

1 Contents

2	Notation	ii
3	1 Introduction	1
4	1.1 XXX motivation - why is it important	1
5	1.2 XXX problembaum / fragestellungen	1
6	1.3 XXX State-of-the-art	1
7	1.4 Roadmap	1
8	2 Problem Description	2
9	2.1 Available Data	2
10	2.1.1 Sentinel 2 Satellite Image Data	2
11	2.1.2 Yieldmapping Data	4
12	2.1.3 Gather Data	4
13	3 Interpolation Methods	7
14	3.1 Setting	7
15	3.2 XXX DAS vs GDD	7
16	3.3 Robustify	7
17	3.3.1 XXX Our Adjustment:	9
18	3.4 Parametric Regression	9
19	3.4.1 Double Logistic	9
20	3.4.2 Fourier Approximation	10
21	3.5 Non-Parametric Regression	10
22	3.5.1 Kernel Regression	11
23	3.5.2 Kriging	11
24	3.5.3 Savitzky-Golay Filter (SG Filter)	12
25	3.5.4 Locally Weighted Regression (LOESS)	14
26	3.5.5 B-splines	15
27	3.5.6 Natural Smoothing Splines	16
28	3.5.7 XXX Whittaker Smoother	16
29	3.6 Tuning parameter estimation	16
30	3.7 Robustification – Recap	17
31	3.7.1 Upper Envelope Approach - Penalty for negative resiudals	17
32	3.8 Performance Assecement	17
33	4 NDVI Correction / Improve NDVI Data	20
34	4.1 Considering other SCL Classes	20
35	4.2 XXX Correction	21
36	4.2.1 Response and Covariates	21
37	4.2.2 Correction Methods	21
38	4.2.3 Uncertainty Estimation	26
39	4.2.4 Interpolation	26
40	4.3 Resulting Interpolation Stragegies	26
41	4.4 XXX Evaluation Method	27
42	4.4.1 XXX Idea	27
43	4.4.2 Yield Estimation	27

44	5 Results	29
45	5.1 XXX small recap from “Interpolation Methods”	29
46	5.2 Robustification and NDVI-Correction	29
47	6 Discussion	30
48	6.1 NDVI Correction	30
49	6.1.1 Shall We Use Additional Covariates?	30
50	7 Outlook	31
51	7.1 Data	31
52	7.2 Interpolation	31
53	7.3 NDVI Correction	31
54	7.4 NDVI Correction + +	31
55	Bibliography	32
56	A XXX Appendix	34

57 Notation

- 58 c : a (vector of) constant(s)
- 59 $\lambda \in \mathbb{R}$: a scalar
- 60 $n \in \mathcal{N}$: sample size
- 61 i, j are indices in $\{1, \dots, n\}$
- 62 $x \in \mathbb{R}^n$: covariate in 1-dim interpolation setting
- 63 $w \in \mathbb{R}^n$: a vector of weights for each location x
- 64 $y \in \mathbb{R}^n$: response in 1-dim interpolation setting
- 65 $\hat{y} \in \mathbb{R}^n$: estimate of y
- 66 $\bar{y} \in \mathbb{R}$: mean of y
- 67 $r \in \mathbb{R}^n$: residuals given by $y - \hat{y}$
- 68 Pixel: A pixel describes a specific location in a field. It has the size of 10 x 10 meters
69 and coincides with the resolution (and location) of the sentinel-2 pixels. Such pixels are
70 illustrated in figure ???. Additional information like yield is also attached.
- 71 P_t : this describes the observed data (weather and spectral bands) at time t and the location
72 of one pixel.
- 73 P : a pixel. We see it as a collection of all the observations at the specified location within
74 one season. More formally, $P := \{P_t | t \text{ is a valid sample time within a defined season}\}$
- 75 SCL: scene classification layer. This indicates what one can expect at a pixel at a sampled
76 time. For an overview cf. table 2.2
- 77 P^{SCL45} : similar to P but we only consider observations which belong to the classes 4 and
78 5. This is used done to get a subset of observations which are less contaminated by clouds
79 and shadows.
- 80 NDVI: normalized vegetation difference index
- 81 DAS: days after sowing
- 82 GDD: growing degree days – cumulative sum of (temperature – threshold)⁺
- 83 XXX ML models and their shortnames
- 84 RYEA : relative yield-estimation-accuracy. Definition 4.4.0.1

85 **Chapter 1**

86 **Introduction**

87 **1.1 XXX motivation - why is it important**

- 88 - NDVI-timeseries is very simple and widely used. Examples are: - Plant Models REF -
89 Season Start (start of spring) (community name: land-surface-plant-phenology) -
90 Since satellite images are “for free” researchers extract

91 **1.2 XXX problebaum / fragestellungen**

92 problem schilderung anhand des Leitfadens: **pictures?**

93 **1.3 XXX State-of-the-art**

- 94 zusammenfassung mit literaturrecherche hier:
95 — Doublelogistic (winter-ndvi)
96 — parametric / non-parametric approaches
97 — spatio-temporal approaches

98 **1.4 Roadmap**

99 In chapter

100 **Chapter 2**

101 **Problem Description**

102 **2.1 Available Data**

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

106 **2.1.1 Sentinel 2 Satellite Image Data**

107 **General Information**

108 The European Space Agency (ESA)¹ freely distributes the high quality images of the two
109 Sentinel satellites 2 (S2). Together, both satellites have a revisit time of 5 days at the
110 equator and 2-3 at mid-latitudes. However, at our study region we only receive an image
111 every 5 days. In order to decrease the effect of atmospheric conditions like reflections
112 and scattering, we will not work with the raw data but with the results of the Level-2A
113 processing²³.

114 **Data Description**

115 The Level-2A processed images we use contain 12 spectral bands with local resolutions up
116 to 10 meters (see 2.1). Bands which have a lower resolution (20 and 60 meters) will be
117 scaled up to 10 meters using cubic interpolation (REF gregor perich). Additional to the
118 spectral bands the ESA also supplies a Scene Classification Layer (*SCL*) where for each
119 location the observed subject is assigned to an *SCL-class* (cf. table 2.2). In chapter 3 we
120 will use this classification to filter out unreliable data points considering only SCL-classes
121 4 and 5.

122 **Data Illustration**

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

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

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

³XXXREF 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”

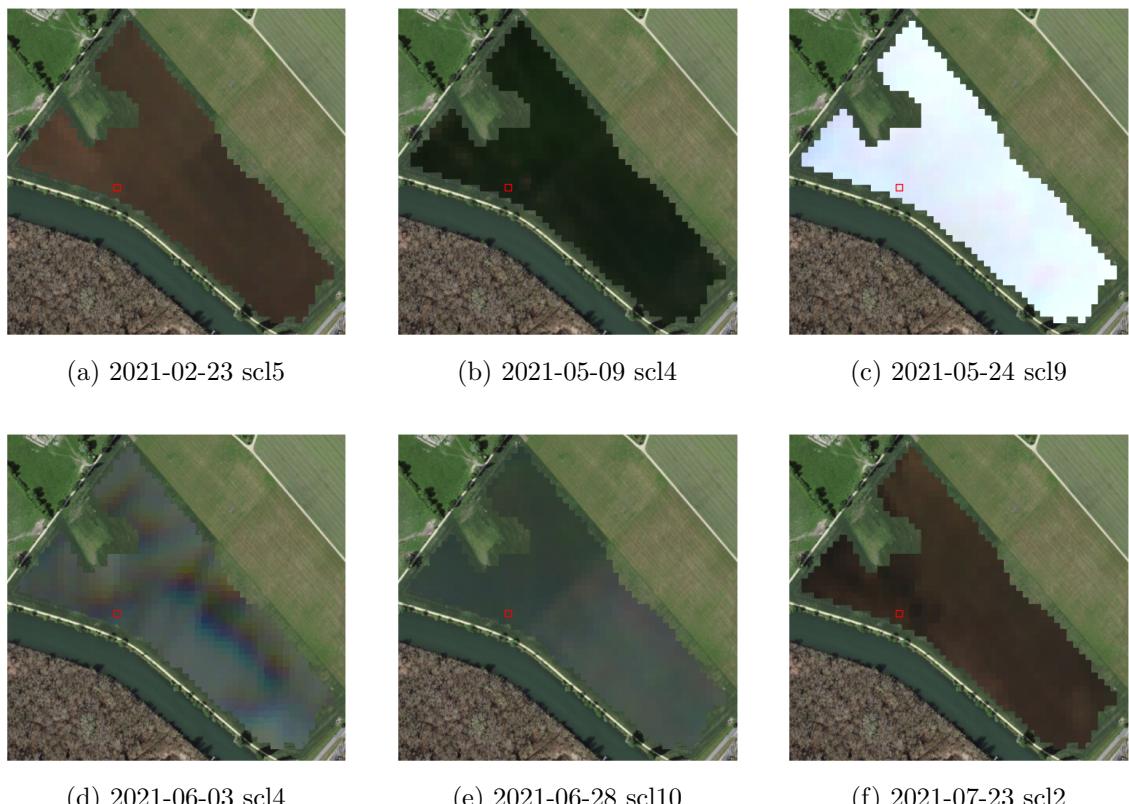


Figure 2.1: Satellite images of a field at selected times with a static background for orientation. The SCL-class of the highlighted pixel is provided in the respective subtitle. (???xxx include scl legend?)

