Contents

Notation

Conventions for Variables

c: a (vector of) constant(s)

$\lambda \in \mathbb{R}$: a scalar

$n \in \mathcal{N}$: sample size

i, j are indices in $\{1, \ldots, n\}$

$x \in \mathbb{R}^n$: covariable in 1-dim interpolation setting

$w \in \mathbb{R}^n$: 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

$\bar{y} \in \mathbb{R}$: mean of y

$r \in \mathbb{R}^n$: residuals given by $y - \hat{y}$

Abbreviations and Objects

Pixel: A pixel describes a specific location in a field. It has the size of 10 x 10 meters and coincides with the resolution (and location) of the sentinel-2 pixels. Such pixels are illustrated in figure **??**. Additional information like yield is also attached.

P_t: this describes the observed data (weather and spectral bands) at time t and the location of one pixel.

P: a pixel. We see it as a collection of all the observations at the specified location within one season. More formally, $P := \{P_t | t \text{ is a valid sample time within a defined season}\}$

SCL: scene classification layer. This indicates what one can expect at a pixel at a sampled time. For an overview cf. table 2.2

P^{SCL45}: similar to P but we only consider observations which belong to the classes 4 and 5. This is used done to get a subset of observations which are less contaminated by clouds and shadows.

NDVI: normalized vegetation difference index

DAS: days after sowing

GDD: growing degree days – cumulative sum of $(\text{temperature} - \text{threshold})^+$

XXX ML models and their shortnames

RYEA : relative yield-estimation-accuracy. Definition 4.4.0.1

OOB : out-of-the-box. Describes the procedure if we estimate the value for a point but not consider the point itself.

Chapter 1

Introduction

1.1 XXX motivation - why is it important

- NDVI-timeseries is very simple and widely used. Examples are: - Plant Models REF - Season Start (start of spring) (community name: land-surface-plant-phenology) -

Since satellite images are “for free” researchers extract

1.2 XXX problembaum / fragestellungen

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

1.3 XXX State-of-the-art

zusammenfassung mit literaturrecherche hier:

- Doublelogistic (winter-ndvi)
- parametric / non-parametric approaches
- spatio-temporal approaches

