}$  be the set of pairs of ‘hockystick functions’

$$\mathcal{B} := \left\{ (b_1, b_2) \mid (b_1(x), b_2(x)) = \left( (x_j - d)_+, (d - x_j)_+ \right), d = X_{1,j}, \dots, X_{n,j}, j = 1, \dots, p \right\}$$

540 and the set  $\mathcal{M} = \{1\}$  of all functions currently in the model. Now, consider  $\mathcal{C}$  the set of  
 541 candidate functions-pairs

$$\mathcal{C} := \{(h(\cdot)b_1(\cdot), h(\cdot)b_2(\cdot)) \mid h \in \mathcal{M}, (b_1, b_2) \in \mathcal{B}\} \quad (4.2.2.2)$$

<sup>10</sup>For computational reasons we will only use splits along one covariate. So we ‘cut’ our covariate space into rectangles.

<sup>11</sup>To calculate the RMSE we need a prediction. Let  $P$  be the current partition, then the predicted value for some  $x \in A \in P$  is the mean of the responses of all the points in  $A$  (included in the training data).

<sup>12</sup>That is we will sample (with replacement) several times  $n$  observations from our original data and fit a Tree to each such sample.

<sup>13</sup>Building the Tree, this time we will not test every covariate at each node (for the RMSE minimization) but a node-specific subsample of the covariates. Thus, also the the “second best split” can be selected.

542 and select the pair (which when added to  $\mathcal{M}$  and the coefficients refitted) reduces the  
 543 RMSE the most. Add the selected pair to  $\mathcal{M}$  and repeat until the RMSE reduction  
 544 becomes insignificant.

545 Finally, to avoid overfitting we prune the set  $\mathcal{M}$  by optimizing a generalized cross validation  
 546 score (GCV).<sup>14</sup>

547 To reduce computational complexity, we follow the recommendation from REF? and re-  
 548 strict  $h$  in equation (4.2.2.2) to be of degree one (so it is also in a pair of  $\mathcal{B}$ ). Consequently,  
 549  $\mathcal{C}$  contains functions with a degree of at most 2.

Pros	Cons
<ul style="list-style-type: none"> <li>— Catches non-linear relationships.</li> <li>— Interpretability via functions in <math>\mathcal{M}</math> and their coefficients.</li> <li>— Allows for interactions with variable selection.</li> </ul>	<ul style="list-style-type: none"> <li>— Computationally expensive (can be reduced by restricting the degree of interactions).</li> </ul>

### 550 General Additive Model (*GAM*)

551 GAMs as described in [Hastie and Tibshirani](#) ([Hastie and Tibshirani](#)) are a special case of  
 552 Projection Pursuit Regression, where only the  $p$  directions parallel to the coordinate axes  
 553 are considered. The result is different to a linear model since the coordinate functions are  
 554 not restricted to be linear but are assumed to be non-parametric functions. The model  
 555 can be written as:

$$g_{add}(x) = \mu + \sum_{i=1}^p g_j(x_j).^{15}$$

556 To estimate the non-parametric functions we can use smoothing splines (ref sec. 3.4.6).  
 557 For this let  $\mathcal{S}_j$  be the function which takes some  $z \in \mathbb{R}^n$  and returns the smoothing splines  
 558 fitted to  $(X_{:,j}, z)$  where the smoothing parameter is optimized by GCV. Since we cannot fit  
 559 all  $g_j$  simultaneously we will use a strategy named backfitting. We basically cycle through  
 560 the indicies  $1, \dots, p$  and refit  $\hat{g}_j$  each time. The following illustrates the procedure:

- 1)  $\hat{g}_1 = \mathcal{S}_1(y - \mu)$
  - 2)  $\hat{g}_j = \mathcal{S}_j(y - \mu - \hat{g}_1(X_{:,1}) - \dots - \hat{g}_{j-1}(X_{:,j-1}))$  for  $j = 2, \dots, p$
  - 3)  $\hat{g}_1 = \mathcal{S}_1(y - \mu - \hat{g}_2(X_{:,2}) - \dots - \hat{g}_p(X_{:,p}))$
  - 4)  $\hat{g}_j = \mathcal{S}_j(y - \mu - \sum_{k \neq j} \hat{g}_k(X_{:,k}))$  for  $j = 2, \dots, p$
- $\vdots$