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 λ in nm with the spectral width $\Delta\lambda$ in nm with a spatial resolution SR in m.

Band	λ	$\Delta\lambda$	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.

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

129 2.1.2 Yieldmapping Data

130 The crop yield data were collected using a combine harvester. Equipped with GPS, the
 131 harvester drives over the fields and continuously estimates the crop density in t/ha (see fig.
 132 2.2a). We take the data set derived from this in REF-Gregor-Perich, where error-prone
 133 measurement points (such as during an egen curve) were removed and then the yield map
 134 was rasterized using linear interpolation (cf. fig. 2.2b).

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

139 2.1.3 Gather Data

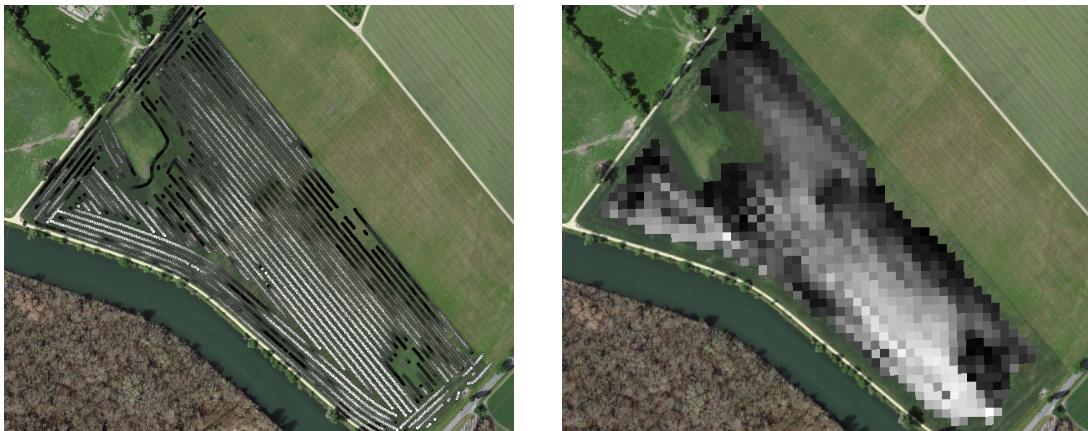
140 Before we join all the data, we define a few concepts.

141 Using bands $B4$ and $B8$, we calculate the well-known Normalized Difference Vegetation
 142 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)

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

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

$$GDD_n := \sum_{i=0}^n \max(T_i - T_{base}, 0). \quad (2.1.3.1)$$

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

⁴XXX For cereals we use $T_{base} = 0$

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

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

158 **Chapter 3**

159 **Interpolation Methods**

160 In this section, we take a closer look at several interpolation methods, which will be used
161 to interpolate and smooth the NDVI time series.

162 First, we give a brief overview in table 3.1.

163 Second, we define the general setting and discuss a general approach to make the interpo-
164 lation more robust (i.e. reduce the impact of outliers).

165 Later, we introduce and discuss each method.

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

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

169 **3.1 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 ε_i is some noise and $m : \mathbb{R} \rightarrow \mathbb{R}$ being 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

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

170 Different assumptions on m will lead to the following methods:

171 **3.2 XXX DAS vs GDD**

172 equation (2.1.3.1)

173 **3.3 Robustify**

174 Now we discuss a general approach of how to robustify an interpolation. The main idea
175 is to give less weight to observations which have high residuals after the initial (or if we
176 reiterate, the last) fit.

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).

	assumptions	pros	cons	w	b
Savitzky-Golay filter	<ul style="list-style-type: none"> - high frequencies are noise (low.pass filter) - equidistant points - local polynomials 	<ul style="list-style-type: none"> - computationally very fast 	<ul style="list-style-type: none"> - cannot deal natively with missing data (need some interpolation) 	no	(yes)
SG + NDVI	<ul style="list-style-type: none"> - upper envelope - vegetation cannot grow faster than some slope 	<ul style="list-style-type: none"> - biological knowledge 	<ul style="list-style-type: none"> - bad “upper envelope” since weights are not used for the estimation itself 	(no)	(yes)
Loess	<ul style="list-style-type: none"> - local polynomial with points closer to the estimated point are more important 	<ul style="list-style-type: none"> - flexible - generalization of SG - weighting function makes intuitive sense 	<ul style="list-style-type: none"> - computationally expensive 	yes	(yes)
Smoothing Splines	<ul style="list-style-type: none"> - 2cd derivative of function is integrable 	<ul style="list-style-type: none"> - intuitive meaning of penalty - general assumptions - flexible shape 	<ul style="list-style-type: none"> - unbounded 	yes	no
B-Splines (Smoothed)	<ul style="list-style-type: none"> - function can be approximated by a linear combination of B-splines basis functions 	<ul style="list-style-type: none"> - general assumption - flexible shape 	<ul style="list-style-type: none"> - unbounded - no intuitive meaning for smoothing 	yes	no
(Gaussian) Kernel Smoothing		<ul style="list-style-type: none"> - simple - general assumptions 	<ul style="list-style-type: none"> - bandwidth: fails if there are big data-gaps 	yes	yes
Double-Logistic	<ul style="list-style-type: none"> - function first increases then decreases - ndvi has a minimal value 	<ul style="list-style-type: none"> - good for evergreen plants (if snow masks ndvi) - upper envelope 	<ul style="list-style-type: none"> - parameterestimation can go seriously wrong - strange behaviour for long data-gaps 	yes	(yes)
Universal Kriging	<ul style="list-style-type: none"> - function is a realization of a stationary gaussian process 	<ul style="list-style-type: none"> - informative parameters - flexible 	<ul style="list-style-type: none"> - regression to the mean - assumptions clearly not met 	yes	(yes)

¹⁷⁷ Even though the procedure is taken from the robust version of the LOESS smoother (cf. section 3.5.4 and [Cleveland \(Cleveland\)](#)), we discuss it now because we will apply it also to other interpolation methods.

¹⁸⁰ XXX¹

Before we describe the procedure, we define a function which will determine the weight given to each observation such that observations with large scaled residuals will have less

¹Note that due to using the median for the normalization, we gain a breakdown point of 50% for outliers in y .

weight. That is the bisquare function B:

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

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

$$w_i^{\text{new}} := w_i^{\text{old}} B\left(\frac{|r_i|}{6 \text{mad}(r_1, \dots, r_n)}\right)$$

183 where $r_i = y_i - \hat{y}_i$ denotes the residuals. We can iterate this reweighting and stop after
 184 several steps or when the change of the values is smaller than some tolerance.

185 Examples of such iterative fits are illustrated in the figures 3.4 3.5, 3.6, 3.4 and 3.7.

186 3.3.1 XXX Our Adjustment:

Since we usually observe outliers with negative residuals we decide to divide the negative residuals by two(XXX) before updating the weights. Furthermore, 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

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

187 for $r, w \in \mathbb{R}^n$

188 3.4 Parametric Regression

189 Parametric Curve estimation tries to fit a parametric function (e.g. a Gaussian function
 190 with parameter μ and σ) to a dataset. In the following, we introduce 2 such parametric
 191 approaches.

192 Optimization Issues

193 Since we aim to minimize the residuals sum of squares over 5 (or 6) parameters, we try
 194 to solve a non-convex optimization problem. Thus, the algorithm² either struggles to find
 195 the global minimum or fails to converge. This was fixed by providing for each parameter
 196 reasonable initial values and generous bounds (which match our experience).

197 3.4.1 Double Logistic

198 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
 199 NDVI time series).

200 Assumptions:

- 202 — There is a minimum NDVI level Y_{\min} in the winter (e.g. due to evergreen plants),
 203 which might be masked by snow. This can be estimated beforehand, taking into
 204 several years into account.

²We used the python function `scipy.optimize.curve_fit`

- 205 — The growth cycle can be divided into an increase and a decrease period, where
 206 the time series follows a logistic function. The maximum increase (or decrease) is
 207 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:

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

208 Where the five free parameters: Y_{\max} , d_0 , d_1 , t_0 , t_1 are initially estimated by least squares.
 209 Such fit can be seen in figure 3.1.