1.4 Roadmap

In chapter

Chapter 2

Problem Description

2.1 Available Data

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

2.1.1 Sentinel 2 Satellite Image Data

General Information

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

Data Description

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

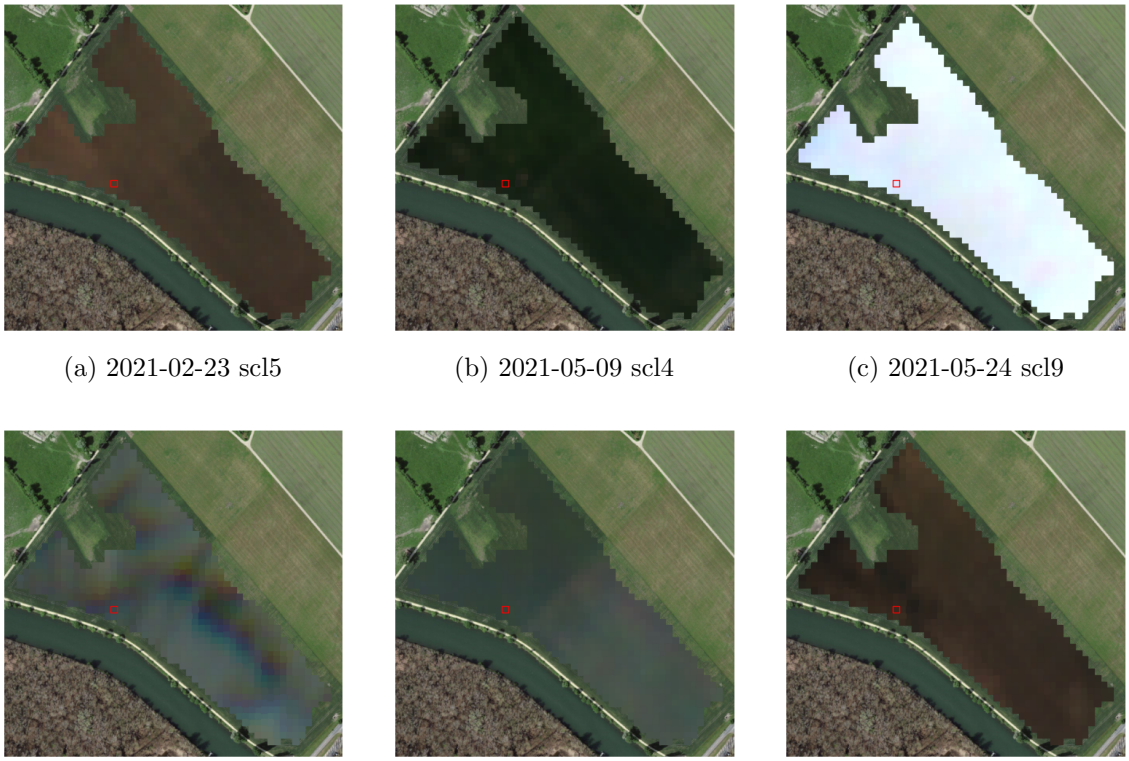
Data Illustration

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

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

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

[3] XXXREF gregor perich "Data prior to March 2018 was only145 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"



(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

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

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

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

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

2.1.2 Yieldmapping Data

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

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

2.1.3 Gather Data

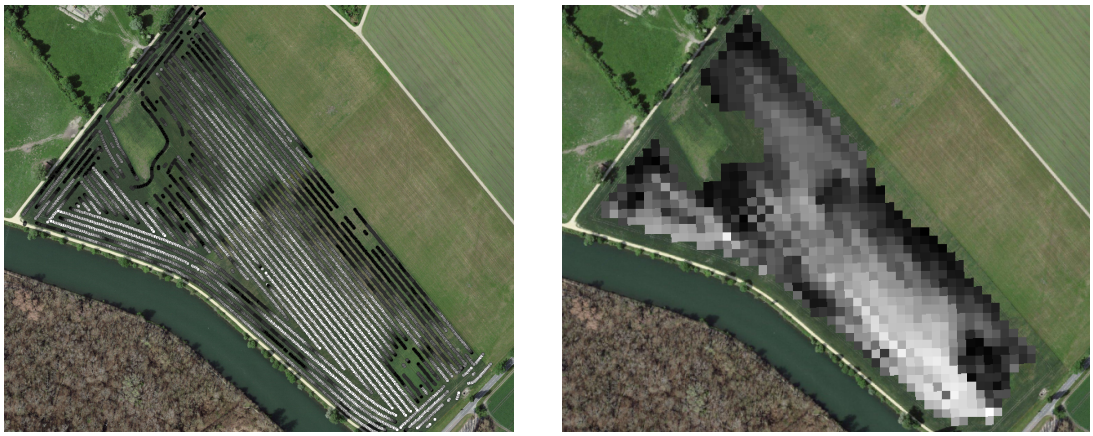
Before we join all the data, we define a few concepts.

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

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

Table 2.2: Overview: Scene Classification Layers (SCL)

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



(a) obtained by a combine harvester (cleaned) (b) rasterized to Sentinel 2 resolution.

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

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

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

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

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

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

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

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

Chapter 3

Interpolation Methods

In this section, we take a closer look at several interpolation methods, which will be used to interpolate and smooth the NDVI time series. A brief overview over the considered interpolation methods can be found in table 3.1.

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

Afterwards, we introduce and discuss each method.

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

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

3.1 DAS vs. GDD

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

3.2 Setting

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

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

where 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

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

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

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

	assumtpions	pros	cons	w	b
Savitzky-Golay 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 itselfe	(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)
Smoothing Splines	– 2cd derivative of function is integrable	– intuitive meaning of penalty – general assumptions – flexible shape	– unbounded	yes	no
B-Splines (Smoothed)	– function can be approximated by a linear combination of B-splines basis functions	– general assumtion – flexible shape	– unbounded – no intuitive meaning for smoothing	yes	no
(Gaussian) Kernel Smoothing		– simple – general assumptions	– bandwidh: failes if there are big data-gaps	yes	yes
Double-Logistic	– function first increases then decreases – ndvi has a minimal value	– good for evergreen plants (if snow masks ndvi) – upper envelope	– parameterestimation can go seriously wrong – strange behaviour for long 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)