561 We repeat step 3) and 4) until the change falls below some tolerance.

Pros	Cons
<ul style="list-style-type: none"> <li>— Captures non-linearity.</li> <li>— Good interpretability.</li> </ul>	<ul style="list-style-type: none"> <li>— No automatic variable selection.</li> <li>— Computationally expensive.</li> </ul>

<sup>14</sup>This means that we perform an iterative procedure to reduce the number of functions in  $\mathcal{M}$ . For every function  $h$  in  $\mathcal{M}$  we compute the model using  $\mathcal{M}$

$\{h\}$ . We discard the function which – when excluding from  $\mathcal{M}$  – leads to the best GCV score.

<sup>15</sup>where  $g_j$  is a real-valued function. For identifiability we also demand  $\mathbb{E}[g_j(X_{:,j})] = 0$  for  $j = 1, \dots, p$ .

562 **4.2.3 Uncertainty Estimation**

563 Once we correct the NDVI using the previous section, we are left with the problem that  
 564 not every correction is equally reliable.<sup>16</sup> Hence, we are interested in a measure of how  
 565 uncertain an estimate is.

566 We do this by replacing the response with the absolute residuals  $v := |y - \hat{y}|$  and modeling  
 567 their relationship with the covariates defined by  $X$ . In this way, we obtain a model for  
 568 the absolute residuals  $v$  and the estimator  $\hat{v}$ .

569 **4.2.4 Interpolation**

570 Consider now a pixel  $P$ ,  $\hat{y}^{(P)}$  its corrected NDVI and  $\hat{v}^{(P)}$  the estimated uncertainties of  
 571  $\hat{y}^{(P)}$ . In order to interpolate  $\hat{y}^{(P)}$ , we will give less weight to unreliable observations. Thus,  
 572 we define the weight function:

$$w_\tau^{(P)} := \frac{1}{R} \frac{1}{\hat{v}_\tau^{(P)}}, \quad \text{for } \tau = 1, \dots, n_P$$

573 where  $\tau$  is an index over the satellite images and  $R := \frac{\sum_i^{n_P} \hat{v}_i^{(P)}}{n_P}$  a normalization constant.  
 574 The normalization is needed since for some interpolation methods inflating the sum of  
 575 weights would decrease the effect of the smoothing.

576 **4.3 Resulting Interpolation Strategies**

577 We have developed the following procedure to obtain a new interpolation (keyword-wise):

- 578 i.) OOB Interpolation (+ robustify?)
- 579 ii.) Correction
- 580 iii.) Uncertainty estimation
- 581 iv.) Interpolation (+ robustify?)

582 At each step we have a choice, more precisely:

- 583 — Interpolation: Smoothing Splines / Double Logistic
- 584 — Robustify: Yes / No
- 585 — Correction & uncertainty estimation: RF / OLS – considering only SCL-classes /  
 586 OLS – considering all selected covariates / MARS / GAM / LASSO / no correction.

587 As it is not feasible to try every possible combination, we make the following restrictions  
 588 on which combinations we will consider:

- 589 — We use the same interpolation method each time.
- 590 — Either we robustify both times or we do not robustify at all.
- 591 — We use the same underlying method for correction and uncertainty estimation.

592 In this fashion, we obtain 28 distinct interpolation strategies, which we will benchmark in  
 593 the next section.

---

<sup>16</sup>One correction is illustrated in the figure B.4f. In this figure, the outer points (labeled as clouds) have a large scatter.

594 **4.4 Evaluation Method**

595 In this section, we introduce the relative yield-estimation-accuracy (*RYEA*) and utilize it  
 596 to evaluate the interpolation strategies from section 4.3.