210 Similar as for the Savitzky-Golay Filter (cf. section 3.5.3) we reestimate (only once) the
 211 parameters by giving less weight to the overestimated observations and more weight to
 212 the underestimated observations³.

Pros	Cons
<ul style="list-style-type: none"> — Incorporates subject specific knowledge in the case of evergreen plants covered in snow. — Optimized parameters have an intuitive meaning. 	<ul style="list-style-type: none"> — Strong shape assumptions on the NDVI curve. — Parameter optimization might go wrong. This can be mitigated to some extent to provide bounds for the parameters — Strange behavior in regions with little observations. (cf. figure 3.1)

213 3.4.2 Fourier Approximation

Similar as in section 3.4.1 we fit a parametric curve to the data by least squares. Here we take the second order Fourier series:

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

214 where $\Phi = 2\pi \times (t - 1)/n$.

Pros	Cons
<ul style="list-style-type: none"> — Assumption of periodicity can be helpful if we are modelling multiyear grow cycles — Flexible curve shape 	<ul style="list-style-type: none"> — Bad behavior in regions with little data (cf. figure 3.1) — Hard to interpret estimated parameters — Parameter estimation can go wrong. Introducing bounds can help.

215 3.5 Non-Parametric Regression

217 In non-parametric curve estimation, we no longer demand our curve to be fully determined
 218 by several parameters, but we allow it to also dependent on the data. That said, we might
 219 still use some tuning-parameters sometimes.

TODO:
include
Weighted
versions

³For the details on the weights we refer to Beck, Atzberger, Høgda, Johansen, and Skidmore (Beck et al.)

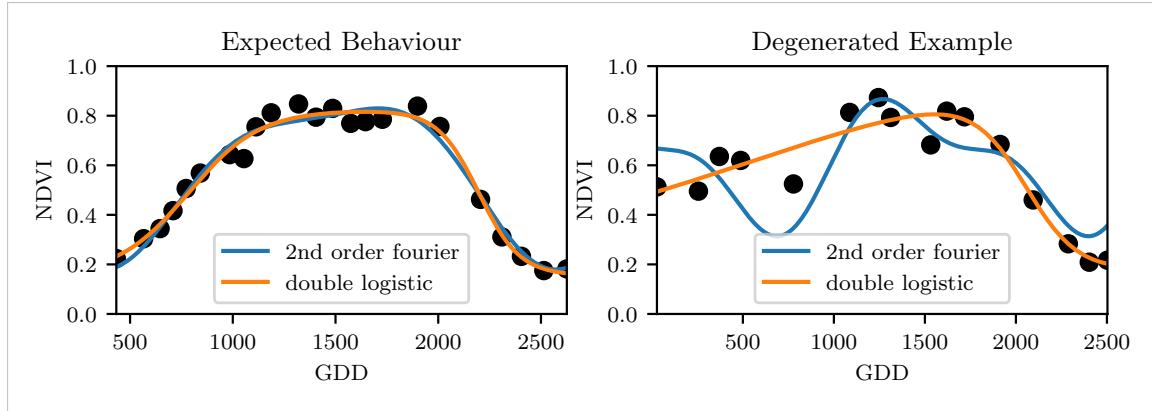


Figure 3.1: Here we observe the nice fitting possibilities of the two parametric methods but notice also some misbehavior

220 3.5.1 Kernel Regression

221 As described previously, we would like to estimate

$$\mathbb{E}[Y \mid X = x] = \int_{\mathbb{R}} y f_{Y|X}(y \mid x) dy = \frac{\int_{\mathbb{R}} y f_{X,Y}(x, y) dy}{f_X(x)}, \quad (3.5.1.1)$$

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

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

By plugging the above into equation (3.5.1.1) we arrive at the *Nadaraya-Watson* kernel estimator:

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

Pros

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

222 **Examples:** Normal, Box For local bandwidth selection see Brockmann et al. (1993)

223 XXX

224 3.5.2 Kriging

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

230 **Definitions and Assumptions**

- 231 A *Gaussian Process* $\{S(t) : t \in \mathbb{R}\}$ is a stochastic process if $(S(t_1), \dots, S(t_k))$ has a multi-
 232 variate Gaussian distribution for every collection of times t_1, \dots, t_k . S can be fully charac-
 233 terized by the mean $\mu(t) := E[S(t)]$ and its covariance function $\gamma(t, t') = \text{Cov}(S(t), S(t'))$
 234 Assumption: We will assume the Gaussian process to be stationary. That is for $\mu(t)$ to be
 235 constant in t and $\gamma(t, t')$ to depend only on $h = t - t'$. Thus, we will write in the following
 236 only $\gamma(h)$.⁴

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

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

- 237 where h is the distance, n is the nugget, r is the range and p is the partial sill visualized
 in figure 3.2.⁵

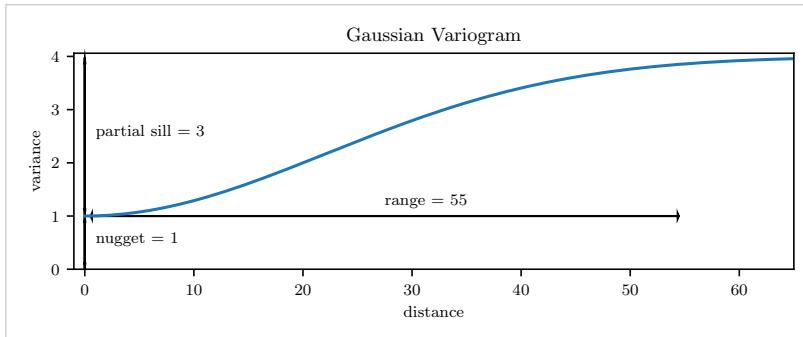


Figure 3.2: Gaussian Variogram with nugget=1, partial sill=3, range=55

238

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

239 **3.5.3 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

⁴Note 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.

⁵Strictly speaking we use a scaled version of the variogram. Thus, only the ratio of p/n matters.

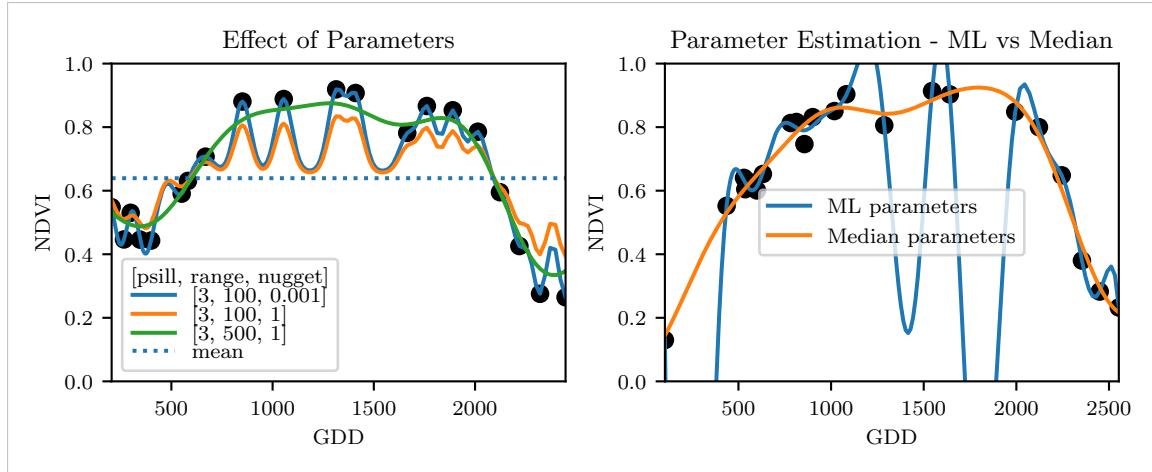


Figure 3.3: On the left, we see how the interpolation change if we increase the nugget and the range parameter. On the right we compare two kriging interpolations, where one takes parameters by numerically maximizing the (which results in a very small nugget) and the other takes the median of many such numerical optimizations.

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:

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

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

For equidistant points this can efficiently be calculated by

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

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

Adaptation to the NDVI

In a rather famous paper [Chen, Jönsson, Tamura, Gu, Matsushita, and Eklundh \(Chen et al.\)](#) a “robust” method based on the Savitzky-Golay has been used. The method is based on the assumption that due to atmospheric effects the observed NDVI tends to be underestimated and that it cannot increase too quickly⁶.

Algorithm:

- i.) Remove points which are labeled as cloudy.
- ii.) Remove points which would indicate an increase greater than 0.4 within 20 days.
- iii.) Linearly interpolate to obtain an equidistant time series X^0 .
- iv.) Apply the Savitzky-Golay Filter to obtain a new time series X^1 .