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

3.3 Parametric Regression

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

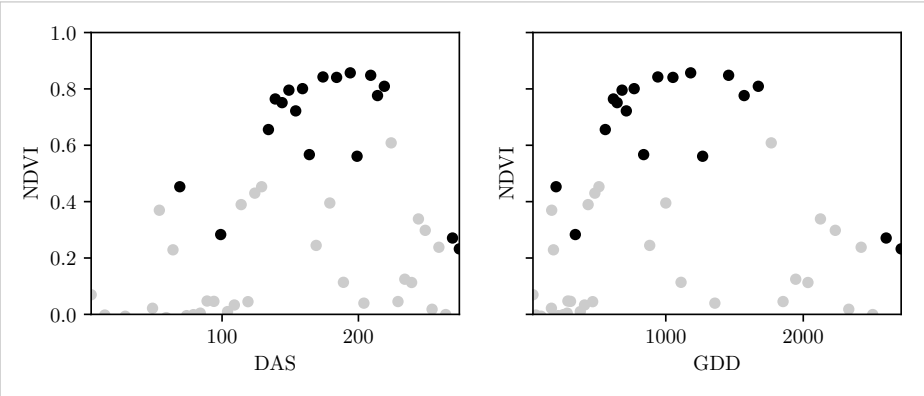


Figure 3.1: The same NDVI timeseries, on the left with DAS as the time scale on the right GDD is the time scale. SCL45 are colord as black. Non-SCL45 (clouds and shadows) are colored in grey.

Optimization Issues

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

3.3.1 Double Logistic

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

Assumptions:

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

— The growth cycle can be divided into an increase and a decrease period, where the time series follows a logistic function. The maximum increase (or decrease) is observed at t_0 (or t_1) with a slope of d_0 (or d_1).

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

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

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

Similar as for the Savitzky-Golay Filter (cf. section 3.4.3) we reestimate (only once) the parameters by giving less weight to the overestimated observations and more weight to the underestimated observations[2].

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

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

Pros	Cons
— Incorporates subject specific knowledge in the case of evergreen plants covered in snow. — Optimized parameters have an intuitive meaning.	— Strong shape assumptions on the NDVI curve. — Parameter optimization might go wrong. This can be mitigated to some extent to provide bounds for the parameters — Strange behavior in regions with little observations. (cf. figure 3.2)

3.3.2 Fourier Approximation

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

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

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

Pros	Cons
— Assumption of periodicity can be helpful if we are modelling multiyear grow cycles — Flexible curve shape	— Bad behavior in regions with little data (cf. figure 3.2) — Hard to interpret estimated parameters — Parameter estimation can go wrong. Introducing bounds can help.

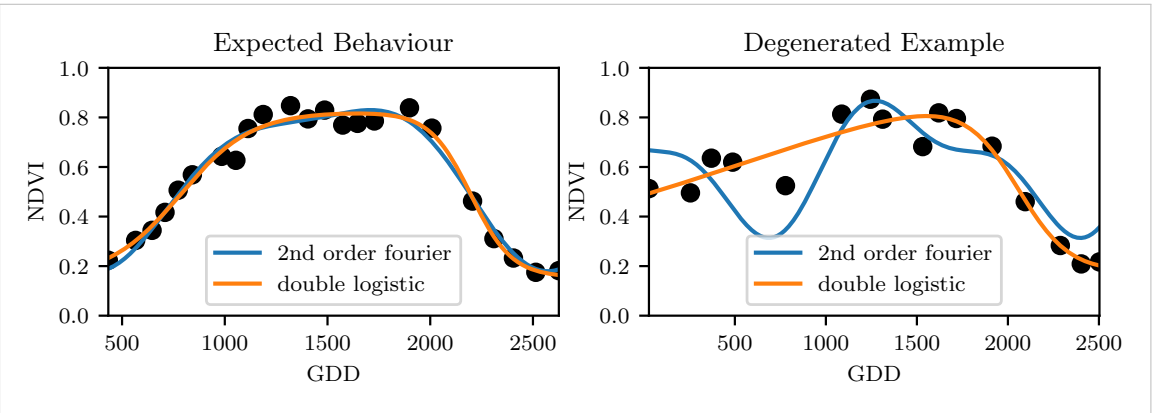


Figure 3.2: Here we observe the nice fitting possibilities of the two parametric methods but notice also some misbehavior

3.4 Non-Parametric Regression

TODO: include Weighted versions

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

3.4.1 Kernel Regression

As described previously, we would like to estimate

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

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

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

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

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

Common choices for the kernel are the normal function or a uniform function (also called "box"function.). Note that we still need to choose the bandwidth of the function (in the case of the normalfunction this is σ the standarddeviation). For local adaptive bandwith-selection we refer to Brockmann, Gasser, and Herrmann (Brockmann et al.).

