ETH
Swiss Federal Institute of Technology Zurich

Seminar for Statistics

Department of Mathematics

Master Thesis Spring 2022

Lukas Graz

Interpolation and Correction of Multispectral Satellite Image Time Series

Submission Date: September 18th 2022

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

Preface

Supplementary Material

Instructions and the relevant code needed to reproduce this thesis can be found in the GitHub repository:
https://github.com/LGraz/MasterThesis-Code

To use our results we recommend the R-package:
https://github.com/LGraz/CorrectTimeSeries

More information is given in the appendix A.

Acknowledgements

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

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

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

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

Abstract

Die Kern-Resultate müssen auch in den Abstract. Ebenso würde ich die vollständige Reproduzierbarkeit und die R-Package erwähnen.

Kurze problemerläuterung (NDVI-ts im Zentrum)

NDVI Interpolation gewinner

erforscht Robusification

NDVI Correction + yield-based evaluation

Contents

Todo list

Notations

Variables

c	a (vector of) constant(s)
$\lambda \in \mathbb{R}$	a scalar
$n \in \mathbb{N}$	sample size
i, j	are indices in $\{1, \ldots, n\}$
$n \in \mathbb{R}^n$	time, usually in GDD
$w \in \mathbb{R}^n$	a vector of weights for each location x
$y \in \mathbb{R}^n$	response in 1-dim interpolation setting
$\hat{y} \in \mathbb{R}^n$	estimate of y
$\bar{y} \in \mathbb{R}$	sample mean of y
$r \in \mathbb{R}^n$	residuals given by $y - \hat{y}$

Abbreviations and Objects

Pixel	A pixel originates of an image pixel and describes a square of 10 x 10 meters in the field which coincides with the resolution (and location) of the Sentinel-2 pixels. Such pixels are illustrated in figure 2.1b. Additional information like yield is also attached.
P_t	describes the observed data (weather and spectral bands) at time t and the location of one pixel.
P	is a pixel. We see it as a collection of all the observations at the specified location within one season. More formally, $P := \{P_t \mid t \text{ is a valid sample time within a defined season}\}$
SCL	Scene Classification Layer provided by the European Space Agency (ESA) that gives an estimation of the land cover class of each pixel. It indicates what one can expect at a pixel at a sampled time. For an overview, c.f. table 2.2
P^{SCL45}	is 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.
NDVI	Normalized Difference Vegetation Index (Rouse, 1974)
DAS	Days After Sowing

GDD	Growing Degree Days – cumulative sum of "$\max(0, \text{temperature} - \text{threshold})$"
RYEA	Relative Yield-Estimation-Accuracy. Definition 4.6.0.1
OOB	Out Of the Box. Describes the procedure of estimating the value for a point but not consider the point itself (c.f. section 2.7.2)

XXX ML models and their shortnames

European Space Agency (ESA)

MATLAB Matrix Notation

We will use the MATLAB '`:`' notation to indicate rows and columns of a matrix. That is 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 X.

XXX only equations that are referenced are equipped with a number

Chapter 1

Introduction

Satelite image time series are used in ... The European Space Agency makes the images from the Sentinel 2 satelites freely avialable Extracting indicies time series (like NDVI) and used to model ... (not only of interest to researchers but also public agents and insurance companies) - Plant Models REF - Season Start (start of spring) (community name: land-surface-plant-phenology) - Yield prediction - crop classification erronous observations –> convervative (SCL) filtration –> Data gaps currently done: interpolation and smoothing techniques we give an overview + review over popular interpolation methods + discuss how data gaps influence the given methods + discuss approach of robustifing against outliers Select suitable ones in our NDVI setting –> benchmark Try to eliminate data gaps by not using strong SCL-filtration but weighting. Weighting comes from an uncertainty estimation done by a statistical model (develope a proxy for the true NDVI) <– we tried various <– evalute different IS's with objective defined quality measure (which relies on the assumption that a NDVI TS which better models the plant growth is more suitable for predicting yield)

Research Questions

i.) 1 review of interpolation methods

ii.) 2 errornous observations — how to deal with them

iii.) 3 data gaps — influence itpl mehtods

iv.) 4 data gaps — how to deal with them

v.) 6 how to compare two NDVI interpolations?

1 in 3 2 robustification 3.5 3 discussed in 6.1.1 4 utilize observations filterd before and estimating how reliable each of them are 4 5

"Similarly, smoothing the time series of satellite data is helpful to address inconsistency in observation frequency and timing due to clouds and other sensor artefacts Skakun, Vermote, Franch, Roger, Kussul, Ju, and Masek (2019)"

Why do we do interpolation in NDVI (and other indices) time series? What are possible shortcomings thereof?

— Doublelogistic (winter-ndvi)

— parametric / non-parametric approaches

182 — spatio-temporal approaches

Chapter 2

Data and Methods

We will start by describing the available data and the challenges associated with it. Our study region is a farm of over 800ha, which is located in western Switzerland. From Perich, Turkoglu, Graf, Wegner, Aasen, Walter, and Liebisch (2022) we acquire satellite image data (section 2.1), yield maps of several cereals from 2017 to 2021 (section 2.2), and meteorological data (section 2.5). Afterwards, we will introduce general methods in section 2.7, which will be used in the remaining chapters.

2.1 Sentinel 2 Data

The European Space Agency (ESA)[1] freely distributes the high-quality images of the two Sentinel satellites (S2). Together, both satellites have a revisit time of 5 days at the Equator and 2-3 days at mid-latitudes. However, in our study region, we only receive an image every 5 days.

The S2 images contain 12 spectral bands with spatial resolutions up to 10 meters (see 2.1). Bands with a lower resolution (20 and 60 meters) were upscaled to 10 meter resolution using cubic interpolation (Perich et al. (2022)). In order to decrease the effect of atmospheric conditions like reflections and scattering, bottom-of-atmosphere, radiometric corrected Level-2A data was used[2]. The ESA also supplies an algorithm[3] produces Scene Classification Layer (*SCL*) where for each location the observed subject is assigned to one of 11 SCL-classes (c.f. table 2.2). In this thesis, we will use this classification to filter out data points, which we belive to be less informative. That are all observations which SCL-class does not correspond to vegetation or bare soils (classes 4 and 5). For convenience, we define the set SCL45 as the observations which belong to SCL-class 4 or 5.

2.2 Crop Yield Data

The crop yield data were collected using a combine harvester. Equipped with GPS, the harvester drives over the fields and continuously estimates the dry crop yield density in

[1] https://sentinel.esa.int/web/sentinel/missions/sentinel-2

[2] According to Perich et al. (2022): "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"

[3] https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm

Table 2.1: 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 (Jaramaz et al. (2013)).

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.

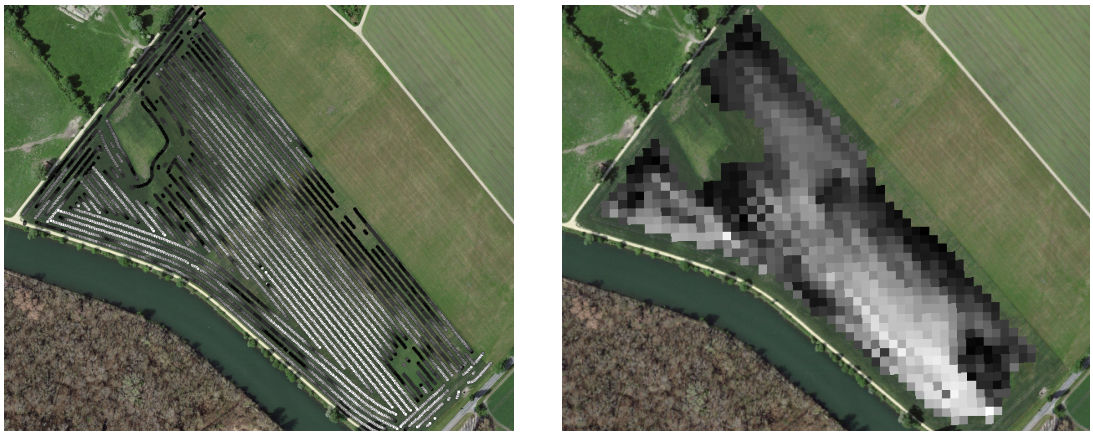
Table 2.2: Overview: Scene Classification Layers (SCL)

Color	No.	Class	Color	No.	Class
	0:	Missing Data		6:	Water
	1:	Saturated or defective pixel		7:	Cloud low probability
	2:	Dark features / Shadows		8:	Cloud medium probability
	3:	Cloud shadows		9:	Cloud high probability
	4:	Vegetation		10:	Thin cirrus cloud
	5:	Bare soils		11:	Snow or ice

t/ha (see fig. 2.1a). We take the data set derived in Perich et al. (2022), where error-prone measurement points (such as during a tight curve of the combine harvester) were removed and then the yield map was rasterized using linear interpolation (c.f. fig. 2.1b). We summarize the rasterized dry-yield values by the following statistics:

Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum	Variance
0.107	6.186	7.560	7.359	8.756	13.35	4.035

Comparing the average per-field crop yield reported by the farmer with the yield estimated by the combine harvester shows that the latter overestimates crop yield by ca. 10% (c.f. Perich et al. (2022)). Since the relative estimation error is approximately constant and we do not aim for an accurate yield prediction, we will not consider this deviation.



(a) Raw combine harvester data (cleaned) (b) rasterized to Sentinel 2 resolution.

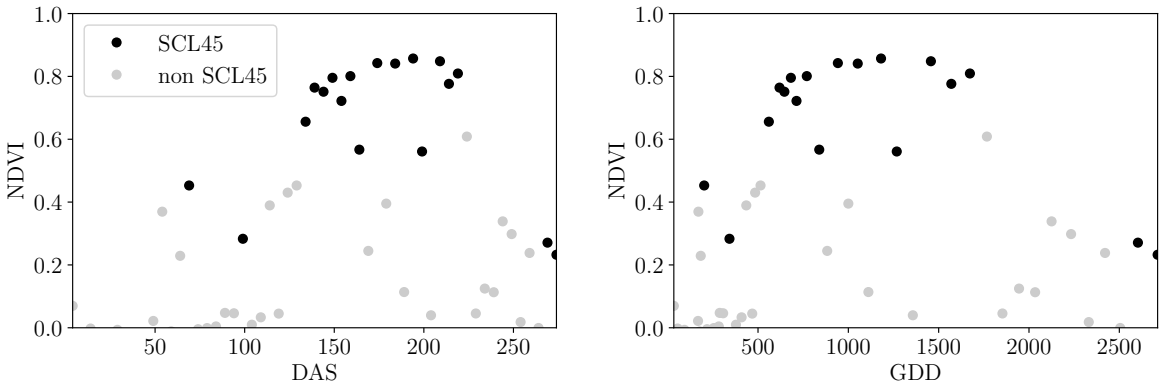
Figure 2.1: Crop yield density map of a field. Ranges from 0.1 t/ha (black) to 5.35 t/ha (white)

2.3 Normalized Difference Vegetation Index (NDVI)

The well-known (*NDVI*) introduced in Rouse (1974) is used to measure vegetation in remote sensing. It utilizes a large jump of reflectancy between red and infrared and can be calculated using the bands $B4$ and $B8$ (table 2.1) by:

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

Since we measure the NDVI via the S2 satelites from space we can not expect to measure the true NDVI. This is especially true if we do not see the ground because of clouds or the ground signal is disturbed by cloud shadows. Even if we only use SCL45 observations we still encouter issues as will be described in section 2.6. Therefore, we call the calculated values merely the *observed NDVI*. In the following chapters, we will study the resulting NDVI time series (for one location and one season) extensively. Such a time series is shown in figure 2.2a.



(a) Days After Sowing (DAS) (b) Growing Degree Days (GDD)

Figure 2.2: NDVI time series plotted against DAS and GDD. GDD are introduced in section 2.4.