⁶The latter is argued by the biological impossibility of such fast vegetation changes

252 v.) Update X^1 by applying again a Savitzky-Golay Filter. Repeat this until $w^T |X^1 - X^0|$
 253 stops decreasing, where w is a weight vector with $w_i = \min\left(1, 1 - \frac{X_i^1 - X_i^0}{\max_i \|X_i^1 - X_i^0\|}\right)$.
 254 This reduces the penalty introduced by outliers⁷ and by repeating this step we approach the “upper NDVI envelope”.

Pros	Cons
— Popular technique in signal processing.	— No natural way of how to estimate points which are not in the data.
— Efficient calculation for equidistant points.	— Not generalizable to other spectral indices.
— Upper envelope matches intuition for the NDVI. Therefore, it is robust against outliers with small values.	— Linear interpolation to account for missing data might be not appropriate.
	— No smooth interpolation between two measurements.

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

257 One notable adaptation of the Savitzky-Golay is the presented by Cao, Chen, Shen, Chen,
 258 Zhou, Wang, and Yang (Cao et al.). The key difference is the additional assumption of the
 259 cloud cover being discontinuous and that we can improve by looking at adjacent pixels⁸.
 260 Because we are working with rather high resolution satellite data, and we need the variance
 261 in the predictors, we will waive this extension.

262 **3.5.4 Locally Weighted Regression (LOESS)**

263 Introduced by : Cleveland (Cleveland) implemented here Cappellari, McDermid, Alatalo,
 264 Blitz, Bois, Bournaud, Bureau, Crocker, Davies, Davis, de Zeeuw, Duc, Emsellem, Khoch-
 265 far, Krajnović, Kuntschner, Morganti, Naab, Oosterloo, Sarzi, Scott, Serra, Weijmans,
 266 and Young (Cappellari et al.)

267 The Locally Weighted Regression (LOESS) can be understood as a generalization of the
 268 Savitzky-Golay Filter (cf. sec. 3.5.3).

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

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

269 where h_i is the minimal distance such that $\lceil \alpha n \rceil$ observations are in the ball $B_{h_i}(x_i)$.⁹ So
 270 for each y_i we only consider a proportion α of the observations.

⁷Here we call a point i an outlier if $X_i^0 < X_i^1$.

⁸Here, 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

⁹If too many weights are set to zero, we might end up considering not enough observations and thus 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.

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

272 The LOESS smoother takes a fraction of points instead of a fixed number and therefore
 273 automatically adapts to the size of the data we wish to interpolate. However, we run
 274 into the danger of considering too little observations, since the estimation breaks down if
 275 $[an] < d + 1$. Furthermore, LOESS gives less weight to points further away. This yields a
 276 "smoother" estimate, since when we slide the window (e.g. for estimating the next value)
 277 an influential point at the border does not suddenly get zero weight from being weighted
 278 equally before. Finally, the LOESS also can be used for non-equidistant data and allows
 279 for arbitrary interpolation.

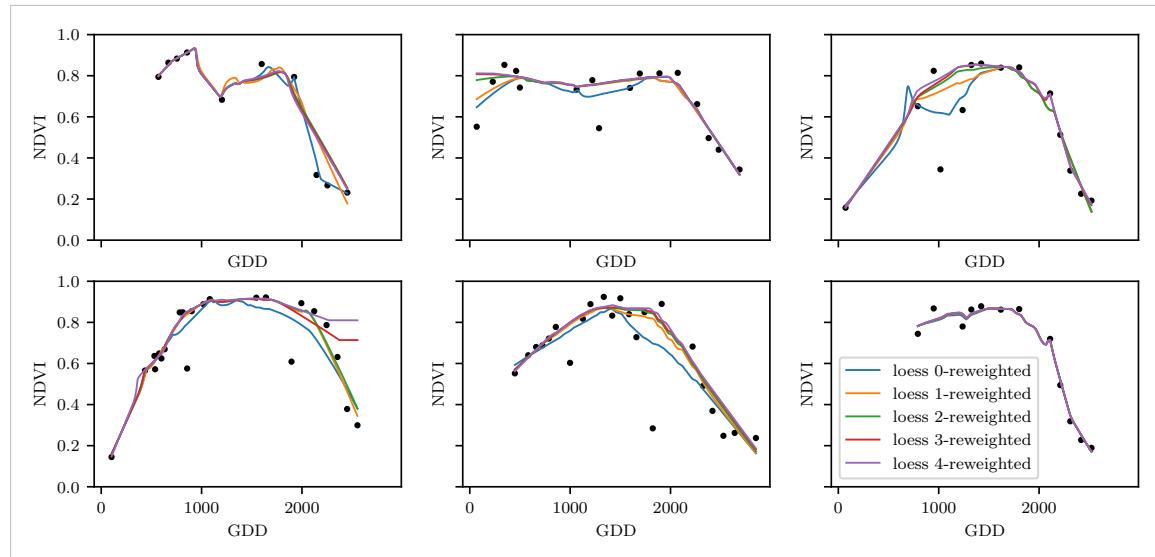


Figure 3.4: The LOESS smoother fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.3) are also displayed

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	— Multiple XXXXXXx
— Intuitive parameters	

280 **3.5.5 B-splines**

from [Lyche and Mørken](#) ([Lyche and Mørken](#))

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

$$B_{i,0}(x) = 1, \text{ if } t_i \leq x < t_{i+1}, \text{ otherwise } 0$$

$$B_{i,k}(x) = \frac{x-t_i}{t_{i+k}-t_i} B_{i,k-1}(x) + \frac{t_{i+k+1}-x}{t_{i+k+1}-t_{i+1}} B_{i+1,k-1}(x)$$

281 ****Smoothing:**** We can relax the constraint that we have to perfectly interpolate. Thus,
 282 we use the minimum number of knots¹⁰ such that: $\sum_{i=1}^n (w(y_i - \hat{y}_i))^2 \leq s$

¹⁰SciPy uses FITPACK and DFITPACK, the documentation suggests that smoothness is achieved by

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

283 **3.5.6 Natural Smoothing Splines**

Let \mathcal{F} be the Sobolev space (the space of functions of which the second derivative is integrable). Then the unique¹¹ minimizer

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

284 is a natural¹² cubic spline (i.e. a piecewise cubic polynomial function). The objective
 285 function has an intuitive meaning, as to avoid lateral acceleration it is desirable to move
 286 the steering wheel as little as possible, when driving a car.

Pros	Cons
— can be assigned degrees of freedom (trace of the hat-matrix)	— choose λ
— efficient estimation (closed form solution)	
— intuitive penalty (we don't want the function to be too "wobbly" — change slopes)	
— performs also well if points are not equidistant	
— fixes the Runge's phenomenon (fluctuation of high degree polynomial interpolation)	

287 **3.5.7 XXX Whittaker Smoother**

288 XXX

289 **3.6 Tuning parameter estimation**

290 lots of cross validation
 291 what is the best? RMSE is bad, since we know that outliers are present optimizing w.r.t
 292 different statistics
 293 ?plot with different densities for each statistic

reducing the number knots used

¹¹Strictly speaking it is only unique for $\lambda > 0$

¹²It is called natural since it is affine outside the data range ($\forall x \notin [x_1, x_n] : \hat{m}''(x) = 0$)

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

294 3.7 Robustification – Recap

- 295 introduced in section ?? we want to review it
 296 robustifieng from loess -> lets try it with all. Result in figures ...
 297 issues when reiterating often (we lose some points completely)
 298 from pictures ... we get that one

299 3.7.1 Upper Envelope Approach - Penalty for negative resiudals

300 discussion of idea, and explenation why we did no use it (arbitrary choice)

301 3.8 Performance Assecement

302 TEMP — Figures

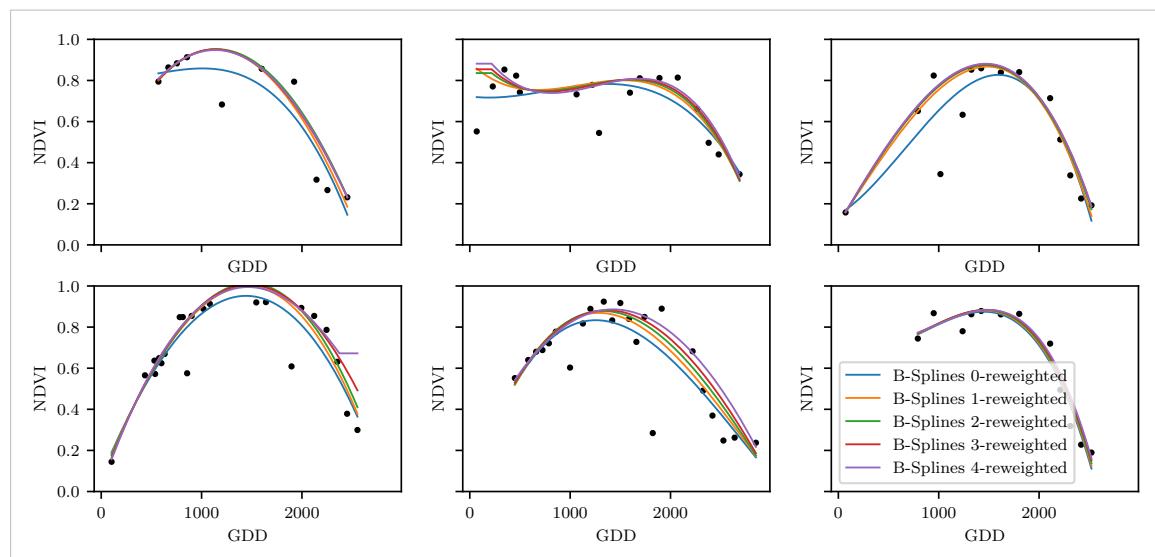


Figure 3.5: B-Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.3) are also displayed