597 **Definition 4.4.0.1.** (*RYEA*) Let  $y \in \mathbb{R}^n$  be the yield,  $M$  be a model for estimating  $y$ , and  
 598  $\hat{y} = M(X)$  where  $X$  describes the data<sup>17</sup>. We define the *RYEA* as the relative RMSE in  
 599 yield estimation. Formally expressed:

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

600

601 **4.4.1 Idea**

602 The fundamental assumption is that the closer the interpolated NDVI time series is to  
 603 the true one, the better it can be used to determine crop yield. Implicitly, we believe that  
 604 an NDVI time series which better models yield will incorporate more true information  
 605 about the underlying vegetation. Therefore, we want to determine a comparable RYEA  
 606 for each interpolation strategy and choose it as a benchmark criterion. This is an objective  
 607 measure, since we have not considered crop yield in any of our previous steps. Moreover,  
 608 this criterion is justified by the fact that yield estimation has been a motivation for the  
 609 interpolation.

610 **4.4.2 Yield Estimation**

611 For all the pixels, we will interpolate the NDVI time series with every interpolation strat-  
 612 egy. From the interpolated NDVI time series, we would like to estimate the yield. However,  
 613 given the high dimensionality and different lengths of the interpolation (not every time  
 614 series has the same start and end point), we must first map each NDVI time series into a  
 615 low-dimensional vector space. For this we will use the following statistics:

- 616 — Maximum slope
- 617 — Minimum slope
- 618 — Integral<sup>18</sup> over all
- 619 — Peak (i.e. maximal NDVI)
- 620 — Peak GDD (i.e. value at which the peak is attained)
- 621 — Integral<sup>18</sup> up to the peak
- 622 — Integral<sup>18</sup> after peak
- 623 — Integral<sup>18</sup> from 0-685 GDD
- 624 — Integral<sup>18</sup> from 685-1075 GDD

625 For the choice we were inspired by REF-kamir. However, we deliberately omit any statistic  
 626 that involves the minimum (e.g. the NDVI-range), since we regard the minimum as very  
 627 error-prone (clouds) and uninformative measure.

---

<sup>17</sup>We will use the matrixes derived in section 4.4.2

<sup>18</sup>We will only consider the integral of the function  $\max(0, NDVI - 0.3)$ , where 0.3 is assumed to be a minimal NDVI value. REF

628 As a result, we obtain for each interpolation strategy a matrix in which each row corre-  
629 sponds to a pixel and contains both the yield and the characterizing statistics. Using this  
630 matrix, we train a random forest<sup>19</sup> for yield estimation, and compute the integrated OOB  
631 estimates<sup>20</sup>  $\hat{y}$ . Finally, for each interpolation strategy, we calculate the RYEA. The results  
632 are shown in table 5.1.

---

<sup>19</sup>The choice of the modeling approach does not matter much, as long as it is general enough (i.e. able to approximate any function) and we use the same one for each interpolation strategy.

<sup>20</sup>By the integrated OOB estimates, we denote the predictions for each pixel where only trees are used, where the pixel has not been used (as  $n_{tree}$ , the number of Trees, grows the fraction of trees which do not contain a certain pixel converges to  $\frac{1}{e}$ ).

633 **Chapter 5**

634 **Results**

635 **5.1 XXX small recap from “Interpolation Methods”**

636 shoud w write 1:1 the sam es in the end of section 3

637 **5.2 Robustification and NDVI-Correction**

638 Discuss table

Table 5.1: XXX RMSE of yield prediction

	rf	lm-scl	lm-all	mars	gam	lasso	no-correction
ss	1.999	1.872	1.829	2.055	2.047	2.033	1.941
dl	1.873	1.886	1.896	1.988	1.898	1.833	2.018
ss-rob	1.895	2.010	2.037	1.970	1.874	1.928	1.880
dl-rob	1.865	1.884	2.002	1.996	1.808	1.875	2.005

# 639 Chapter 6

## 640 Discussion

### 641 6.1 Interpolation Methods

### 642 6.2 NDVI Correction

#### 643 6.2.1 Do we need to separate test and training data strictly by year?

644 While we could use this to evaluate whether our model learned a general pattern or only  
645 learned the given years. However, we have not used any ground truth at any point (until  
646 the evaluation). Instead, we estimated the “true” NDVI with the assumption 1 via OOB.  
647 Thus, we have bootstrapped our way out of the problem. Consequently, we reason that  
648 we can apply our method to a new (comparable) dataset and solve the correction again  
649 via this bootstrap.