2.4 Timescale Transformation

Regarding the Days After Sowing (DAS) time scale shown in fig. 2.2a, we detect two drawbacks. First, this scale makes it difficult to compare two NDVI time series because wheat is not always sown on the same day of the year and in some years plants begin to emerge earlier. Second, because there are only few SCL45 observations in the winter, we face significant data gaps in this period. The time scale transformation introduced in McMaster and Wilhelm (1997) fixes both problems. The resulting Growing Degree Days (GDD) are defined as the cumulative sum since sowing of temperature above a given base temperature T_{base}. For cereals, we use $T_{base} = 0$ (Perich et al. (2022)). Thus, the GGD for n days after sowing will be equal to:

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

Important plant growth stages and their corresponding GDD values are tabultaed in B.1.1

In figure 2.2 we see an example for comparison of the DAS and GDD timescale. Here we see that the first 120 DAS are compressed to just 500 GDD and hence the gap in observations was succesfully compressed. Due to the reasons mentioned above, from now on we will only consider GDD.

2.5 The Concept of a 'Pixel'

Now we create a new data structure that we call Pixel. This originates from the pixels of the S2 satellite images. It will contain all the information needed to confront the tasks in the following chapters.

Consider a 10 by 10 meter square that coinsides with a S2 image pixel and T the GDD values for which S2 images are avialable in a given season. For $t \in T$ let P_t be a tupel of all the spectral bands, the observed NDVI and the SCL class (at the considered location at time t). Then, define P as the collection of all the P_t and the estimated dry-yield for this square. Analogously to P, define P^{SCL45} by only considering P_t with SCL-class 4 or 5 (vegetation and soil).

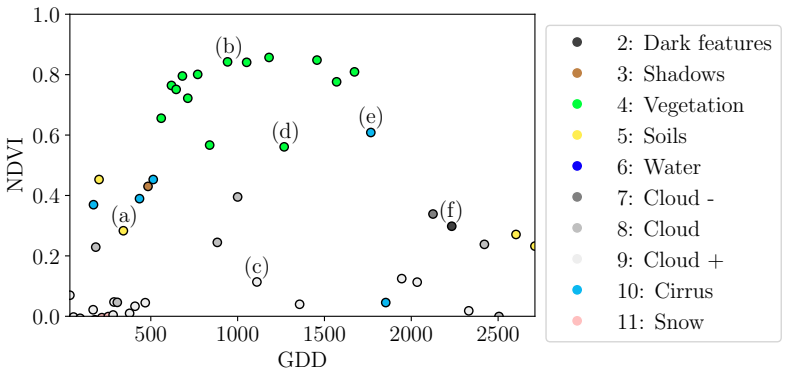
2.6 Challenges in S2 Data

Now, we shall illustrate with an example pixel the challenges, we will confront in the coming chapters. The figure 2.3 shows a selection of 6 satellite images of a field, one selected Pixel and the NDVI time series of that pixel. In February (image a), we see no vegetation but bare soil and thus also a low NDVI. At the beginning of May (b), we observe a cloudless dark green field with a high NDVI. In (c) heavy cloud cover (SCL class 9) leads to a complete loss of plant information in this S2 observation. Figure (d) shows that the SCL classification is not reliable, since we evidently observe clouds which is also reflected in a sudden NDVI drop. Even though SCL indicates that (e) are thin cirrus clouds, we see a pale green and we also note a NDVI.

So in conclusion, we remark that some SCL45 observations are not accurate and even though a few non-SCL45 observations contain useful information, most of them are too unreliable (e.g. all SCL 9 observations). Thus, we aim to substitute the unreliable ones with interpolated versions and correct corrupt ones.

(a) 2021-02-23 scl5 (b) 2021-05-09 scl4 (c) 2021-05-24 scl9

(d) 2021-06-03 scl4 (e) 2021-06-28 scl10 (f) 2021-07-23 scl2



(g) Corresponding NDVI time series

Figure 2.3: Satellite images of a field at selected times with a static background for orientation. Moreover, the NDVI time series of the red-highlighted pixel is shown in (g) colored by the SCL labels.

2.7 General Methods

Here we will only introduce Methods which will accure in several places. For interpolation methods we refer to sections 3.2 and 3.3, for a robust interpolation strategy to section 3.5. In section 3.4 we describe a method to objectively determine the quality of an interpolation, and in chapter 4 we present the NDVI correction together with an adapted interpolation strategy.

2.7.1 Root Mean Square Error (RMSE)

In this section we describe different criteria to evaluate models. Hence, given a vector $y \in \mathbb{R}^n$ and its estimator $\hat{y}$ (estimated using the model), we define the RMSE as:

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

2.7.2 Out-Of-Bag (OOB) and Leave-One-Out-Cross-Validation ($LOOCV$)

The rationale for OOB and LOOCV is that we intend to evaluate a model M with unseen data. That is, if D describes the entire dataset and we train a model on a subset of D, we can use the remaining data to evaluate the model.

To formally introduce this, let:

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

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 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 obtain $\hat{y} := (\hat{y}_1, \dots, \hat{y}_n)$ the OOB estimator for $y \in \mathbb{R}^n$.

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

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 the corresponding $\hat{y}_i$ also the LOOCV estimator.

If we optimize some parameter via OOB (or LOOCV) this means that we search for the parameter that minimizes some loss function which takes the OOB (or LOOCV) residuals. Usually we approximate this parameter by searching on a grid.

Chapter 3

Interpolation Methods

verdeutliche dem leser, dass ein auftrag das findne von interpolation-methoden war

In section 2.6 we have established the need for interpolating the NDVI time series. In this chapter we first specify a setting for the interpolation and divide the interpolation methods into those that make fundamental shape assumptions (parametric) and those that are more flexible (non-parametric). We give an introduction for each method with an compact definition, highlight adjustments or give remarks where appropriate, and then point out strengths and weaknesses of each method. Additionally, a brief overview of the considered interpolation methods is provided in table 3.1. Afterwards, we extract an robustification strategy from the one interpolation method and generalize it so we can use it for all methods that allow for a priori weighted observations. Finally, using LOOCV, we tune the parameters (where necessary) and get a first idea of the performance of each method.

3.1 Interpolation Setup

In this chapter, we will only consider SCL45 observations, since they are more reliably. Hence, data in the form of (t_i, y_i) for $i = 1, \dots, n$ is given, where t_i is the time in GDD and y_i denotes the NDVI at time t_i. Assume that it can be represented by

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

where ε_i is some noise and $m : \mathbb{R} \to \mathbb{R}$ 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

$$m(t) = \mathbb{E}[y \mid t]$$

We will introduce parametric and non-parametric approaches to estimate m in section 3.2 and 3.3 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 (t_i, y_i) should have in the interpolation.

Paper zitieren wo eingeführt oder wo benutzt (falls einführung fast schon trivial)

3.2 Parametric Regression

Parametric Curve estimation tries to fit a parametric function, such as, for example, a Gaussian function with parameters μ and σ, to a dataset. In the following, we introduce two parametric approaches.

Table 3.1: Summary of the studied interpolation methods containing important assumptions, advantages and disadvantages and whether the method supports weighted observations (w) and if the resulting interpolation is bounded w.r.t. a fixed interval (b).

	Assumptions	**Advantages**	**Disadvantages**	**w**	**b**
Double-Logistic	– Function first increases then decreases – NDVI has a minimal value	– Good for evergreen plants (if snow masks NDVI) – Upper envelope	– Parameter estimation can be very difficult – Strange behavior for long data-gaps	Yes	(Yes)
Fourier Approximation	– NDVI can be approximated by a 2cd order Fourier series.	– Incorporates periodical growth-cycles	– Parameter estimation can be very difficult – Curve easily exceeds bounds of the NDVI	Yes	No
(Gaussian) Kernel Smoothing	– Close points are related to each other via a kernel function	– Simple – Computationally very fast	– Biased, especially at 'peaks' and 'valleys' – Bandwidth: fails if there are big data-gaps	Yes	Yes
Universal Kriging	– Function is a realization of a stationary Gaussian process	– Informative parameters – Flexible	– Regression to the mean – Assumptions clearly not met	Yes	(Yes)
SG Filter	– High frequencies are noise (Low-Pass-Filter) – Equidistant points – Local polynomials	– Computationally very fast	– Cannot deal natively with missing data (need some interpolation)	No	(Yes)
SG + NDVI	– Upper envelope – Vegetation cannot grow faster than some slope	– Biological knowledge	– Bad "upper envelope" since weights are not used for the estimation itself	(No)	(Yes)
LOESS	– Local polynomial with points closer to the estimated point are more important	– Flexible – Generalization of SG – Weighting function makes intuitive sense	– Computationally expensive	Yes	(Yes)
B-Splines (Smoothed)	– Function can be approximated by a linear combination of B-splines basis functions	– General assumption – Flexible shape	– Unbounded – No intuitive meaning for smoothing	Yes	No
Smoothing Splines	– 2cd derivative of function is integrable	– Intuitive meaning of penalty – General assumptions – Flexible shape	– Choice of smoothing parameter	Yes	No

3.2.1 Double Logistic

The Double Logistic smoothing as described in Beck, Atzberger, Høgda, Johansen, and Skidmore (2006)REF heavily relies on shape assumptions of the fitted curve (i.e. the NDVI time series). First, we assume that 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 several years into account. Second, we assume that 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 3.1.

Robustification

Similar as for the Savitzky-Golay Filter (c.f. section **??**) one can reestimate (only once) the parameters by giving less weight to the overestimated observations and more weight to the underestimated observations. For the details on the choice of the weights we refer to Beck et al. (2006). We will not apply this reestimation but rather the robustification introduced later in section 3.5.

Advantages	Disadvantages
— Incorporates subject specific knowledge in the case of evergreen plants covered in snow.	— Strong shape assumptions on the NDVI curve.
— Optimized parameters have an intuitive meaning.	— 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. (c.f. figure 3.1)

3.2.2 Fourier Approximation

Analogous to section 3.2.1 we fit a parametric curve to the data by least squares. Here we take the second order Fourier series:

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

where $\Phi = 2\pi \times (t-1)/n$. Thus, we periodical behavior. If we would set the period to match one year this would coinced with the nothion that plans grow every year. Example fits can be seen in figure 3.1

Advantages	Disadvantages
— Assumption of periodicity can be helpful if we are modelling multiyear grow cycles	— Bad behavior in regions with little data (c.f. figure 3.1)
— Flexible curve shape	— Hard to interpret estimated parameters
	— Parameter estimation can go wrong. Introducing bounds can help.

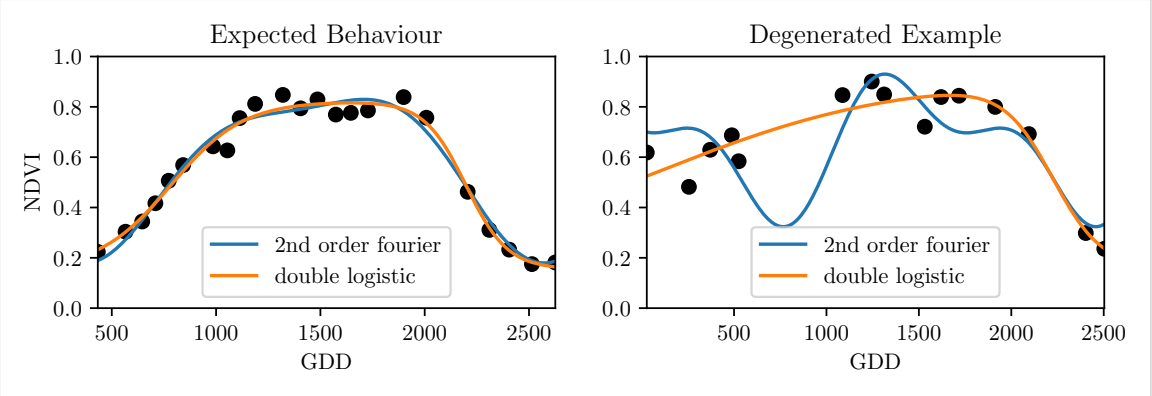


Figure 3.1: Here we observe the possibilities of a precise fit for the two parametric methods but notice also some misbehavior

3.2.3 Optimization Issues

We shall mention some optimization issues we countered during implementation. Since we aim to minimize the residual sum of squares over 5 (or 6) parameters, we try to solve a non-convex optimization problem. Thus, the algorithm[1] 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 Non-Parametric Regression

In non-parametric curve estimation, the curve does no longer have to be fully determined by parameters, but we allow it to flexibly approximate the data. Note, that we do not exclude the use of tuning-parameters.