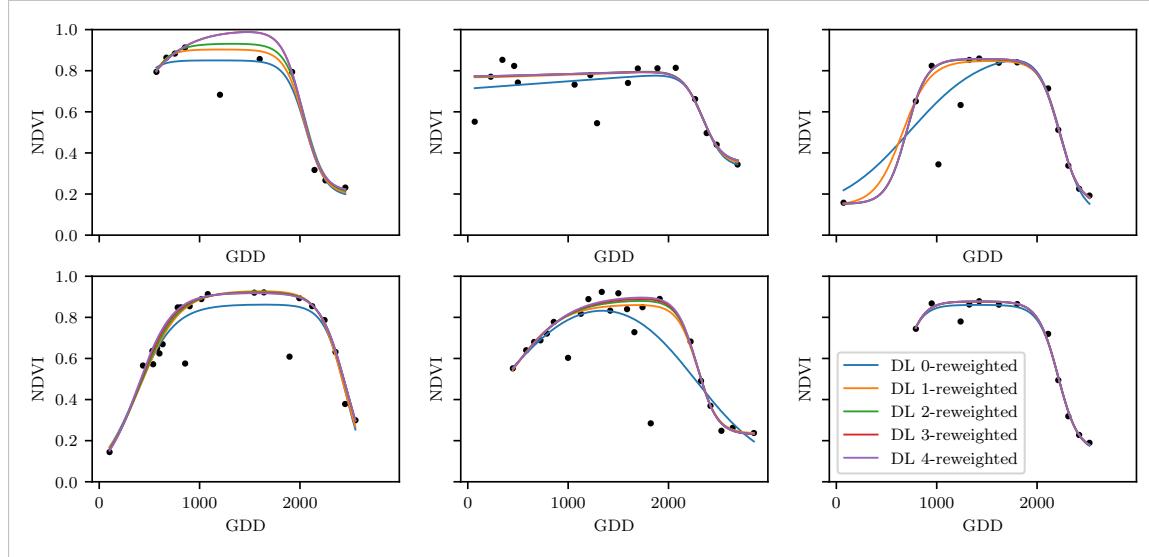


Figure 3.6: A Double Logistic curve fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.3) are also displayed

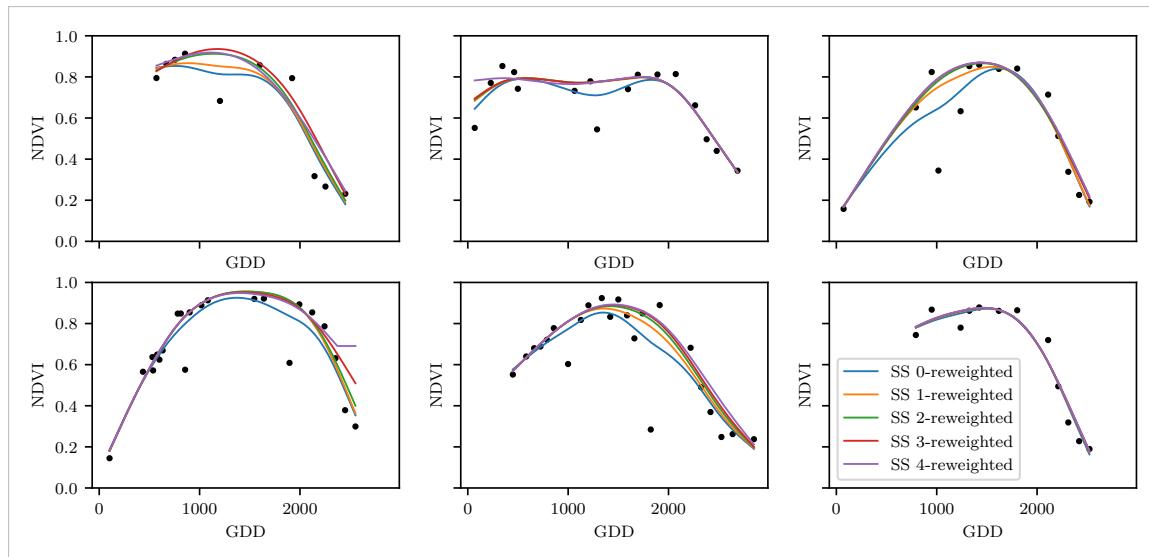


Figure 3.7: Smoothing Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifying refit (as indicated in section 3.3) are also displayed

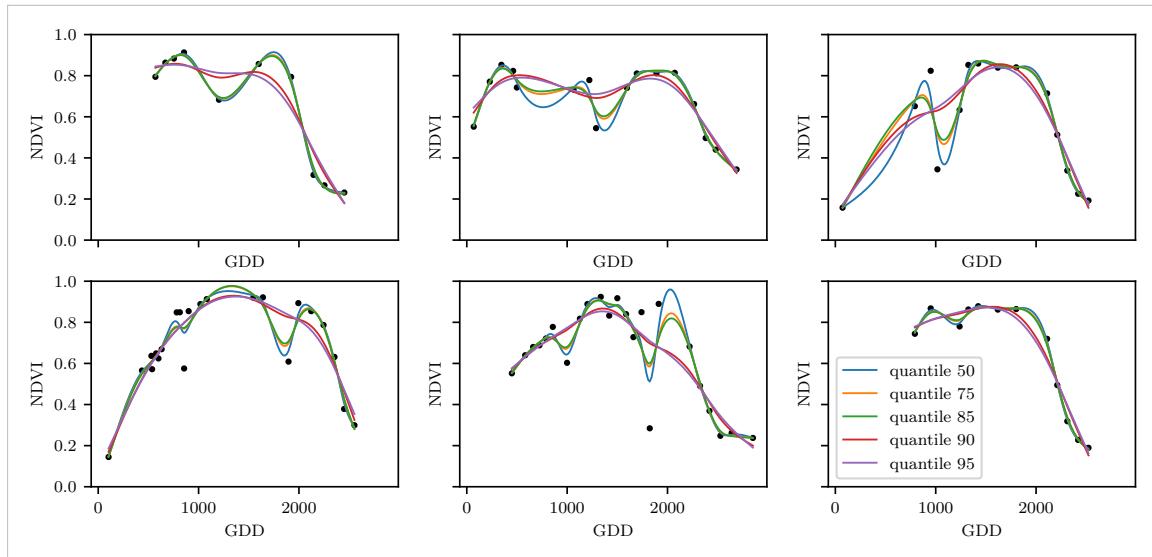


Figure 3.8: Smoothing splines fit with smoothing parameter optimized by minimizing the “...”-quantile of the absolute leave-one-out residuals. Note that the larger the considered quantile is, the smoother the resulting curve becomes.

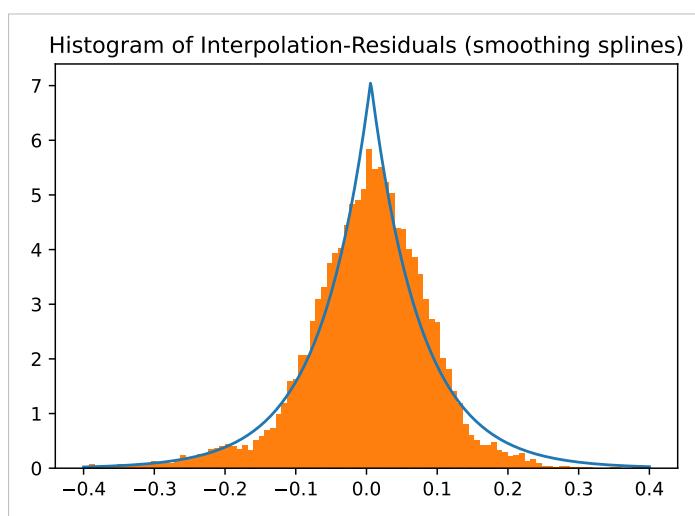


Figure 3.9: XXX caption XXX

303 **Chapter 4**

304 **NDVI Correction / Improve NDVI
305 Data**

306 Let's remind ourselves that the data from the Sentinel-2 is equipped with a scene classi-
307 fication layer (SCL) and we therefore have some information of what is observed at each
308 pixel for each sampled time (cf. table 2.2). In this chapter we would like to improve the
309 observed NDVI values by using more information than just the two bands used to calculate
310 the NDVI (B4 and B8).