#### 650 6.2.2 Shall We Use Additional Covariates?

652 In section 4.2.1 we have only used the spectral data (and the observational NDVI calculated  
653 from them) as covariates. Since we have the weather data available (cf. REF-SEC), it  
654 would be a small effort to incorporate it, together with statistics collected from it (i.e.  
655 GDD or ‘rainfall in the last 30 days’).

656 We decided against using this data, because on the one hand we have the problem that  
657 we have practically too few observations (we observe only 5 years) and we expect the  
658 weather in our study region to be rather homogeneous <sup>1</sup>. On the other hand, we want  
659 the underlying model not to learn improper relationships. For example, the model might  
660 automatically predict a high NDVI for a day in summer (detected by high GDD / many  
661 sunshine hours / high temperature) just because it is “used” to observing a lot of vegetation  
662 in summer. Including temporally (e.g.,  $P_{t-1}$  and  $P_{t+1}$ ) and geographically adjacent pixels  
663 would likely improve performance. However, for simplicity, we omit it here<sup>2</sup>.

664 - weight/uncertainty function (problem of weight function -> some outer points get really  
665 low weights (just because others in the middle have very little residuals and thus very high  
666 weight))

where  
does  
this sec-  
tion be-  
long to?  
Chapter  
‘NDVI  
Correc-  
tion’ or  
‘Further  
Work’?

<sup>1</sup>The weather data are published by Meteoswiss for a grid with a resolution of 1 km

<sup>2</sup>This is done for simplicity of understanding and using the model, since one would need to adapt  
to some convention of how to supply the data of adjacent pixels without redundancy (i.e. supplying  $P_t$   
multiple times).

667 **6.2.3 High RMSE in Yield Prediction**

- 668 How much can we expect to get? We have multiple sources of uncertainty in the data:
- 669 i.) Uncertainty in Yield data collected by the combine harvester
- 670 ii.) Uncertainty in Yield data through rasterization
- 671 iii.) Uncertainty in satellite images through “measurement errors” introduced via clouds  
672 and other atmospheric effects
- 673 iv.) Uncertainty introduced by interpolating (especially when long data-gaps are present)

674 **Chapter 7**

675 **Summary**

676

```
- itpl methods,  
  parametric dl  
  non-param  
  discarded  
  kernel methods because of strong bias  
  kriging because assumptions and highdim parameters  
  savitzky-golay filter since we will investigate the LOESS which can be thought a  
  loess slightly best performing itpl method but we notice non-smooth behaviour if  
  loess > ss > bspl  
  choose ss because of its meaningful definition (minimizing the integral of the second  
  - robustifying useful?
```

688

689 XXX Summarize the presented work. Why is it useful to the research field or institute?

690 **7.1 Future Work**

691 **7.1.1 Time Series Correction-Interpolation as a General Method**

692 Throughout this thesis, we developed a correction and interpolation method for the NDVI.  
693 However, we never used features of the NDVI. Only the parameter estimated via cross-  
694 validation in chapter 3.5 depends on the scale of the time series. For simplicity, we could  
695 thus determine the parameter using Generalized Cross Validation (as Ripley and Maechler  
696 suggests). Therefore, our approach of interpolation and correction of time series can be  
697 applied to arbitrary time series as long as additional information is available. However,  
698 further research is required, to demonstrate the usefulness of this approach in general.

699 **Example: Cloud Correction with Uncertainty Estimation and Interpolation**

700 This generalization can be used in particular for cloud correction. In the same manner as  
701 we corrected the NDVI time series in chapter 4, we can correct each spectral band and  
702 reunite the corrected bands with the uncertainties. If desired, the time series can also be  
703 interpolated before merging as in chapter 4.2.4. The resulting question would be how well  
704 this approach performs.