3.3.1 Kernel Regression

As described in section 3.1, we aim to estimate

$$\mathbb{E}[Y \mid T = t] = \int_{\mathbb{R}} y f_{Y|T}(y \mid t) dy = \frac{\int_{\mathbb{R}} y f_{T,Y}(t, y) dy}{f_T(t)}, \tag{3.3.1.1}$$

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

$$\hat{f}_T(t) = \frac{\sum_{i=1}^n K\left(\frac{t-t_i}{h}\right)}{nh}, \quad \hat{f}_{T,Y}(t, y) = \frac{\sum_{i=1}^n K\left(\frac{t-t_i}{h}\right) K\left(\frac{y-Y_i}{h}\right)}{nh^2},$$

[1] We used the python function `scipy.optimize.curve_fit`.

where h, the bandwidth, symbolizes the windowsize of to consider. By using the above function in equation (3.3.1.1) we arrive at the *Nadaraya-Watson* kernel estimator:

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

Common choices for the kernel are the normal function or a uniform function (also called 'bot' function).

Choose Bandwidth

Note that we still need to choose the bandwidth of the function. This can be done with the help of LOOCV while optimizing the RMSE. For non-equidistant data we refere to Brockmann, Gasser, and Herrmann (1993) where a local adaptive bandwidth selection is presented.

Advantages	Disadvantages
— fletible due to different possible kernels	— if the $t \mapsto K(t)$ is not continuous, $\hat{m}$ isn't either
— can be assigned degrees of freedom (trace of the hat-matrit)	— choice of bandwidth, especially if t_i are not equidistant.
— estimation of the noise variance $\hat{\sigma}_{\varepsilon}^2$ (REF c.f. CompStat 3.2.2)	

3.3.2 Kriging

Kriging as described in Diggle and Ribeiro (2007) was developed in geostatistics to deal with autocorrelation of the response variable at locations which are spatially close. By applying the notion that two spectral indices which are timewise close should also take similar values, we justify the application of Kriging. In the end, we would like to fit a smooth Gaussian process to the data.

A Gaussian Process $\{S(t) : t \in \mathbb{R}\}$ is a stochastic process if $(S(t_1), \ldots, S(t_k))$ has a multivariate Gaussian distribution for every collection of times $t_1, \ldots, t_k$. S can be fully characterized by the mean $\mu(t) := E[S(t)]$ and its covariance function $\gamma\left(t, t'\right) := \mathrm{Cov}\left(S(t), S\left(t'\right)\right)$. Furthermore, we will assume the Gaussian process to be stationary. That is for $\mu(t)$ to be constant in t and $\gamma(t, t')$ to depend only on $h = t - t'$. Thus, we will write in the following only $\gamma(h)$.[2]

Now, we need to make some assumption on the covariance function. For this we introduce the variogram of a Gaussian process as

$$V(h) := V\left(t, t + h\right) := \frac{1}{2}\mathrm{Var}\left(S(t) - S(t + h)\right) = \gamma(0) + \gamma(t)$$

and define γ via the above equation by choosing the Gaussian Variogram defined by

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

[2]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.

Here h denotes the distance, n is the nugget, r is the range and p is the partial sill. The influence of the parameters is visualized in figure 3.2.[3]

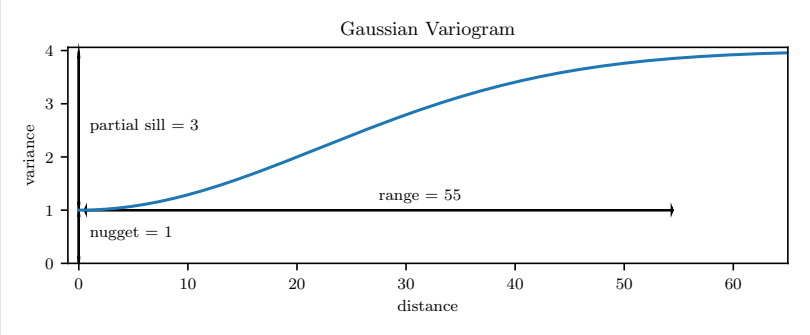


Figure 3.2: Gaussian Variogram with nugget=1, partial sill=3, range=55

Finally, we consider a one-dimensional Gaussian process G_γ with variogram γ and tune the variogram parameters using matimum likelihood[4]. Let z be a vector with the new values to ettrapolate, then we can determine the values $m(z) = \mathbb{E}\left[G_\gamma(z)|(t,y)\right]$ using Bayes rule[5]. For an etample fit, we refer to figure 3.3.

Violated Assumption

Since we observe a clear pattern of a growth period in spring and harvest in the end of summer, we have to admit that our stationarity assumtpion with the constant mean is structurally violated. This is also the reason why we observe (for every variogram parameter) a tendency to the mean, as indicated in figure 3.3.

Advantages	Disadvantages
— It is a well-studied method.	— Regression to the mean.
— Variogram parameters have an intuitive meaning.	— Violated assumption of constant mean and constant variance. Thus, the NDVI is not a stationary process.
— Flexible covariance structure.	— Pure maximum likelihood can result in overfitting.

3.3.3 Savitzky-Golay Filter (SG Filter)

The *Savitzky-Golay Filter*, introduced in Savitzky and Golay (1964) is a technique in signal processing and can be used to filter out high frequencies (low-pass filter) (Schafer, 2011). Furthermore, it can also 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, $j \in \{m, m+1, \ldots, n-m\}$ we fit

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

[4] As illustrated in figure 3.3 matimum likelihood estimation can lead to overfitting. Thus, we will in practice sample several such optimized parameters and use their median in the end.

[5] Bayes rule generally claims, that for two random variables A and B we have that $P(A|B) = P(B|A)/P(B)$

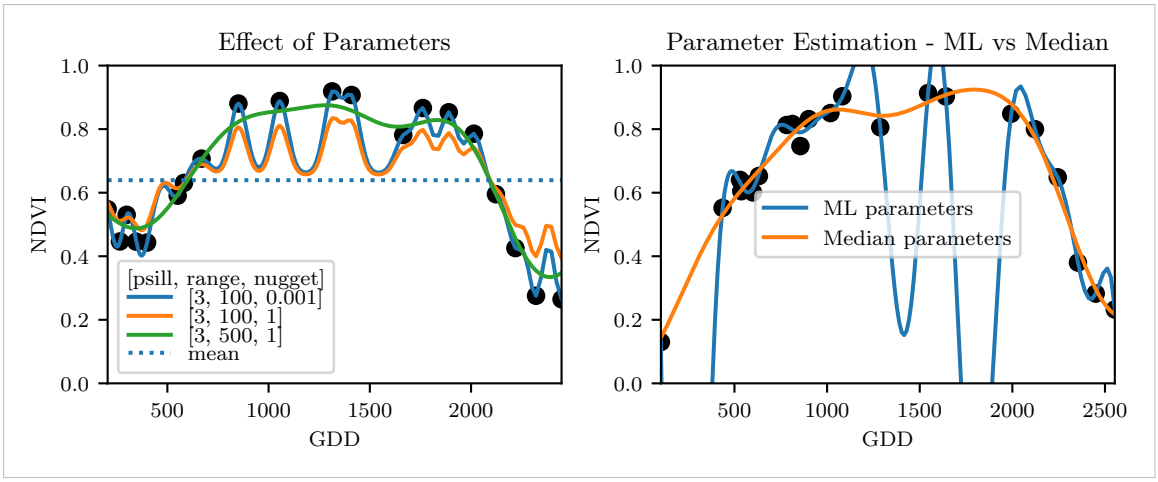


Figure 3.3: On the left, we see how the interpolation change if we increase the nugget and the range parameter. On the right, we compare two kriging interpolations, where one takes parameters by numerically matimizing the (which results in a very small nugget) and the other takes the median of many such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

a polynomial of degree k by:

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

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.

Chen, Jönsson, Tamura, Gu, Matsushita, and Eklundh (2004) developed a 'robust' interpolation method for the NDVI based on the SG Filter. 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 quickly. The latter is argued by the biological impossibility of such fast vegetation changes. Their proposed algorithm is:

figure / tabelle / pseudocode anstatt aufzählung

i.) Remove non-SCL45 points.

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 SG Filter to obtain a new time series X^1.

v.) Update X^1 by applying again a SG Filter. Repeat this until $w^T|X^1 - X^0|$ stops decreasing, where w is a weight vector with $w_i = \min\left(1, 1 - \frac{X_i^1 - X_i^0}{\max_i \|X_i^1 - X_i^0\|}\right)$. This reduces the penalty introduced by outliers[6] and by repeating this step we approach the "upper NDVI envelope".

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

Extension: Spatial-Temporal-Savitzky-Golay Filter

One notable adaptation of the SG Filter is the presented by Cao, Chen, Shen, Chen, Zhou, Wang, and Yang (2018). The key difference is the additional assumption of the cloud cover being discontinuous and that we can improve by looking at adjacent pixels[7]. Because we are working with rather high resolution satellite data, and we need the variance in the predictors, we will waive this extension.

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

3.3.4 Locally Weighted Regression (LOESS)

The LOESS introduced by Cleveland (1979) can be understood as a generalization of the SG Filter (c.f. sec. 3.3.3).

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

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

where h_i is the minimal distance such that $\lceil \alpha n \rceil$ observations are in the ball $B_{h_i}(t_i)$.[8] So for each y_i we only consider a proportion α of the observations.

Differences between the Robust LOESS and the SG Filter?

The LOESS smoother takes a fraction of points instead of a fixed number and therefore automatically adapts to the size of the data we wish to interpolate. However, we run into the danger of considering too little observations, since the estimation breaks down if $\lceil \alpha n \rceil < d + 1$.[8] Furthermore, LOESS gives less weight to points further away. This yields a "smoother" estimate, since when we slide the window (e.g. for estimating the next value) an influential point at the border does not suddenly get zero weight from being weighted equally before. Finally, the LOESS also can be used for non-equidistant data and allows for arbitrary interpolation.

[7] 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

[8] If too many weights are set to zero, we might end up considering not enough observations and thus get a singular design-matrit (for the least squares estimation). Therefore, we substitute h_i with $1.01h_i$, so that the observation on the boundary of $B_{h_i}(t_i)$ does not get completely ignored. But we also have to assure that α is big enough.

Advantages	Disadvantages
— Flexible generalization of SG Filter — arbitrary interpolation possible — Intuitive parameters	— The nature of local regression might lead to surprising estimates (no smoothness guarantees for the second derivative)

3.3.5 B-splines

B-splines as discussed in Lyche and Morken (2005) are piecewise cubic polynomials defined by

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

where B are basis functions and recursively defined by:

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

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

Assuming that all t_i are distinct, this yields an 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 basis functions[9] such that:

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

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

3.3.6 Natural Smoothing Splines

Let $\mathcal{F}$ be the Sobolev space (the space of functions of which the second derivative is integrable). Then the unique[10] minimizer

$$\hat{m} := \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} w_i \left(y_i - f\left(t_i\right)\right)^2 + \lambda \int f''(t)^2 dt \tag{3.3.6.1}$$

is a natural[11] cubic spline (i.e. a piecewise cubic polynomial function). The objective function ensures that we decrease the curvature while keeping the RMSE low.

[9] So we do not require one basis function for each neighboring pair of knots. SciPy uses FITPACK and DFITPACK, the documentation suggests that smoothness is achieved by reducing the number of knots used

[10] Strictly speaking it is only unique for $\lambda > 0$

[11] It is called natural since it is affine outside the data range ($\forall t \notin [t_1, t_n] : \hat{m}''(t) = 0$)

Advantages	Disadvantages
— Can be assigned degrees of freedom (trace of the hat-matrix). — Efficient estimation (closed form solution). — Intuitive penalty (we don't want the function to be too "wobbly" — change slopes). — Also performs well if points are not equidistant. — Fixes the Runge's phenomenon (fluctuation of high degree polynomial interpolation).	— The tuning parameter λ must be chosen. This can be done via cross validation and optimizing a score function (e.g. the RMSE).