311 **4.1 Considering other SCL Classes**

312 In figure 4.1 we see for example that some blue points¹ follow the interpolated line closely
313 and that they might be useful in improving an interpolation fit.

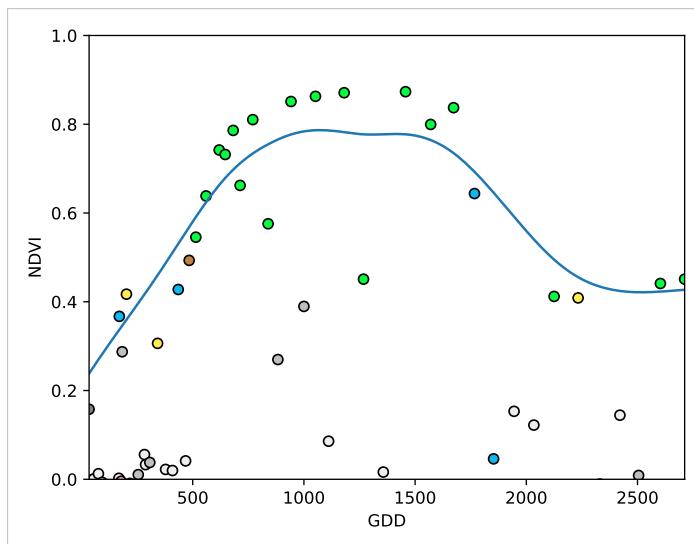


Figure 4.1: A smoothing splines fit considering green and yellow points (SCL45)

314 To get an impression whether there is some useful information contained in the remaining
315 SCL-classes (all except 4 and 5) we would like to compare the observed NDVI with the

¹The blue points correspond to the SCL-class 10: Thin cirrus clouds

316 true NDVI. But since we do not have any ground truth data, we will make the following
 317 assumption:

318 **Definition 4.1.0.1.** *XXXAssumption (true NDVI)* *The true NDVI value at time t can be*
 319 *successfully estimated by out-of-bag interpolation using high quality observations. That is*
 320 *the interpolated value (using XXX) considering the points $P^{SCL45} \setminus P_t$. In the following,*
 321 *we will call this estimate the “true”-NDVI*

322 shall pair every observed NDVI value with its out-of-bag-estimate. Then for each category
 323 we collect all pairs and create a scatter plot in fig 4.2XXXXXXXXXXXXXX

- 324 i.) For each pixel and for each observation (every SCL-class):
 325 estimate the NDVI value (via out-of-the-box interpolation²)
- 326 ii.)

327 4.2 XXX Correction

328 We recall the satellite images in figure 2.1d, where we had cloudy images despite scl4
 329 labeled and see fragments in figure 2.1e even though we are supposed to see clouds (scl 10
 330 - Cirrus clouds). The SCL classification is based only on a mixed model trained using the
 331 s2 bands.

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

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

340 4.2.1 Response and Covariates

341 For training a NDVI correction model, we need ground-truth (response) and informative
 342 covariates. We organize those in a table, where each row corresponds to a P_t (i.e., a pixel
 343 at a time t). For the response we will again use the assumption 4.1.0.1. There is no
 344 canonical answer to the question which covariates we should use. It is a tradeoff between
 345 simplicity/generalizability and performance (with the danger of overfitting). Our desire
 346 with the NDVI correction is to develop a product that is simple for others to understand
 347 and use. Therefore, in the subsequent we will only take the spectral data of the satellite
 348 and the observed NDVI derived from it as covariates³.

349 4.2.2 Correction Methods

350 In the following, we will introduce different modelling approaches, which we will use to
 351 model the relation between the response $y = y_{\text{true OOB NDVI}} \in \mathbb{R}^n$ and the covariates

²That is, we use all observations (in SCL45) but the current one.

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

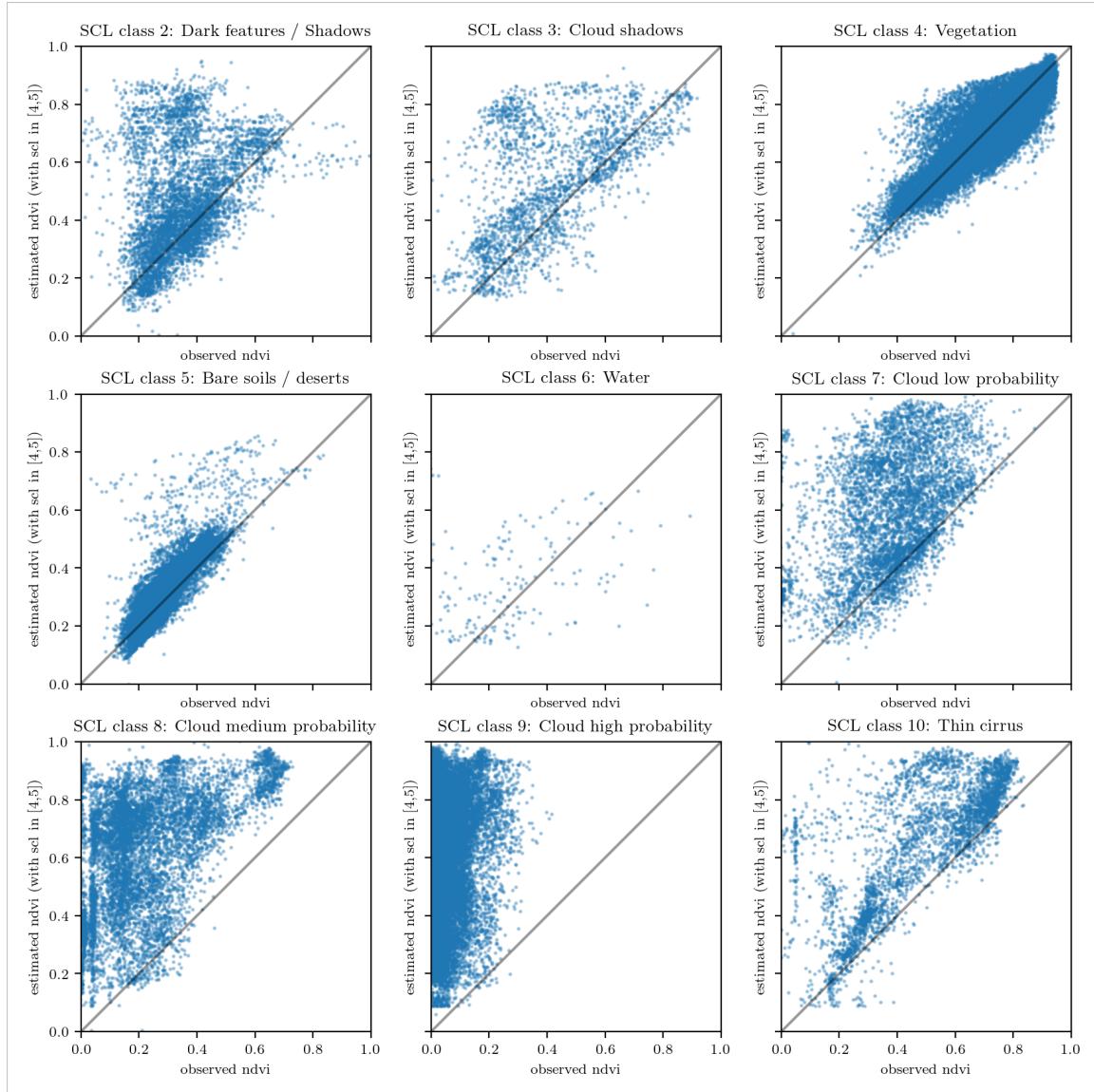


Figure 4.2: XXX caption XXX

352 encoded in the design matrix⁴ $X \in \mathbb{R}^{n \times p}$. Furthermore, we will use the matlab ‘:’ notation
 353 to indicate rows and columns of a matrix (e.g. $X(:, 3)$ is the 3rd column of X).