705 **7.1.2 Minor Improvements**

706 During this project, we also noticed some minor issues that we would have liked to invest-  
707 tigate further if more resources were available. The most relevant of these are:

- 708 — **Data:** Method how data has been extrapolated to the grid could possibly be improved  
709 — **Data:** For computational reasons we mostly considered all years and split the data  
710 (on the pixel level) randomly into a train/test set. A leave one year out cross  
711 validation might yield more accurate results.  
712 — **Data:** We have not included the spectral bands which have a resolution of 60m. But  
713 precisely these seem to be promising for cloud correction, since they are a proxy of  
714 the water (content and form) in the atmosphere.  
715 — **NDVI Correction:** Explore the effect of different link functions between the esti-  
716 mated absolute residuals and the weights in section 4.2.4.  
717 — **NDVI Correction:** Yield is not the only target variable of interest. Other variables  
718 like protein content could also be used in section 4.4 for the method evaluation.

# 719 Bibliography

- 720 Gaussian models for geostatistical data. In P. J. Diggle and P. J. Ribeiro (Eds.), *Model-  
721 Based Geostatistics*, pp. 46–78. Springer.
- 722 Beck, P. S. A., C. Atzberger, K. A. Høgda, B. Johansen, and A. K. Skidmore. Improved  
723 monitoring of vegetation dynamics at very high latitudes: A new method using MODIS  
724 NDVI. *100*(3), 321–334.
- 725 Breiman, L. Random Forests. *45*(1), 5–32.
- 726 Brockmann, M., T. Gasser, and E. Herrmann. Locally Adaptive Bandwidth Choice for  
727 Kernel Regression Estimators. *88*(424), 1302–1309.
- 728 Cao, R., Y. Chen, M. Shen, J. Chen, J. Zhou, C. Wang, and W. Yang. A simple method to  
729 improve the quality of NDVI time-series data by integrating spatiotemporal information  
730 with the Savitzky–Golay filter. *217*, 244–257.
- 731 Chen, J., P. Jönsson, M. Tamura, Z. Gu, B. Matsushita, and L. Eklundh. A simple method  
732 for reconstructing a high-quality NDVI time-series data set based on the Savitzky–Golay  
733 filter. *91*(3), 332–344.
- 734 Cleveland, W. S. Robust Locally Weighted Regression and Smoothing Scatterplots.  
735 *74*(368), 829–836.
- 736 Friedman, J. H. Multivariate Adaptive Regression Splines. *19*(1), 1–67.
- 737 Hastie, T. and R. Tibshirani. Generalized Additive Models: Some Applications. *82*(398),  
738 371–386.
- 739 Jaramaz, D., V. Perović, S. Belanovic Simic, E. Saljnikov, D. Cakmak, V. Mrvić, and  
740 L. Zivotic. The ESA Sentinel-2 mission Vegetation variables for Remote sensing of  
741 Plant monitoring.
- 742 Lyche, T. and K. Mørken. Spline Methods.
- 743 McMaster, G. S. and W. W. Wilhelm. Growing degree-days: One equation, two interpre-  
744 tations. *87*(4), 291–300.
- 745 Ripley, B. D. and M. Maechler. R: Fit a Smoothing Spline.
- 746 Savitzky, A. and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified  
747 Least Squares Procedures. *36*(8), 1627–1639.
- 748 Schafer, R. W. What Is a Savitzky–Golay Filter? [Lecture Notes]. *28*(4), 111–117.
- 749 Tibshirani, R. Regression shrinkage and selection via the lasso: A retrospective. *73*(3),  
750 273–282.

751 **Appendix A**

752 **Reproducibility**

753 **A.1 Reproduce Results**

754 For reproducibility of the whole computations we refere to our codebase at:

755 <https://github.com/LGraz/MasterThesis-Code>

756 In order to reproduce our computations and results, setup the directory as described  
757 in the README and execute the computations via `./shell_scripts/reproduce.sh`  
758 and do not execute the python and R scripts by hand (unless you follow the order in  
759 `./shell_scripts/reproduce.sh`).