3.4 Tuning Parameter Estimation

Many of the interpolation methods introduced in section 3.2 and 3.3 include a free parameter. To determine this parameter for a specific interpolation method, we will estimate the absolute residuals using OOB estimation and then optimize the parameter using a score function. We clarify the procedure step by step:

i.) Construct a set Λ of candidate parameters that generously covers the parameter space.

ii.) Consider $\mathcal{P}$, a set of Pixels.

iii.) For each parameter $\lambda \in \Lambda$ consider the individual pixels and compute the LOOCV[12] for the absolute residuals of the specific NDVI interpolation method for all Pixels in $\mathcal{P}$ and store them in the set R_λ.

iv.) Determine $\lambda_{optimal} = \arg\min_{\lambda \in \lambda} q_{90}(R_\lambda)$, where we describe the 90% quantile with q_{90}.

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

Figure 3.4 exemplifies the effect of the optimization function (different quantiles). To summarize, we may say that the higher the quantile, the stronger the smoothing.

3.5 Robustification

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

Even though the procedure is taken from the robust version of the LOESS smoother (c.f. section 3.3.4 and Cleveland (1979)), we can apply it to every interpolation method that allows for prior weighting of observations.

[12] For a definition of the leave-one-out-cross-validation we refer to section 2.7.2

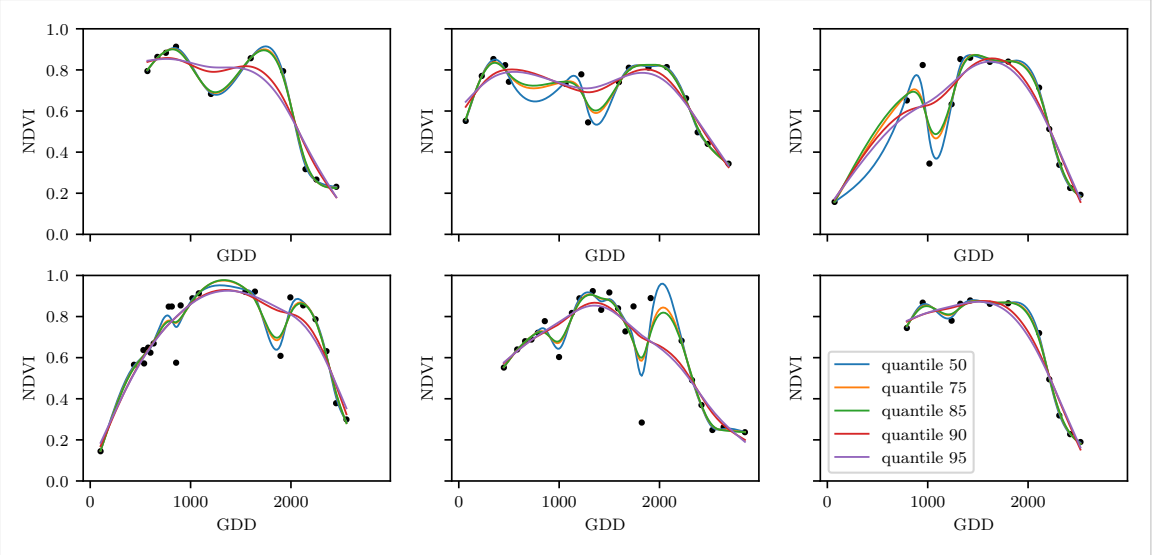


Figure 3.4: Smoothing splines fit with smoothing parameter optimized by minimizing the given quantile of the absolute leave-one-out residuals. Note that the larger the considered quantile is, the smoother the resulting curve becomes.

After an initial fit we calculate the residuals $r_i := y_i - \hat{y}_i$ and obtain $\tilde{r}_i$ by scaling with the median of the absolute residuals:

$$\tilde{r}_i := \frac{r_i}{6\,\mathrm{med}\,(|r_1|, \ldots, |r_n|)}$$

Next, we compute new weights by

$$w_i^{\text{new}} := w_i^{\text{old}} \begin{cases} \left(1 - \tilde{r}_i^2\right)^2, & \text{if } |\tilde{r}_i| < 1 \\ 0, & \text{else} \end{cases}; \tag{3.5.0.1}$$

Using the new weights, we can re-interpolate. This reweighting can be iterated for several steps or till the change of the values is smaller than some tolerance.

Note that this procedure is indeed robust since we use the median for the normalization which has a breakdown point[13] of 50%.[14]

3.5.1 Our Adjustment:

During the iterations or when supplying prior weights, low-weighted observations can corrupt our estimation of scale (the median of absolute residuals). Thus, we introduce the weighted median as

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

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

[13] Intuitively, the breakdown point denotes the fraction of observations a "vicious" player can replace without breaking the estimator. For example, the median has a breakdown point of 50%.

[14] 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).

3.5.2 Examples and Conclusions

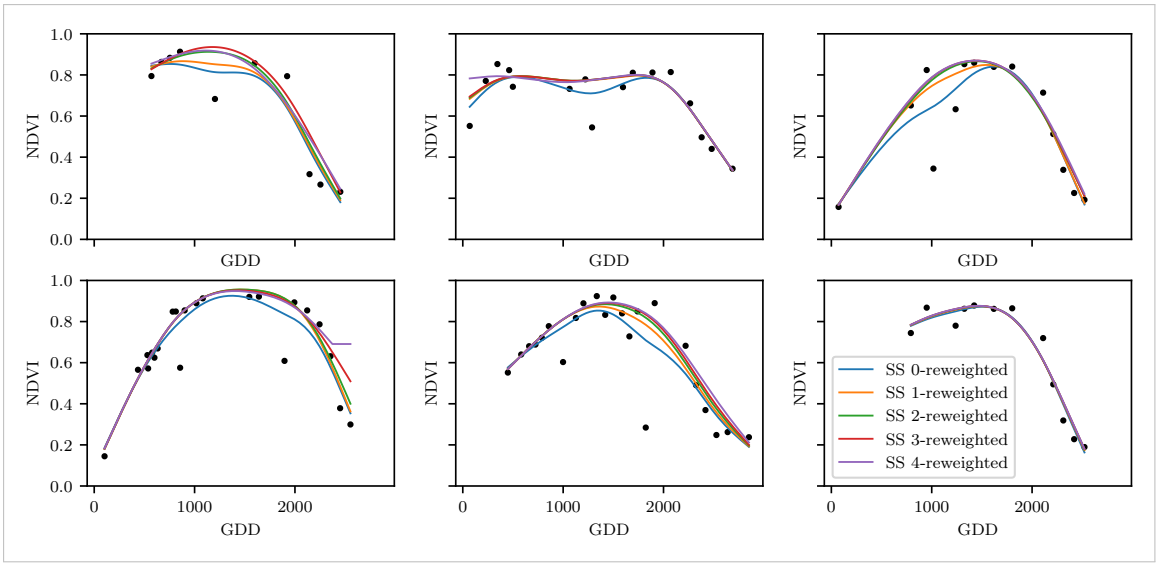


Figure 3.5: Smoothing Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.5) are also displayed

Examples of the first four iterative fits using smoothing splines are shown in figure 3.5 for six pixels. For the analogous figures of the other interpolation methods c.f. figures B.1, B.2, B.3 and B.1. Indeed, we observe how the interpolated time series is less affected by outliers after each iteration. We notice the biggest difference in the first iteration. Furthermore, in the plot at the bottom left we see how the interpolation 'escapes' from the right endpoint with each successive iteration, even though our intuition does not necessarily identify this point as an outlier. Therefore, in the following, we will always stop after one iteration.

consider naming the sub-plots

3.5.3 Upper Envelope Approach - Penalty for Negative Residuals

If we artificially increase the negative residuals in 3.5.0.1 by multiplying (e.g. factor 2), the corresponding points will get less weight in the next iteration. This allows us to create an interpolation that resembles an upper envelope. Intuitively, this upper envelope can be thought of as a sheet that is laid on top of the points.

This approach is based on the premise that we tend to underestimate the NDVI (as argued in Cao et al. (2018)). Since we want to develop a general method that is in principle not related to the NDVI, we will not pursue this approach further.

3.6 Performance Assessment

Next, we will benchmark the in section 6.1.2 preselected interpolation methods with and without robustification. For this, we will use the same technique as we did for the parameter determination in section 3.4. On B_λ we apply the RMSE and different quantiles.

The results are presented in section 5.1 and are discussed in section 6.1. The double logistic turns out to be the best convincing parametric method and from the non-parametric methods we choose the smoothing splines.

Chapter 4

NDVI Correction

Let's remind ourselves that the data from the S2 satellites is distributed with an SCL and we therefore have some evidence about what is observed at each pixel for each sampled time (c.f. table 2.2). So far, we have only considered points, labeled as cloud- and shadow-free (SCL45). However, we remind ourselves of the satellite images in figure 2.3d, where we had cloudy images despite the 'vegetation' label and see vegetation in figure 2.3e even though we are supposed to observe 'cirrus clouds'.

In this chapter, we will try to improve our NDVI interpolation by not relying only on the observed NDVI, but by training our own model to correct the NDVI using all S2 bands. For this, we introduce several statistical modelling approaches and discuss the strengths and weaknesses for each of them. After correcting the observed NDVI, we will assess the uncertainties of our corrections and translate them into weights. These will be used for the subsequent interpolation. This step-by-step procedure is illustrated by the figure B.4 in the appendix. Finally, we will evaluate which combination of interpolation methods and correction model performs the best.

4.1 Considering other SCL Classes

In figure 4.1 we plot the observed NDVI and notice that some blue points which correspond to the SCL-class 10 (thin cirrus clouds) follow the interpolated line closely. Hence, they might be useful in improving an interpolation fit.

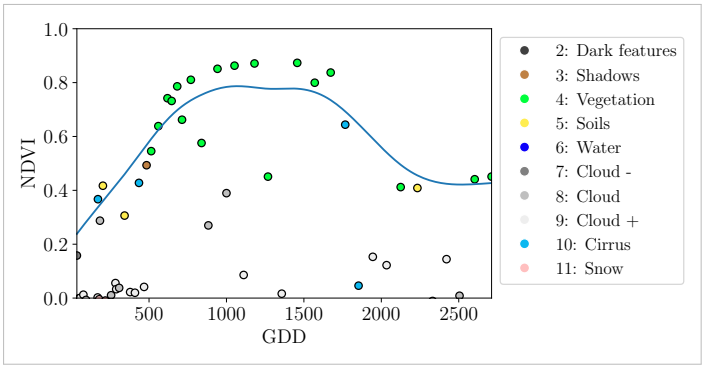


Figure 4.1: A smoothing splines fit considering green and yellow points (SCL45)

To get an impression of whether there is some useful information contained in non-SCL45

observations, we would like to compare the observed NDVI with the true NDVI. But since, we do not have any ground truth data, we will make the following assumption:

Assumption 1. The "true" NDVI value at time t can be successfully estimated by robustified LOOCV interpolation using high-quality observations. That is, the interpolated value (using a robustified interpolation method from chapter 3) considering the points $P^{SCL45} \setminus P_t$. In the following, we will call this estimate the "true"-NDVI.

We would like to get an idea if there is any information that can be recovered from non-SCL45 observations. For that, we will check for the other SCL-classes if there is a relation between the "true" NDVI (derived with robustified Smoothing Splines) and the observed NDVI. Thus, we pair each "true" NDVI with its observed one, collect all pairs, and create a scatter plot for each SCL-class in fig 4.2. As expected, the "true" and the observed NDVI seem to be highly correlated for SCL45. But we can also detect some patterns of correlation in the SCL-classes 2, 3, 7, 8 and 10.

It might be tempting to just include some of the mentioned SCL classes for interpolation. But on the one hand, the choice would not be objective and on the other hand, the correlation seems to be weaker than for SCL45. Therefore, in the following section, we will correct the observed NDVI and estimate the uncertainty of each correction.

4.2 Correction Models

For training an NDVI correction model, we require ground-truth data which we will aim to model using informative covariates. Since ground-truth NDVI data is not available, we will again use the assumption 1 and use the "true" NDVI instead. There is no canonical answer to the question of which covariates we should use. It is a tradeoff between simplicity, generalizability and performance (with the danger of overfitting). Our desire with the NDVI correction is to develop a product that is simple to use and understand. Therefore, in the subsequent, we will only take the spectral data of the satellite (i.e. all the bands) and the observed NDVI derived from it as covariates. We organize the choosen covariats in the desing matrix X[1], where each row corresponds to a P_t (i.e., a pixel at a time t) and each column to one covariate.