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

356 Ordinary Least Squares (*OLS*)

357 The OLS is a linear model which aims to minimize the sum of the squared residuals. Let
 358 $y \in \mathbb{R}^n$ be the vector of responses and $X \in \mathbb{R}^{n \times p}$ be the design matrix, where each row
 359 corresponds to one pixel and each column consist of one covariate⁵. We assume a linear

⁴This is the Matrix which contains of all covariates.

⁵Strictly speaking since SCL-classes are dummy variables

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

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

361 Assuming that X is regular, we can estimate the regression coefficients β by

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

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

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

364 LASSO

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

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

367 Even though we do not have a closed form solution for equation (4.2.2.1) we can solve
368 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
369 continious and convex.

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

372 In order to know which λ to choose we try a huge range of possible values. For each β_λ we
373 calculate the cross-validated $RMSE_\lambda$ ⁸ (and its standard deviation σ_λ using the k folds)
374 and define the λ with the smallest corresponding $RMSE_\lambda$ as λ_{min} . From here we choose
375 the largest λ for which the $RMSE_\lambda$ is smaller than $RMSE_{\lambda_{min}} + \sigma_\lambda$. This yields a simpler
376 model while keeping the $RMSE$ reasonable model.

377 We will apply the Lasso using the selected covariates in section 4.2.1 and their second
378 degree of interactions.⁹

379 Random Forest (RF)

380 To define a random Forest introduced by Breiman (Breiman) we will first define what a
381 Tree is. A (*decision*) Tree is a graph (V, E) without circles, a distinct root node, every
382 node has at most two children and every leaf has a value assigned to it. At each node there
383 is a boolean condition testing if one variable is greater than some value and a pointer to

⁷The last two terms are equivalent by lagrangian optimization

⁸The 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 $RMSE$:

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

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

384 one child depending on the boolean value. To evaluate a tree we start at the root node,
 385 test the boolean expression and go to the node indicated by the resulting pointer. This
 386 we repeat until we end up at a leaf-node where we return the value assigned to it.

387 To build such a Tree we will recursively partition the covariate space using greedy splits¹⁰
 388 decreasing the RMSE¹¹ each time. If the set we want to split contains less than a certain
 389 amount of training points we stop.

390 To build a *Random Forest* we will bootstrap-aggregate¹² many such Trees¹³. The prediction
 391 of the Random Forest for a new point x is then the mean of the predictions from all
 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.

392

393 Multivariate Adaptive Regression Splines (*MARS*)

394 REF[Friedman](#) ([Friedman](#))

395 A MARS model can be described by

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

396 where the h_m are simple functions (explained later) and the β_m are estimated via least
 397 squares.

398 In the building procedure of a MARS model we first select many of those simple functions
 399 and later drop some of them to avoid overfitting.

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

¹¹To 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).

¹²That is we will sample (with replacement) n observations from our original data and fit a Tree to this new sample.

¹³Building 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.

400 For the construction of those simple functions define \mathcal{B} be the set of pairs of ‘hockystick
401 functions’

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

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

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

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

407 Finally, to avoid overfitting we prune the set \mathcal{M} by optimizing a generalized cross validation
408 score (GCV).¹⁴

409 To reduce computational complexity, we follow the recommendation from `REFleaps wrapper`
410 `(leaps wrapper)` and restrict h in equation (4.2.2.2) to be of degree one (so it is also
411 in a pair of \mathcal{B}). Consequently, \mathcal{C} contains functions with a degree of at most 2.

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

412 General Additive Model (*GAM*)

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

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

418 To estimate the non-parametric functions we can use smoothing splines (ref sec. 3.5.6).
419 For this let \mathcal{S}_j be the function which takes some $z \in \mathbb{R}^n$ and returns the smoothing splines
420 fitted to $(X_{:,j}, z)$ where the smoothing parameter is optimized by GCV. Since we cannot fit
421 all g_j simultaneously we will use a strategy named backfitting. We basically cycle through

¹⁴This 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.

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

422 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

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

Pros	Cons
— Captures non-linearity.	— No automatic variable selection.
— Good interporeability.	— Computationally expensive.

424 4.2.3 Uncertainty Estimation

425 Once we correct the NDVI using the previous section, we are left with the problem that
 426 not every correction is equally reliable.¹⁶ Hence, we are interested in a measure of how
 427 uncertain an estimate is.

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

431 4.2.4 Interpolation

432 Consider now a pixel P , $\hat{y}^{(P)}$ its corrected NDVI and $\hat{v}^{(P)}$ the estimated uncertainties of
 433 $\hat{y}^{(P)}$. In order to interpolate $\hat{y}^{(P)}$ we will give less weight unreliable observations. Thus,
 434 we define the weightfunction:

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

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

438 4.3 Resulting Interpolation Strategies

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

- 440 i.) OOB Interpolation (+ robustify?)
- 441 ii.) Correction
- 442 iii.) Uncertainty estimation
- 443 iv.) Interpolation (+ robustify?)

¹⁶One correction is illustrated in the figure A.1f. In this figure, the outer points (labeled as clouds) have a large scatter.

- 444 At each step we have a choice, more precisely:
- 445 — Interpolation: Smoothing Splines / Double Logistic
- 446 — Robustify: Yes / No
- 447 — Correction & uncertainty estimation: RF / OLS – considering only SCL-classes /
- 448 OLS – considering all selected covariates / MARS / GAM / LASSO / no correction.
- 449 As it is not feasible to try every possible combination, we make the following restrictions
- 450 of which combinations we will consider:
- 451 — We use the same interpolation method each time.
- 452 — Either we robustify both times or we do not robustify at all.
- 453 — We use the same underlying method for correction and uncertainty estimation.
- 454 In this fashion, we obtain 28 distinct interpolation strategies, which we will benchmark in
- 455 the next section.

456 4.4 XXX Evaluation Method

- 457 In this section, we introduce the relative yield-estimation-accuracy (*RYEA*) and utilize it
- 458 to evaluate the interpolation strategies from section 4.3.
- 459 **Definition 4.4.0.1.** (*RYEA*) Let $y \in \mathbb{R}^n$ be the yield, M be a model for estimating y , and
- 460 $\hat{y} = M(X)$ where X describes the data¹⁷. We define the *RYEA* as the relative RMSE in
- 461 yield estimation. Formally expressed:

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

462

463 4.4.1 XXX Idea

- 464 yield estimation is a main goal. Claim that yield-estimation-accuracy is a objective mea-
- 465 sure : - we have not looked at the yield so far - if the one NDVI-time-series predicts the
- 466 yield better than a different one, we conclude that the first time-series carries more true
- 467 information about the plants Now: "yield NDVI-TS / derived-covariates"

468 4.4.2 Yield Estimation

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

- 474 — Maximum slope
- 475 — Minimum slope
- 476 — Integral¹⁸ over all

¹⁷We will use the matrixes derived in section 4.4.2

¹⁸ 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

- 477 — Peak (i.e. maximal NDVI)
478 — Peak GDD (i.e. value at which the peak is attained)
479 — Integral¹⁸ up to the peak
480 — Integral¹⁸ after peak
481 — Integral¹⁸ from 0-685 GDD
482 — Integral¹⁸ from 685-1075 GDD
- 483 For the choice we were inspired by REF-kamir. However, we deliberately omit any statistic
484 that involves the minimum (e.g. the NDVI-range), since we regard the minimum as very
485 error-prone (clouds) and uninformative measure.
- 486 As a result, we obtain for each interpolation strategy a matrix in which each row corre-
487 sponds to a pixel and contains both the yield and the characterizing statistics. Using this
488 matrix, we train a random forest¹⁹ for yield estimation, and compute the integrated OOB
489 estimates²⁰ \hat{y} . Finally, for each interpolation strategy, we calculate the RYEA. The results
490 are shown in table 5.1.

¹⁹The choice of the modelling approach does not matter too 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.

²⁰By 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 $\frac{1}{e}$).

491 **Chapter 5**

492 **Results**

493 **5.1 XXX small recap from “Interpolation Methods”**

494 **5.2 Robustification and NDVI-Correction**

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

495 **Chapter 6**

496 **Discussion**

497 **High RMSE in ...:** How much can we expect to get? We have multiple sources of uncer-
498 tainty in the data: 1. Uncertainty in Yield data collected by the combine harvester 2.
499 Uncertainty in Yield data through rasterization 3. Uncertainty in satellite images through
500 “measurement errors” introduced via clouds and other atmospheric effects 4. Uncertainty
501 introduced by interpolating (especially when long data-gaps are present)

502 Bootstrap in NDVI correction

503 **6.1 NDVI Correction**

504 **6.1.1 Shall We Use Additional Covariates?**

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

509 We decided against using this data, because on the one hand we have the problem that
510 we have practically too few observations (we observe only 5 years) and we expect the
511 weather in our study region to be rather homogeneous ¹. On the other hand, we want
512 the underlying model not to learn improper relationships. For example, the model might
513 automatically predict a high NDVI for a day in summer (detected by high GDD / many
514 sunshine hours / high temperature) just because it is “used” to observing a lot of vegetation
515 in summer. Including temporally (e.g., P_{t-1} and P_{t+1}) and geographically adjacent pixels
516 would likely improve performance. However, for simplicity, we omit it here².