760 **A.2 R-Package**

761 We also provide an R package for a general time series correction and interpolation if  
762 additional data is aviable at:

763 <https://github.com/LGraz/CorrectTimeSeries>

764 In our case we consider the NDVI time series and the additional data consists of the unused  
765 spectral bands.

766 We recommend installing it via the `devtools` package by:

767 `devtools::install_github("Greenstone/CorrectTimeSeries")`

768 In the following we shall give a stand-alone example of how the R package can be used:

```
769 1 library(CorrectTimeSeries)
770 2
771 3 # load a list of dataframes, each one describes one pixel with the covariates and
772 4 # the response
773 5 data(timeseries_list)
774 6 str(timeseries_list[[1]])
775 7
776 8 # Train/Load RF
777 9 train_model_myself <- TRUE
778 10 if (train_model_myself){
779 11     # Add "true" NDVI (or generally the response), by Out-Of-Bag estimation
780 12     timeseries_list <- lapply(timeseries_list, function(df) {
781 13         df$oob_ndvi <- OOB_est(df$gdd, df$ndvi_observed) # gdd is the time-axis
782 14         df
783 15     })
784 16     # Train correction model
785 17     formula <- "oob_ndvi ~ B02+B03+B04+B05+B06+B07+B08+B8A+B11+B12+scl_class"
786 18     RF <- train_RF_with_fromula(formula, timeseries_list, robustify=TRUE)
787 19 } else {
```

```
789 19  data(RF_for_NDVI)
790 20  RF <- RF_for_NDVI
791 21 }
792 22
793 23 # ADD CORRECTION
794 24 timeseries_list <- lapply(timeseries_list, function(df) {
795 25   df$corrected_ndvi <- randomForest:::predict.randomForest(RF, df)
796 26   df
797 27 })
798 28
799 29 # Get interpolation for each timeseries
800 30 newx <- 1:1000
801 31 lapply(timeseries_list, function(df){
802 32   ss <- smoothing_spline(df$gdd, df$corrected_ndvi)
803 33   predict(ss, newx)$y
804 34 })
```

Example of how to use the `CorrectTimeSeries` package

806 **Appendix B**

807 **Further Material**

808 **B.1 Interpolation**

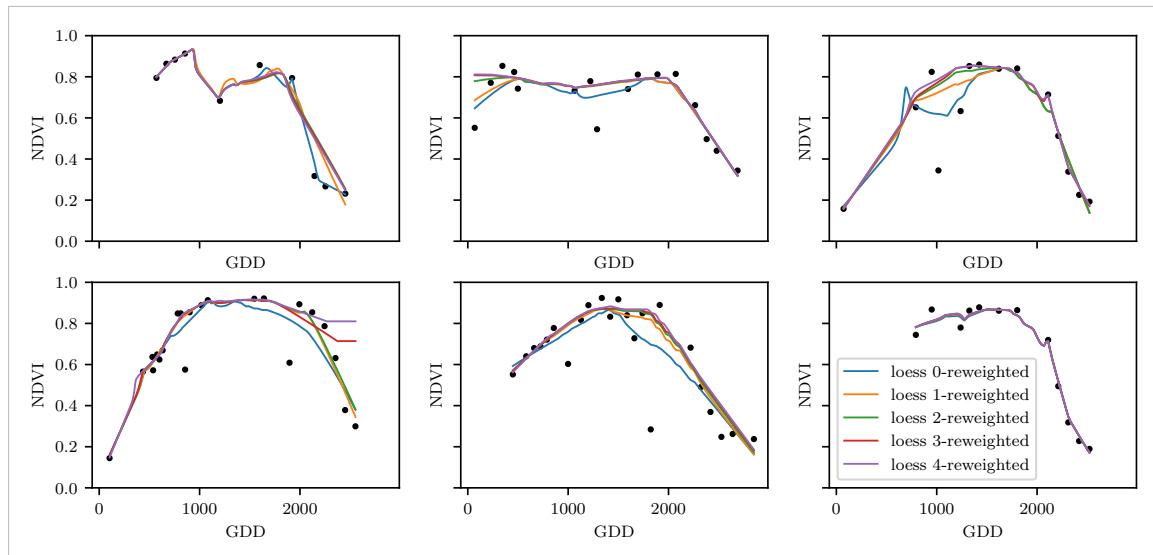


Figure B.1: The LOESS smoother fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.6) are also displayed