In the following, we will introduce different approaches, to model the relationship between the response $y := \text{NDVI}_{\text{“true”}} \in \mathbb{R}^n$ and the design matrix $X \in \mathbb{R}^{n \times p}$. First, we will study the basic OLS. Second, we look at the LASSO, an penalized adaptation of the OLS which is known to successfully deal with highly correlated covariates. Afterwards, GAMs are introduced which model the response similar to OLS but allow for non-linear relations. Last but not least, we discuss RF and MARS, which are both flexible modelling approaches.

Note that in order to reduce computation time, only 10% of the data has been used to fit the subsequent models, which are still more than 120'000 observations.

4.2.1 Ordinary Least Squaers (*OLS*)

The OLS is a linear model which aims to minimize the sum of the squared residuals. We assume a linear relationship between y and X and allow for Gaussian noise. That is:

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

[1] Strictly speaking, we include also the intercept and introduce one dummy variable for each SCL-class

Assuming that (X^TX) is regular, we can estimate the regression coefficients β by

$$\hat{\beta} = (X^TX)^{-1}X^Ty = \underset{\beta in \mathbb{R}^p}{\arg\min} \|y - X\beta\|_2^2$$

We will train two models, one using all covariates discussed above and one using only the SCL-classes and the observed NDVI.

Advantages	Disadvantages
— Simple method with good interpretability of coefficients. — Computationally cheap.	— Catches only linear relationships. — No integrated variable selection.[2]

4.2.2 Least Absolute Shrinkage and Selection Operator (LASSO)

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

$$\hat{\beta}_\lambda = \underset{\beta \in \mathbb{R}^p}{\arg\min} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 = \underset{\beta \in \mathbb{R}^p \text{ and } \|\beta\|_1 < \lambda}{\arg\min} \|y - X\beta\|_2^2.[3] \quad (4.2.2.1)$$

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

Tibshirani (2011) shows that the LASSO solution tends to be sparse. That is $\beta_i = 0$ for most $i = 1, \ldots, p$. The larger λ, the more $\beta_i = 0$ and hence the simpler the resulting model.

In order to know which λ to choose, we try a huge range of possible values. For each β_λ, we calculate the cross-validated $RMSE_\lambda$ [4] (and its standard deviation σ_λ using the k folds) and define the λ with the smallest corresponding $RMSE_\lambda$ as λ_{min}. From here we choose the largest λ for which the $RMSE_\lambda$ is smaller than $RMSE_{\lambda_{min}} + \sigma_\lambda$. This yields a simpler model while keeping the $RMSE$ reasonable model.

We will apply the LASSO using the selected covariates in section 4.2 and their second degree of interactions.[5]

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

[3] The last two terms are equivalent by lagrangian optimization

[4] The cross validated Root Mean Square Error is the mean of the RMSE's obtained for each fold using the model trained on the remaining folds.

[5] This is if our covariates are $\{1, a, b\}$, then we will now use $\{1, a, b, ab, a^2, b^2\}$.

4.2.3 General Additive Model (*GAM*)

GAMs as described in Hastie and Tibshirani (1987) are a special case of Projection Pursuit Regression, where only the p directions parallel to the coordinate axes are considered. The result is different to a linear model since the coordinate functions are not restricted to be linear but are assumed to be non-parametric functions. The model can be written as:

$$g_{add}(x) = \mu + \sum_{i=1}^{p} g_j(x_j).^{6}$$

To estimate the non-parametric functions, we can use smoothing splines (ref sec. 3.3.6). For this let $\mathcal{S}_j$ be the function which takes some $z \in \mathbb{R}^n$ and returns the smoothing splines fitted to $(X_{:,j}, z)$ where the smoothing parameter is optimized by GCV. Since we cannot fit all g_j simultaneously, we will use a strategy named Backfitting. We basically cycle through the indices $1, \dots p$ and refit $\hat{g}_j$ each time. The following illustrates the procedure:

$$\begin{aligned}
&1) \quad \hat{g}_1 &&= \mathcal{S}_1(y - \mu) \\
&2) \quad \hat{g}_j &&= \mathcal{S}_j(y - \mu - \hat{g}_1(X_{[:,1]}) - \cdots - \hat{g}_{j-1}(X_{[:,j-1]})) \quad \text{for } j = 2, \dots, p \\
&3) \quad \hat{g}_1 &&= \mathcal{S}_1(y - \mu - \hat{g}_2(X_{[:,2]}) - \cdots - \hat{g}_p(X_{[:,p]})) \\
&4) \quad \hat{g}_j &&= \mathcal{S}_j(y - \mu - \sum_{k \neq j} \hat{g}_k(X_{[:,k]})) \quad \text{for } j = 2, \dots, p \\
&\quad \vdots
\end{aligned}$$

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

Advantages	Disadvantages
— Captures non-linearity.	— No automatic variable selection.
— Good interpretability.	— Computationally expensive.

4.2.4 Random Forest (*RF*)

To define a random Forest introduced by Breiman (2001) we will first define what a Tree is. A *(decision) Tree* is a graph (V, E) without circles, a distinct root node, every node has at most two children and every leaf has a value assigned to it. At each node there is a boolean condition testing if one variable is greater than some value and a pointer to one child depending on the boolean value. To evaluate a tree we start at the root node, test the boolean expression and go to the node indicated by the resulting pointer. This we repeat until we end up at a leaf-node, where we return the value assigned to it.

To build such a Tree, we will recursively partition the covariate space using greedy splits[7] decreasing the RMSE[8] each time. If the set we want to split contains less than a certain amount of training points, we stop.

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

[7] For computational reasons, we will only use splits along one covariate. So we 'cut' our covariate space into rectangles.

[8] 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).

To build a *Random Forest* we will bootstrap-aggregate[9] many such Trees[10]. The prediction of the Random Forest for a new point x is then the mean of the predictions from all the Trees.

Advantages	Disadvantages
— Captures non-linear relationships.	— The resulting (prediction) function is not continuous but locally constant.
— Captures all interactions and performs automatic variable selection.	— Computationally expensive.
— Can deal with missing data.	— No interpretability.

4.2.5 Multivariate Adaptive Regression Splines (*MARS*)

A MARS model as introduced in Friedman (1991) can be described by

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

where the h_m are simple functions (explained later) and the β_m are estimated via Least Squares.

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

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

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

$$\mathcal{C} := \{(h(\cdot)b_1(\cdot), h(\cdot)b_2(\cdot)) \mid h \in \mathcal{M},\ (b_1, b_2) \in \mathcal{B}\} \tag{4.2.5.1}$$

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

Finally, to avoid overfitting, we prune the set $\mathcal{M}$ by optimizing a LOOCV score.[11]

To reduce computational complexity, we follow the recommendation from Stephen (2021) and restrict h in equation (4.2.5.1) to be of degree one (so it is also in a pair of $\mathcal{B}$). Consequently, $\mathcal{C}$ contains functions with a degree of at most 2.

[9] That is we will sample (with replacement) several times n observations from our original data and fit a Tree to each such sample.

[10] 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 "second best split" can be selected.

[11] 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 LOOCV score.

Advantages	Disadvantages
— Catches non-linear relationships. — Interpretability via functions in $\mathcal{M}$ and their coefficients. — Allows for interactions with variable selection.	— Computationally expensive (can be reduced by restricting the degree of interactions).

4.3 Uncertainty Estimation

Once we corrected the NDVI using the models described in the previous section, we are left with the problem that not every correction is equally reliable.[12]. Hence, we are interested in a measure of how uncertain an estimate is.

We achive this analogously as we corrected the NDVI, by replacing the response ($\text{NDVI}_{\text{“true”}}$) with the absolute residuals $v := |y - \hat{y}|$ and modeling their relationship with the covariates defined by X. In this way, we obtain a model for the absolute residuals v and the estimator $\hat{v}$.

4.4 Interpolation

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

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

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. The normalization is needed since for some interpolation methods, inflating the sum of weights would decrease the effect of the smoothing.

4.5 Resulting Interpolation Strategies

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

i.) LOOCV Interpolation (+ robustify?) to get “true” NDVI
ii.) Correction
iii.) Uncertainty estimation
iv.) Interpolation (+ robustify?)

At each step we have a choice, more precisely:

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

[12] One correction is illustrated in the figure B.4f. In this figure, the outer points (labeled as clouds) have a large scatter.

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

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

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

4.6 Evaluation Method

In this section, we introduce the relative yield-estimation-accuracy (*RYEA*) and utilize it to evaluate the 28 interpolation strategies from section 4.5. The fundamental assumption is that the closer the interpolated NDVI time series is to the true one, the better it can be used to determine crop yield. Implicitly, we believe that an NDVI time series which better models yield will incorporate more true information about the underlying vegetation. Therefore, we want to determine a comparable RYEA for each interpolation strategy and choose it as a benchmark criterion. This is an objective measure, since we have not considered crop yield in any of our previous steps. Moreover, this criterion is justified by the fact that yield estimation has been a motivation for the interpolation.

Definition 4.6.0.1. *(RYEA) Let $y \in \mathbb{R}^n$ be the yield, M be a model for estimating y, and $\hat{y} = M(X)$ where X describes the data*[13]*. We define the RYEA as the relative RMSE in yield estimation. Formally expressed:*

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

where $\bar{y}$ denotes the sample mean.

4.6.1 Yield Estimation

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

— Maximum slope
— Minimum slope
— Integral[14] over all
— Peak (i.e. maximal NDVI)
— GDD for the Peak
— Integral[14] up to the peak
— Integral[14] after peak
— Integral[14] from 0-685 GDD
— Integral[14] from 685-1075 GDD

[13] We will use the matrixes derived in section 4.6.1
[14] 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

For the choice we were inspired by (c.f. table 2 in Kamir, Waldner, and Hochman (2020)). However, we deliberately omit any statistic that involves the minimum (e.g. the NDVI-range), since we regard the minimum as a very error-prone measure due to the large influence of clouds in the time series.

As a result, for each interpolation strategy, a matrix is obtained in which each row corresponds to a pixel and both the yield and the covariates (computed by applying the above statistics) are contained. Using this matrix, we train a random forest for yield estimation, and compute the integrated OOB estimates[15] $\hat{y}$. Note that 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. Finally, for each interpolation strategy, we calculate the RYEA and describe the results in section 5.2.

[15] 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 $\frac{1}{e}$).