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

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

¹The weather data are published by Meteoswiss for a grid with a resolution of 1 km

²This 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).

521 **Chapter 7**

522 **Outlook**

523 **7.1 Data**

- 524 — Method how data has been extrapolated to the grid could possibly be improved
525 — For computational reasons we mostly considered all years and split the data (on the
526 pixel level) randomly into a train/test set. A cross Validation with leaving one year
527 out would be

528 **7.2 Interpolation**

- 529 — Penalized Regressions as described in ... are similar to smoothing splines (cf. ...)
530 but different. Better?

531 **7.3 NDVI Correction**

- 532 — try different link functions in section ... between estimated absolute residuals and
533 weights

534 **7.4 NDVI Correction + +**

- 535 — NDVI Correction can be applied to all sorts of land observed via. satellites (without
536 the need of ground truth data)
537 — The idea of NDVI Correction could be applied to other spectral indices like the
538 Green Leaf Area Index.
539 — Yield is not the only target variable of interest. Other variables like protein content
540 could also be used in section ... for the method evaluation.

541

Bibliography

- 542 Gaussian models for geostatistical data. In P. J. Diggle and P. J. Ribeiro (Eds.), *Model-
543 Based Geostatistics*, pp. 46–78. Springer.
- 544 Beck, P. S. A., C. Atzberger, K. A. Høgda, B. Johansen, and A. K. Skidmore. Improved
545 monitoring of vegetation dynamics at very high latitudes: A new method using MODIS
546 NDVI. *100*(3), 321–334.
- 547 Breiman, L. Random Forests. *45*(1), 5–32.
- 548 Cao, R., Y. Chen, M. Shen, J. Chen, J. Zhou, C. Wang, and W. Yang. A simple method to
549 improve the quality of NDVI time-series data by integrating spatiotemporal information
550 with the Savitzky–Golay filter. *217*, 244–257.
- 551 Cappellari, M., R. M. McDermid, K. Alatalo, L. Blitz, M. Bois, F. Bournaud, M. Bu-
552 reau, A. F. Crocker, R. L. Davies, T. A. Davis, P. T. de Zeeuw, P.-A. Duc, E. Em-
553 sellem, S. Khochfar, D. Krajnović, H. Kuntschner, R. Morganti, T. Naab, T. Oosterloo,
554 M. Sarzi, N. Scott, P. Serra, A.-M. Weijmans, and L. M. Young. The ATLAS3D project -
555 XX. Mass-size and mass-sigma distributions of early-type galaxies: Bulge fraction drives
556 kinematics, mass-to-light ratio, molecular gas fraction and stellar initial mass function.
557 *432*, 1862–1893.
- 558 Chen, J., P. Jönsson, M. Tamura, Z. Gu, B. Matsushita, and L. Eklundh. A simple method
559 for reconstructing a high-quality NDVI time-series data set based on the Savitzky–Golay
560 filter. *91*(3), 332–344.
- 561 Cleveland, W. S. Robust Locally Weighted Regression and Smoothing Scatterplots.
562 *74*(368), 829–836.
- 563 Friedman, J. H. Multivariate Adaptive Regression Splines. *19*(1), 1–67.
- 564 Hastie, T. and R. Tibshirani. Generalized Additive Models: Some Applications. *82*(398),
565 371–386.
- 566 Jaramaz, D., V. Perović, S. Belanovic Simic, E. Saljnikov, D. Cakmak, V. Mrvić, and
567 L. Zivotic. The ESA Sentinel-2 mission Vegetation variables for Remote sensing of
568 Plant monitoring.
- 569 leaps wrapper, S. M. D. f. m. b. T. H. a. R. T. U. A. M. F. u. w. T. L. Earth: Multivariate
570 Adaptive Regression Splines.
- 571 Lyche, T. and K. Mørken. Spline Methods.
- 572 McMaster, G. S. and W. W. Wilhelm. Growing degree-days: One equation, two interpre-
573 tations. *87*(4), 291–300.

- 574 Savitzky, A. and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified
575 Least Squares Procedures. *36*(8), 1627–1639.
- 576 Schafer, R. W. What Is a Savitzky-Golay Filter? [Lecture Notes]. *28*(4), 111–117.
- 577 Tibshirani, R. Regression shrinkage and selection via the lasso: A retrospective. *73*(3),
578 273–282.

579 **Appendix A**

580 **XXX Appendix**

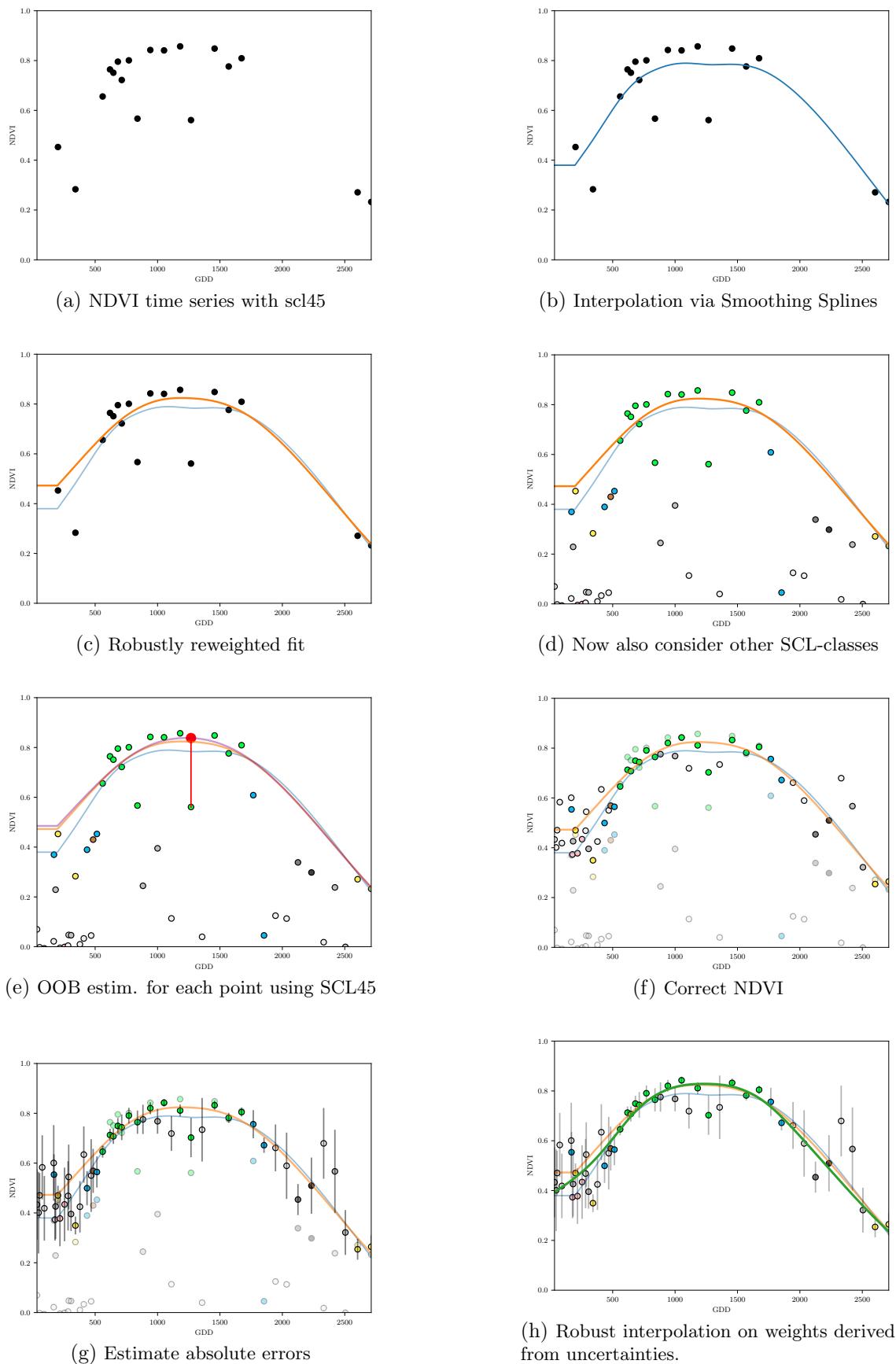


Figure A.1: Stepwise illustration of robust NDVI-Correction. For the color encoding of the SCL classes we refer to table 2.2.