809 **B.2 NDVI correction**

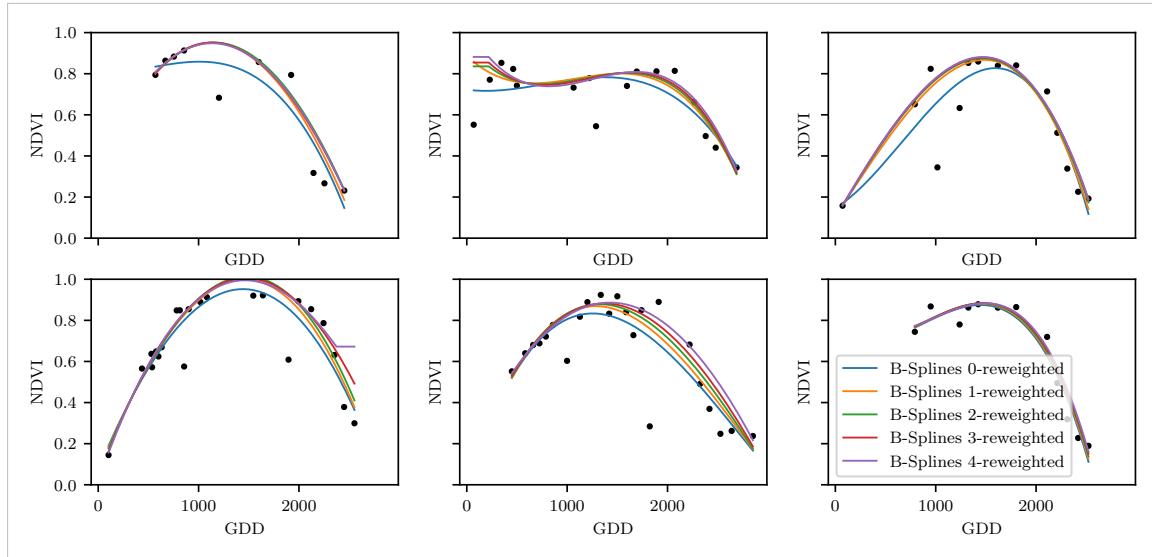


Figure B.2: B-Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.6) are also displayed

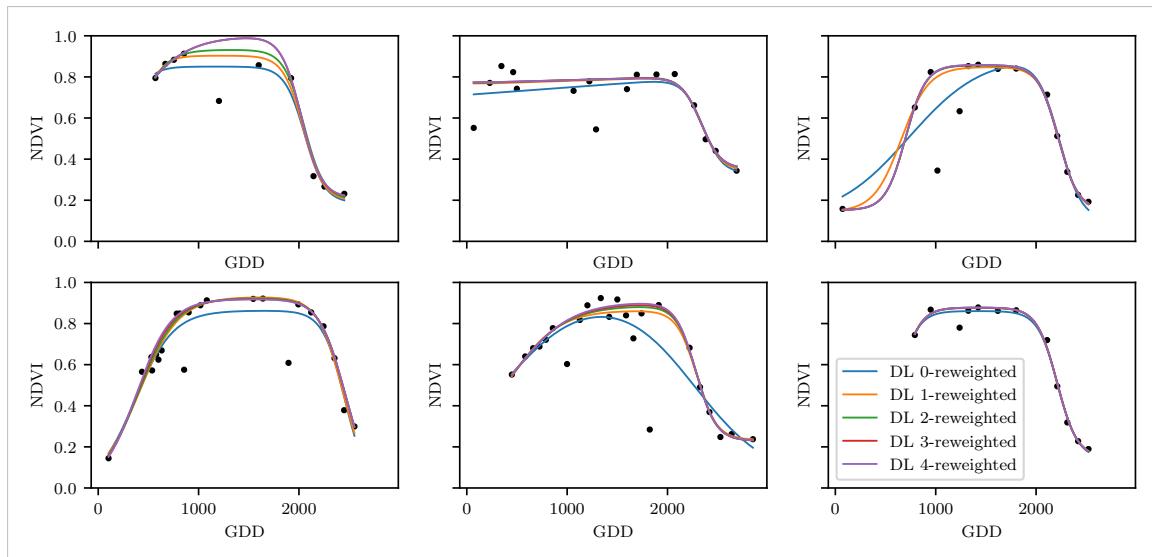


Figure B.3: A Double Logistic curve fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.6) are also displayed