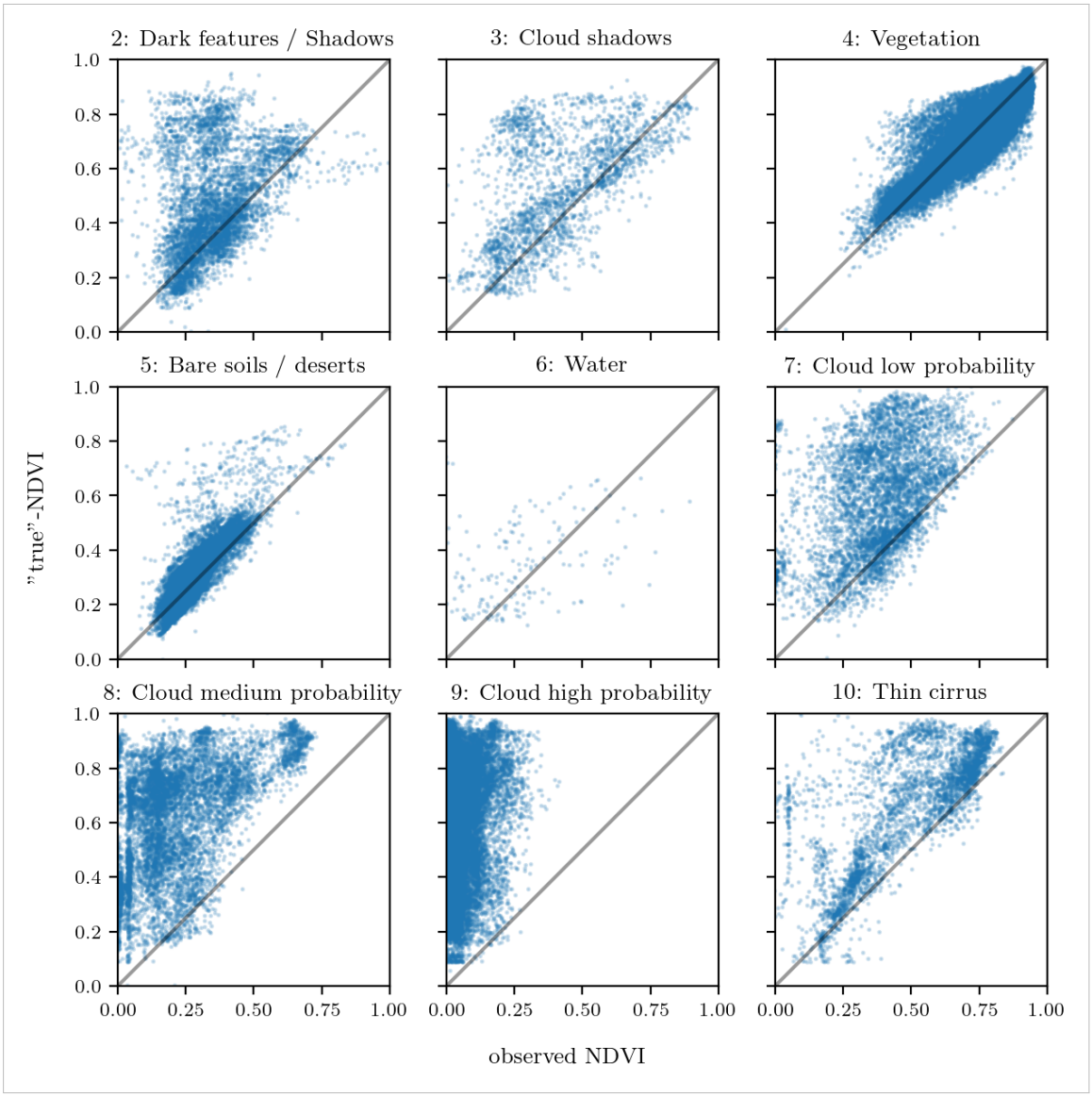


Figure 4.2: For each SCL class, we compare the true NDVI with the observed NDVI. (The true NDVI was estimated with OOB smoothing splines, and we used all observations of 10% of the total training pixels.)

Chapter 5

Results

5.1 Goodness of Fit for Selected Interpolation Methods

Table 5.1 benchmarks the selected[1] interpolation methods (on P^{SCL45}) with respect to various score functions. The score functions take the absolute values of the LOOCV residuals and summarize them in a number (the smaller, the better). For each of the 5 selected interpolation methods, we consider the basic and the robustified (see section 3.5) version.

Table 5.1: Comparing the goodness of fit for selectedinterpolation methods (on P^{SCL45}) measured with the score functions (which take the LOOCV residuals as input) listed in the left column. q_X denotes here the $X\%$ quantile.

	SS	LOESS	DL	BSPL	FR	SS^{rob}	$LOESS^{rob}$	DL^{rob}	$BSPL^{rob}$	FR^{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

DL is the best among both robustified and non-robustified with respect to most of the score functions used (all except q95) and is especially superior to the other parametric approach, which is Fourier interpolation. Especially the robust Fourier interpolation performs poorly. The LOESS dominates (i.e. is superior on every score function) all other non-prametric methods, but is closely followed by the SS. The BSPL, on the other hand, is the worst non-parametric method tested here.

5.2 XXX (Robustification and) NDVI-Correction

defiition of RYEA, it is not an accuracy but an error

The RYEA for the 28 (in section 4.5) chosen interpolation strategies is given in table 5.2.

Robustification in the interpolation strategies, does not improve the quality of the fit

[1] For the discussion which methods have been selected c.f. section 6.1.2.

Table 5.2: RYEA. For the non-relative RMSE and the coefficient of determination (R^2) see table B.1 and B.2.

	RF	OLS-SCL	OLS-all	MARS	GAM	LASSO	no-correction
ss	0.155	0.140	0.143	0.142	0.142	0.142	0.149
dl	0.156	0.151	0.152	0.152	0.149	0.149	0.158
ss-rob	0.155	0.143	0.147	0.149	0.146	0.145	0.148
dl-rob	0.157	0.153	0.152	0.145	0.148	0.150	0.157

(measured via the RYEA) in most cases. In addition, SS (rob) are better than DL(rob) in terms of RYEA, with one exception.

The interpolation strategy that leads to the lowest RYEA is the OLS-SCL with SS. Given that the OLS-SCL models have very good interpretability, we also present the regression equations below. The corrected NDVI is calculated using

$$\begin{aligned}\text{NDVI}_{\text{corr}} =& 0.711\,\text{NDVI}_{\text{observed}} + \mathbb{1}_{SCL=2}0.215 + \mathbb{1}_{SCL=3}0.237 + \mathbb{1}_{SCL=4}0.210 \\ &+ \mathbb{1}_{SCL=5}0.116 + \mathbb{1}_{SCL=6}0.162 + \mathbb{1}_{SCL=7}0.327 + \mathbb{1}_{SCL=8}0.474 \\ &+ \mathbb{1}_{SCL=9}0.575 + \mathbb{1}_{SCL=10}0.306 + \mathbb{1}_{SCL=11}0.512\end{aligned} \tag{5.2.0.1}$$

where $\mathbb{1}_{SCL=2}$ is equal to one if the current observation corresponds to SCL class 2 and zero otherwise.[2]. Whereas, we obtain the estimated absolute residuals by:

$$\begin{aligned}\widehat{\text{abs}}\,(\text{NDVI}_{\text{“true”}} - \text{NDVI}_{\text{corr}}) =& -0.133\,\text{NDVI}_{\text{observed}} + \mathbb{1}_{SCL=2}0.186 + \mathbb{1}_{SCL=3}0.185 \\ &+ \mathbb{1}_{SCL=4}0.146 + \mathbb{1}_{SCL=5}0.089 + \mathbb{1}_{SCL=6}0.167 \\ &+ \mathbb{1}_{SCL=7}0.203 + \mathbb{1}_{SCL=8}0.181 + \mathbb{1}_{SCL=9}0.173 \\ &+ \mathbb{1}_{SCL=10}0.180 + \mathbb{1}_{SCL=11}0.172\end{aligned} \tag{5.2.0.2}$$

In the equation 5.2.0.1, we notice the strongest upwards correction for SCL classes 8, 9 and 11 (correspond to ‘medium probability clouds’, ‘high probability clouds’ and ‘thin cirrus clouds’). The estimated absolute resiudas, however, are the smallest for SCL classes 4 and 5 (‘vegetation’ and ‘bare soil’). Furthermore, the higher the observed NDVI the lower are the estimated absolute residuals.

For the `R`-output of the `summary` function of the two models, we refer to the appendix B.3.1.

[2] $\mathbb{1}$ is also called an indicator function or characteristic function in mathematics.

Chapter 6

Discussion

Here in the discussion, you should take up the points you mentioned in the introduction

6.1 Interpolation Methods

6.1.1 Data Gaps in Time Series

Kernel regression estimates the value for t by relating to the points near t. To determine what "near" means, a bandwidth h is used (c.f. equation 3.3.1). This gets problematic as soon as the data gaps become larger than h, since in this case no points are left that are considered to be close to t.

Regarding the GK, we expect that because of the stationarity assumption, the interpolation will tend to the mean if data gaps are present (c.f. figure 3.3).

Since the SG Filter requires equidistant points, it is clear that data gaps will break it. wertend
The linear interpolation, which is supposed to recover this, we consider as not being a satisfying solution.

We do not trust the FR interpolation if there are noticeable data gaps. On the one hand, it corresponds to our experience that the curve can escape strongly there (c.f. figure 3.1). On the other hand, the unreliability is illustrated by the poor values in table 5.1 for the robustified variant. These are meaningful in describing the ability to cope with data gaps, since more data points are ignored during the robustification and thus data gaps are simulated.

Similarly, for SS, LOESS, DL and B-splines we compare the values in table 5.1 between the robustified and non-robust variant. We find that the robust variant is not very different from the non-robust variant (unlike FR). Thus, we conclude that these methods do not have systematic failures.

Regarding the LOESS, we observe in the figure B.1 in plot (c) a strange peak between the first and second observation. This peak is due to the local weighting. In case of data gaps, the weights can attain non-intuitive values. For instance, the first data point in the plot, although adjacent to the peak, is given a low weight compared to the points to the right of the peak (for estimating the value at this peak).

In our experience, the DL handles data gaps well, but it may happen that the model describes the NDVI increase as abrupt. This however was fixed, by bounding the first derivative (c.f. section 3.2.3).

6.1.2 Preselection

We shall now justify our preselection of the interpolation methods tested in section 3.6. We decided against kernel regression because it has systematic errors at peaks and valleys. Moreover, this method handles data gaps poorly (c.f. 6.1.1). Moreover, we will not consider kriging since the underlying assumptions are not met and therefore a systematic bias is introduced. On top of that, ML parameter finding occasionally fails. Also, we do not include the SG filter in the next selection, since we think of it as a special case of LOESS.

6.1.3 Candidate Selection

Given that DL convinces regarding most of the selected score functions in table 5.1 we will certainly investigate this method in chapter 4. Moreover, we see that the robustification mostly improved the score regarding the 50, 75, 85, and 90 % Quantiles. Only for the outlier-sensitive score functions (RMSE and q95)[1] we notice significant worsening (we consider the robust Fourier separately in section 6.1.1). Consequently, we will also use the robustification in section 4. Not wanting to rely on the form assumptions of the DL, we further choose a non-parametric method for further consideration. Despite the LOESS slightly dominating the SS in table 5.1, we choose the SS. This is due to the strange behavior of the LOESS in case of data gaps (see section 6.1.1) and the good interpretability of the SS using the minimization function 3.3.6.1.

XXX discuss results from table B.1

6.2 NDVI Correction

6.2.1 Bootstrap

The question arises if we can build the correction model on the same year as we want to apply it on. Usually, a similar approach might carry the danger of overfitting. However, we have not used any ground truth at any point (until the evaluation). Instead, we estimated the "true" NDVI with the assumption 1 via OOB. Thus, we have bootstrapped our way out of the problem. Consequently, we reason that we can apply our method to a new (comparable) dataset and solve the correction again via this bootstrap.

6.2.2 Using Additional Covariates

In section 4.2 we have only used the spectral data (and the observational NDVI calculated from them) as covariates. Since we have the weather data available (c.f. REF-SEC), it would be a small effort to incorporate it, together with statistics collected from it (i.e. GDD or 'rainfall in the last 30 days').

We decided against using this data, because on the one hand we have the problem that we have practically too few observations (we observe only 5 years) and we expect the

where does this section belong to? Chapter 'NDVI Correction' or 'Further Work'?

[1] For the RMSE one outlier is enough to take away the usefulness of the statics, in the case of q95 it is enough if 5% of the data are corrupt to break the statics.

weather in our study region to be rather homogeneous which is suggested by the fact that the weather data published by Meteoswiss are for a grid with a resolution of 1 km. On the other hand, we want the underlying model not to learn improper relationships. For example, the model might automatically predict a high NDVI for a day in summer (detected by high GDD / many sunshine hours / high temperature) just because it is "used" to observing a lot of vegetation in summer. Including temporally (e.g., P_{t-1} and P_{t+1}) and geographically adjacent pixels would likely improve performance. However, for simplicity, we omit it here[2].

6.2.3 Which Interpolation Strategy should we choose

table mit OLS SCL als sieger diskutieren

6.2.4 High RMSE in Yield Prediction

How much can we expect to get? We have multiple sources of uncertainty in the data:

i.) Uncertainty in Yield data collected by the combine harvester

ii.) Uncertainty in Yield data through rasterization

iii.) Uncertainty in satellite images through "measurement errors" introduced via clouds and other atmospheric effects

iv.) Uncertainty introduced by interpolating (especially when long data-gaps are present)

You already capture the "main" structure of your thesis with the interpolation and the NDVi correction sections. Can you combine them both in a "synthesis" subsection at the end of the discussion?

kurzer kontext von vergle-ichbaren values von gregor — diese sektion ist für dena uf-traggeberr

[2]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).