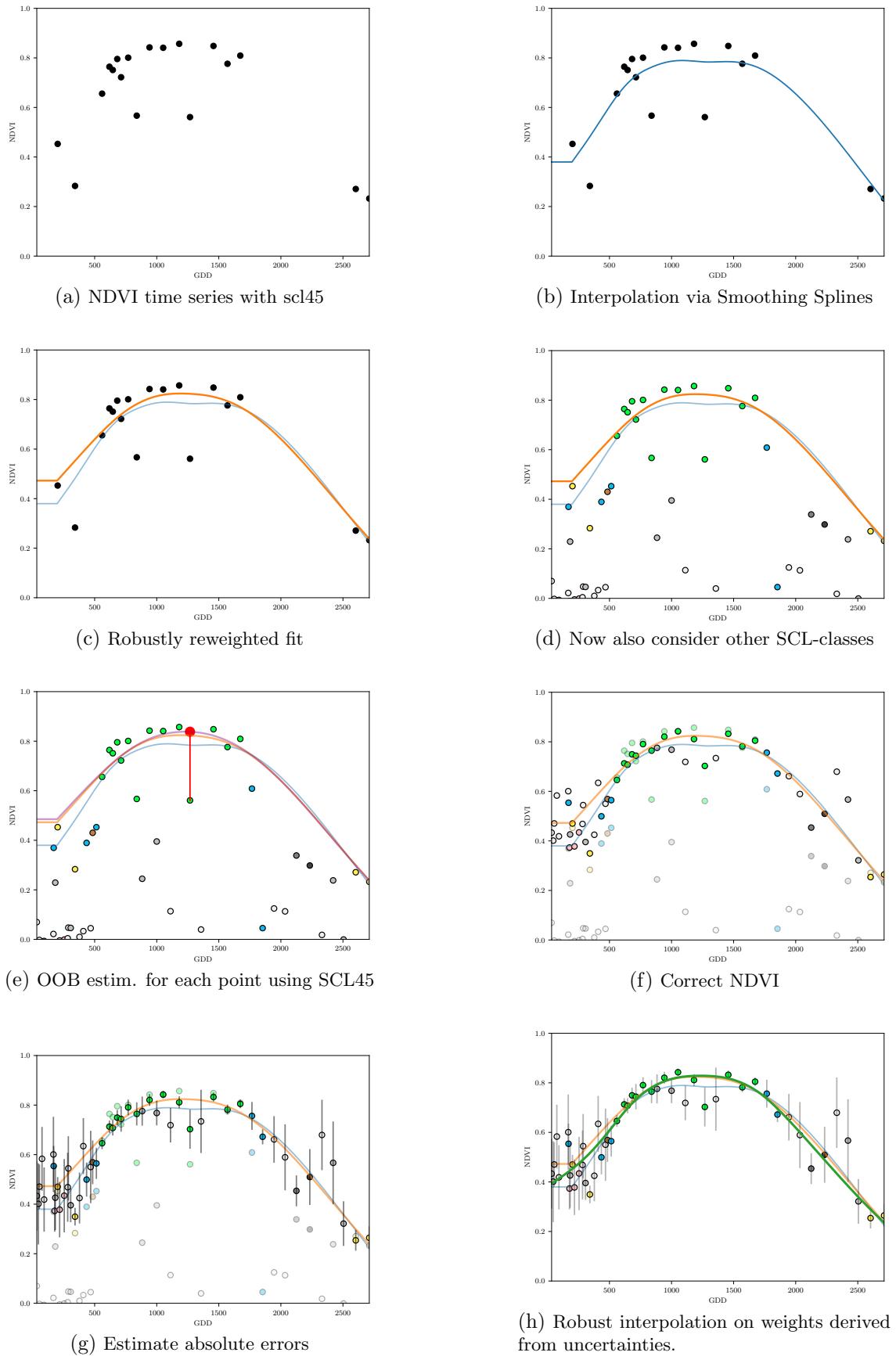


Figure B.4: Stepwise illustration of robust NDVI-Correction. For the color encoding of the SCL classes we refer to table 2.2.