Chapter 7

Conclusion

```
- itpl methods,
    parametric dl
    non-param
        discarded
            kernel methods because of strong bias
            kriging because assumptions not met and ML parameter erstimation issiues
            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 secon
- robustifying apparently not responsible for big improvements
```

XXX draw your conclusion to which you came during this thesis

Let us recapitulate the interpolation strategy introduced in chapter 4: We estimate the true NDVI using SS via LOOCV, then obtain the corrected NDVI using the OLS-SCL model. Subsequently, we estimate the absolute error with the OLS-SCL model and thereby obtain weights that are supposed to reflect the reliability of the corrected NDVI. Finally, we perform a weighted interpolation with SS.

7.1 Future Work

7.1.1 Time Series Correction-Interpolation as a General Method

Throughout this thesis, we developed a correction and interpolation method for the NDVI. However, we never used features of the NDVI. Only the parameter estimated via cross-validation in chapter 3.4 depends on the scale of the time series. For simplicity, we could thus determine the parameter using Generalized Cross Validation (as Ripley and Maechler suggest). Therefore, our approach of interpolation and correction of time series can be applied to arbitrary time series as long as additional information is available. However, further research is required, to demonstrate the usefulness of this approach in general.

Example: Cloud Correction with Uncertainty Estimation and Interpolation

This generalization can be used in particular for cloud correction. In the same manner as we corrected the NDVI time series in chapter 4, we can correct each spectral band and reunite the corrected bands with the uncertainties. If desired, the time series can also be interpolated before merging as in chapter 4.4. The resulting question would be how well this approach performs.

7.1.2 Minor Improvements

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

- **Data:** Method how data has been extrapolated to the grid could possibly be improved
- **Data:** For computational reasons, we mostly considered all years and split the data (on the pixel level) randomly into a train/test set. A leave one year out cross validation might yield more accurate results.
- **Data:** We have not included the spectral bands which have a resolution of 60 m. But precisely these seem to be promising for cloud correction, since they are a proxy of the water (content and form) in the atmosphere.
- **Data:** Raiyani, Gonçalves, Rato, Salgueiro, and Marques da Silva (2021) presents an Machine Learing approach that supposedly improves the SCL and thus could improve our results which are based on the SCL.
- **NDVI Correction:** Explore the effect of different link and normalizing functions in section 4.4. Currently we run into the danger of some outer points getting nearly ignored just because one estimated absolute residual for some interior point is very small.
- **NDVI Correction:** Yield is not the only target variable of interest. Other variables like protein content could also be used in section 4.6 for the method evaluation.

which data? I assume the combine harvester point data?

Bibliography

(2007). Gaussian models for geostatistical data. In P. J. Diggle and P. J. Ribeiro (Eds.), *Model-Based Geostatistics*, pp. 46–78. New York, NY: Springer.

Bailey, S. J. (2018, July). Using Growing Degree Days to Predict Plant Stages. pp. 8.

Beck, P. S. A., C. Atzberger, K. A. Høgda, B. Johansen, and A. K. Skidmore (2006, February). Improved monitoring of vegetation dynamics at very high latitudes: A new method using MODIS NDVI. *Remote Sensing of Environment 100*(3), 321–334.

Breiman, L. (2001, October). Random Forests. *Machine Learning 45*(1), 5–32.

Brockmann, M., T. Gasser, and E. Herrmann (1993, December). Locally Adaptive Bandwidth Choice for Kernel Regression Estimators. *Journal of the American Statistical Association 88*(424), 1302–1309.

Cao, R., Y. Chen, M. Shen, J. Chen, J. Zhou, C. Wang, and W. Yang (2018, November). A simple method to improve the quality of NDVI time-series data by integrating spatiotemporal information with the Savitzky-Golay filter. *Remote Sensing of Environment 217*, 244–257.

Chen, J., P. Jönsson, M. Tamura, Z. Gu, B. Matsushita, and L. Eklundh (2004, June). A simple method for reconstructing a high-quality NDVI time-series data set based on the Savitzky–Golay filter. *Remote Sensing of Environment 91*(3), 332–344.

Cleveland, W. S. (1979, December). Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association 74*(368), 829–836.

Friedman, J. H. (1991, March). Multivariate Adaptive Regression Splines. *The Annals of Statistics 19*(1), 1–67.

Hastie, T. and R. Tibshirani (1987, June). Generalized Additive Models: Some Applications. *Journal of the American Statistical Association 82*(398), 371–386.

Jaramaz, D., V. Perović, S. Belanovic Simic, E. Saljnikov, D. Cakmak, V. Mrvić, and L. Zivotic (2013, May). The ESA Sentinel-2 mission Vegetation variables for Remote sensing of Plant monitoring.

Kamir, E., F. Waldner, and Z. Hochman (2020, February). Estimating wheat yields in Australia using climate records, satellite image time series and machine learning methods. *ISPRS Journal of Photogrammetry and Remote Sensing 160*, 124–135.

Lyche, T. and K. Morken (2005, January). Spline Methods.

McMaster, G. S. and W. W. Wilhelm (1997, December). Growing degree-days: One equation, two interpretations. *Agricultural and Forest Meteorology 87*(4), 291–300.

Perich, G., M. O. Turkoglu, L. V. Graf, J. D. Wegner, H. Aasen, A. Walter, and F. Liebisch (2022, July). Pixel-based yield mapping and prediction from Sentinel-2 using spectral indices and neural networks.

Raiyani, K., T. Gonçalves, L. Rato, P. Salgueiro, and J. R. Marques da Silva (2021, January). Sentinel-2 Image Scene Classification: A Comparison between Sen2Cor and a Machine Learning Approach. *Remote Sensing 13*(2), 300.

Ripley, B. D. and M. Maechler. R: Fit a Smoothing Spline. https://stat.ethz.ch/R-manual/R-patched/library/stats/html/smooth.spline.html.

Rouse, J. W. (1974, May). Monitoring the vernal advancement and retrogradation (green wave effect) of natural vegetation. Technical Report NASA-CR-139243.

Savitzky, A. and M. J. E. Golay (1964, July). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry 36*(8), 1627–1639.

Schafer, R. W. (2011, July). What Is a Savitzky-Golay Filter? [Lecture Notes]. *IEEE Signal Processing Magazine 28*(4), 111–117.

Skakun, S., E. Vermote, B. Franch, J.-C. Roger, N. Kussul, J. Ju, and J. Masek (2019, July). Winter Wheat Yield Assessment from Landsat 8 and Sentinel-2 Data: Incorporating Surface Reflectance, Through Phenological Fitting, into Regression Yield Models. *Remote Sensing 11*(15), 1768.

Stephen, M. (2021, July). Earth: Multivariate Adaptive Regression Splines.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: A retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(3), 273–282.

Appendix A

Reproducibility

A.1 Reproduce Results

For reproducibility of the whole computations, we refer to our codebase at:
https://github.com/LGraz/MasterThesis-Code
In order to reproduce our computations and results, set up the directory as described in the README and execute the computations via `./shell_scripts/reproduce.sh` and do not execute the python and R scripts by hand (unless you follow the order in `./shell_scripts/reproduce.sh`).

A.2 R–Package

We also provide an `R` package for a general time series correction and interpolation if additional data is available at:
https://github.com/LGraz/CorrectTimeSeries
In our case we consider the NDVI time series and the additional data consists of the unused spectral bands.

We recommend installing it via the `devtools` package by:
`devtools::install_github("LGraz/CorrectTimeSeries")`

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

```
library(CorrectTimeSeries)

# load a list of dataframes, each one describes one pixel with the covariates and
    the response
data(timeseries_list)
str(timeseries_list[[1]])

# Train/Load  RF
train_model_myself ← TRUE
if (train_model_myself){
    # Add "true" NDVI (or generally the response), by Out-Of-Bag estimation
    timeseries_list ← lapply(timeseries_list, function(df) {
        df$oob_ndvi ← OOB_est(df$gdd, df$ndvi_observed) # gdd is the time-axis
        df
    })
    # Train correction model
    formula ← "oob_ndvi ~ B02+B03+B04+B05+B06+B07+B08+B8A+B11+B12+scl_class"
    RF ← train_RF_with_fromula(formula, timeseries_list, robustify=TRUE)
} else {
```

```
    data(RF_for_NDVI)
    RF ← RF_for_NDVI
}

# ADD CORRECTION
timeseries_list ← lapply(timeseries_list, function(df) {
    df$corrected_ndvi ← randomForest:::predict.randomForest(RF, df)
    df
})

# Get interpolation for each timeseries
newx ← 1:1000
lapply(timeseries_list, function(df){
    ss ← smoothing_spline(df$gdd, df$corrected_ndvi)
    predict(ss, newx)$y
})
```

Example of how to use the `CorrectTimeSeries` package

Appendix B

Further Material

B.1 Data and Methods

B.1.1 GDD

Bailey (2018) tabulates the corresponding GDD for each stage of wheat.

Stage	Description	GDD
Emergence	Leaf tip just emerging from above-ground coleoptyle.	125 – 160
Leaf development	Two leaves unfolded.	169 – 208
Tillering	First tiller visible	369 – 421
Stem elongation	First node detectable.	592 – 659
Anthesis	Flowering commences; first anthers of cereals are visible.	807 – 901
Seed fill	Seed fill begins. Caryopsis of cereals watery ripe (first grains have reached half of their final size).	1068 – 1174
Dough stage	Soft dough stage, grain contents soft but dry, fingernail impression does not hold.	1434 – 1556
Maturity complete	Grain is fully mature and drydown begins. Ready for harvest when dry.	1538 – 1665

B.2 Interpolation

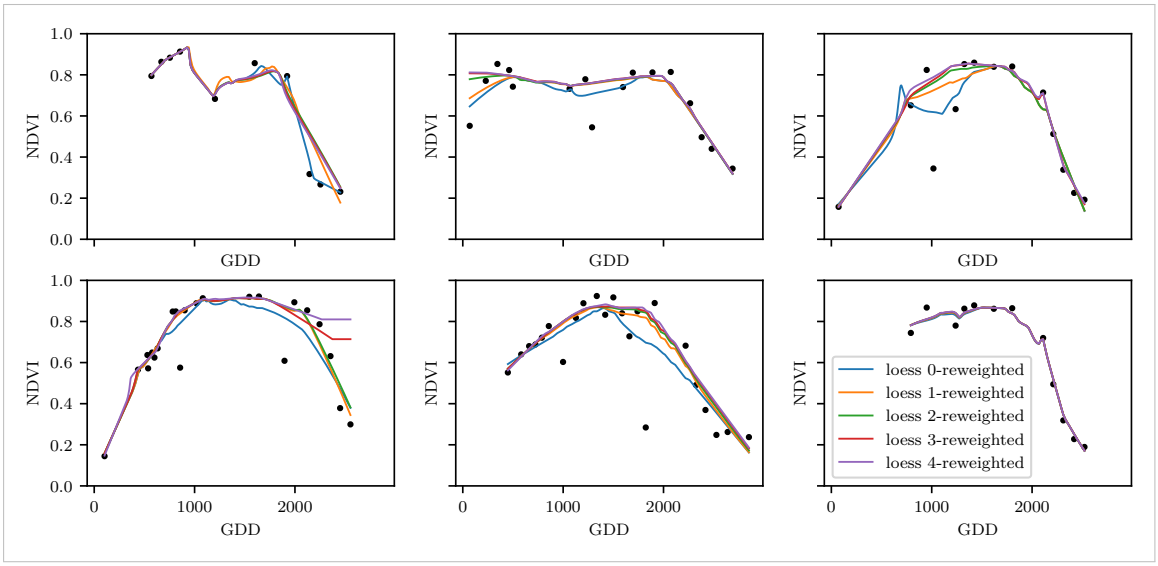


Figure B.1: The LOESS smoother fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.5) are also displayed

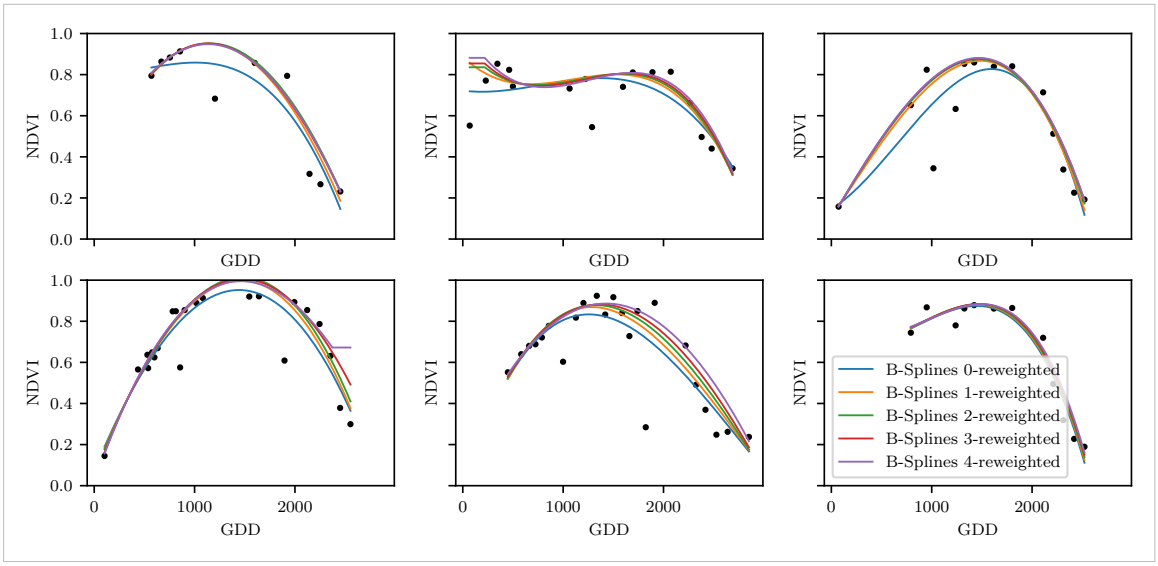


Figure B.2: B-Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.5) are also displayed

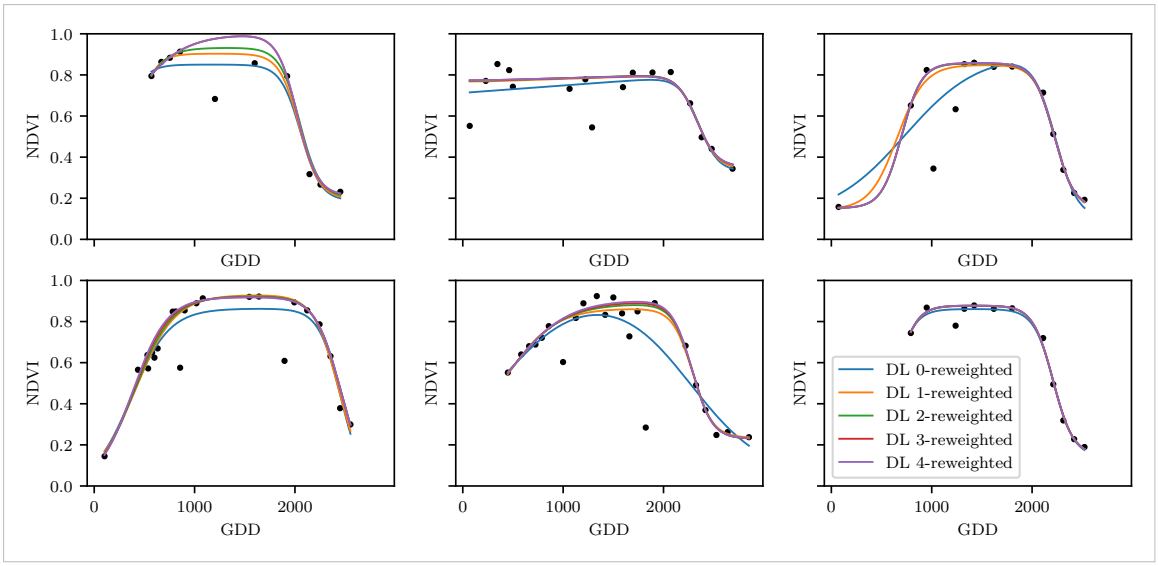


Figure B.3: A Double Logistic curve fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.5) are also displayed

B.3 NDVI correction

page breaks

Table B.1: Non-relative RMSE for yield prediction

	RF	OLS-SCL	OLS-all	MARS	GAM	LASSO	no-correction
ss	1.144	1.033	1.051	1.042	1.046	1.042	1.095
dl	1.150	1.115	1.116	1.116	1.097	1.098	1.159
ss-rob	1.144	1.054	1.084	1.094	1.072	1.071	1.091
dl-rob	1.159	1.128	1.117	1.064	1.093	1.105	1.156

Table B.2: Coefficient of determination (R^2) of yield prediction

	RF	OLS-SCL	OLS-all	MARS	GAM	LASSO	no-correction
ss	0.431	0.486	0.477	0.481	0.479	0.481	0.455
dl	0.427	0.445	0.444	0.444	0.454	0.453	0.423
ss-rob	0.431	0.475	0.461	0.456	0.467	0.467	0.457
dl-rob	0.423	0.439	0.444	0.470	0.456	0.450	0.424

B.3.1 OLS-SCL Model Outputs

```
Call:
lm(formula = (paste(response, " ~ ", "ndvi_observed + scl_class")),
    data = ndvi_df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.7997  -0.0717   0.0039   0.0695   0.6632

Coefficients:
```

```
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.21465    0.00230   93.46  < 2e-16 ***
ndvi_observed   0.71116    0.00346  205.65  < 2e-16 ***
scl_class3      0.02205    0.00356    6.20  5.8e-10 ***
scl_class4     -0.00431    0.00251   -1.72    0.085 .
scl_class5     -0.09875    0.00234  -42.15  < 2e-16 ***
scl_class6     -0.05301    0.01104   -4.80  1.6e-06 ***
scl_class7      0.11245    0.00274   41.09  < 2e-16 ***
scl_class8      0.25963    0.00253  102.57  < 2e-16 ***
scl_class9      0.35994    0.00236  152.47  < 2e-16 ***
scl_class10     0.09091    0.00308   29.54  < 2e-16 ***
scl_class11     0.29784    0.00392   76.06  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.146 on 124978 degrees of freedom
Multiple R-squared:  0.532,     Adjusted R-squared:  0.532
F-statistic: 1.42e+04 on 10 and 124978 DF,  p-value: <2e-16
```

R Summary of the NDVI correction model (c.f. equation 5.2.0.1)

```
Call:
lm(formula = (paste(get_res(), " ~ ", "ndvi_observed + scl_class")),
    data = ndvi_df)

Residuals:
    Min      1Q  Median      3Q     Max
-0.2051 -0.0427 -0.0074  0.0329  0.6589

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.18647    0.00126  147.74  < 2e-16 ***
ndvi_observed  -0.13265    0.00190  -69.80  < 2e-16 ***
scl_class3     -0.00180    0.00196   -0.92   0.3587
scl_class4     -0.04069    0.00138  -29.55  < 2e-16 ***
scl_class5     -0.09698    0.00129  -75.32  < 2e-16 ***
scl_class6     -0.01906    0.00606   -3.14   0.0017 **
scl_class7      0.01641    0.00150   10.91  < 2e-16 ***
scl_class8     -0.00560    0.00139   -4.02  5.7e-05 ***
scl_class9     -0.01384    0.00130  -10.67  < 2e-16 ***
scl_class10    -0.00690    0.00169   -4.08  4.5e-05 ***
scl_class11    -0.01446    0.00215   -6.72  1.8e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08 on 124978 degrees of freedom
Multiple R-squared:  0.352,     Adjusted R-squared:  0.352
F-statistic: 6.8e+03 on 10 and 124978 DF,  p-value: <2e-16
```

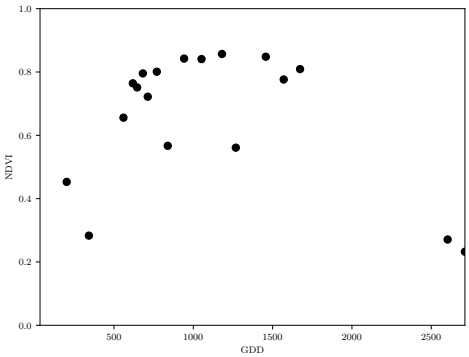
R Summary of the NDVI correction model (c.f. equation 5.2.0.2)

replace space before ref by tilda

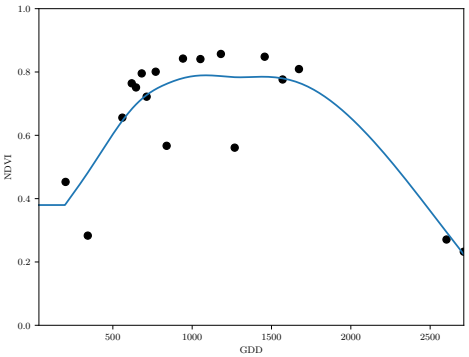
check quantile definitions

schwarz weiss färbung der IS tabelle korrigieren

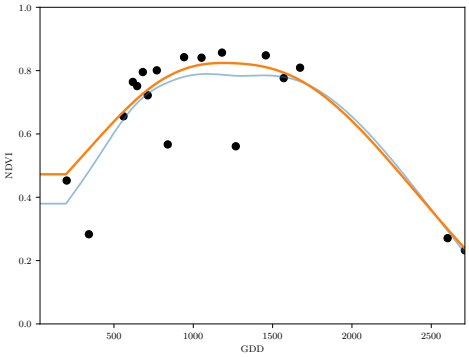
so wenig wie möglich abkürzungen in den fig und table captions



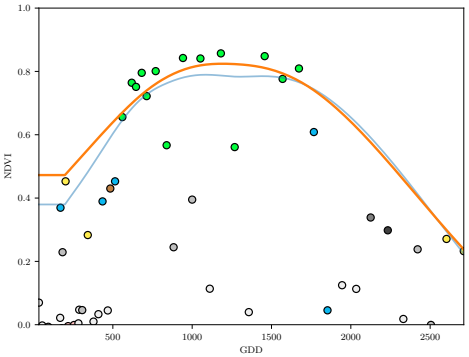
(a) NDVI time series with scl45



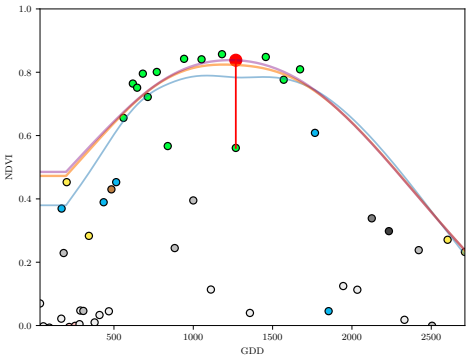
(b) Interpolation via Smoothing Splines



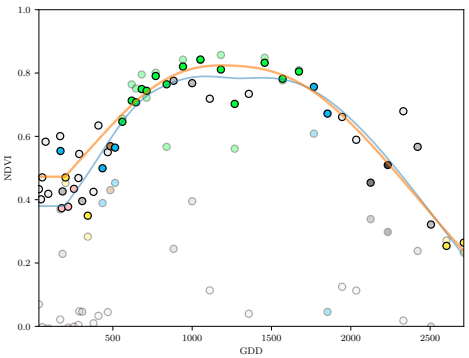
(c) Robustly reweighted fit



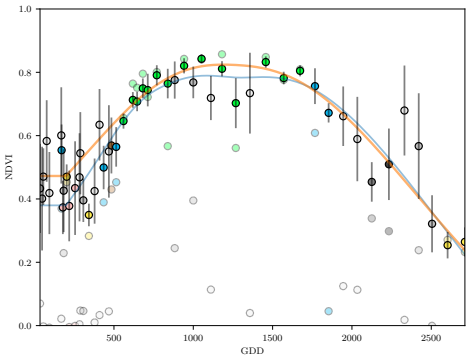
(d) Now also consider other SCL-classes



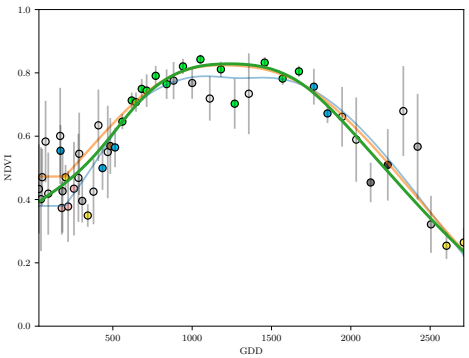
(e) OOB estim. for each point using SCL45



(f) Correct NDVI



(g) Estimate absolute errors



(h) Robust interpolation on weights derived from uncertainties.

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