Pros	Cons
— flexible due to different possible kernels	— if the $x \mapsto K(x)$ is not continuous, $\hat{m}$ isn't either
— can be assigned degrees of freedom (trace of the hat-matrix)	— choice of bandwidth, especially if x_i are not equidistant.
— estimation of the noise variance $\hat{\sigma}_\varepsilon^2$ (XXX cf. CompStat 3.2.2)	

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

3.4.2 Kriging

Kriging was developed in geostatistics to deal with autocorrelation of the response variable at nearby points. 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. For this subsection, we will follow Diggle and Ribeiro (dig).

Definitions and Assumptions

Definition 3.4.2.1. *(Gaussian Process) 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* $\gamma(t, t') = \mathrm{Cov}(S(t), S(t'))$

Assumption 1. 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 $\gamma(h)$.[3]

[3] 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.

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

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

And decide to use a Gaussian Variogram defined by

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

where h is the distance, n is the nugget, r is the range and p is the partial sill visualized in figure 3.3.[4]

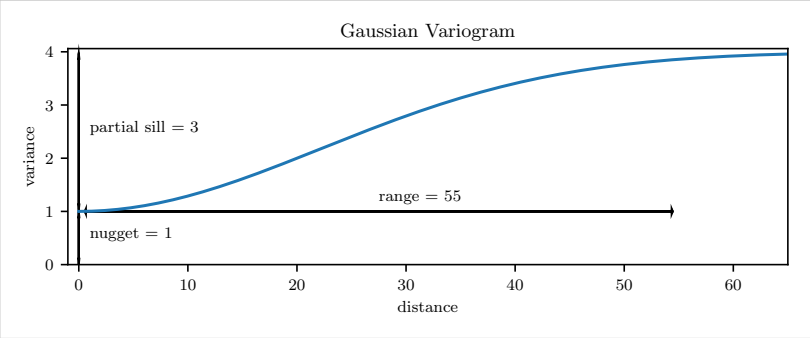


Figure 3.3: Gaussian Variogram with nugget=1, partial sill=3, range=55

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

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

Pros	Cons
— 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.	— Skewness of errors is not taken into account.

3.4.3 Savitzky-Golay Filter (SG Filter)

The *Savitzky-Golay Filter*, introduced in Savitzky and Golay (Savitzky and Golay) is a technique in signal processing and can be used to filter out high frequencies (low-pass

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

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

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

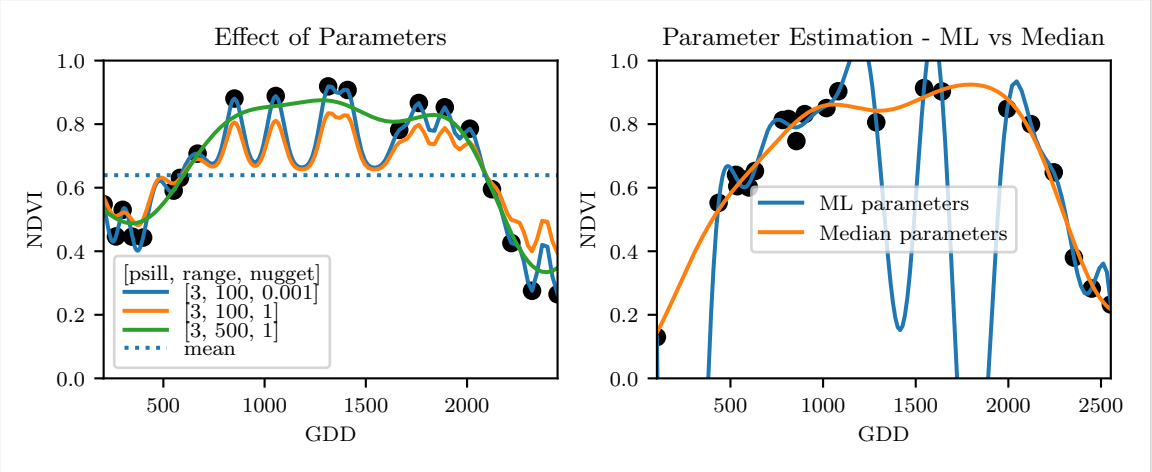


Figure 3.4: On the left, we see how the interpolation change if we increase the nugget and the range parameter. On the right we compare two kriging interpolations, where one takes parameters by numerically maximizing the (which results in a very small nugget) and the other takes the median of many such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

filter) as argued in Schafer (Schafer). Furthermore, it also can be used for smoothing by filtering high frequency noise while keeping the low frequency signal. First, we choose a window size m. Then, for each point, $j \in \{m, m+1, \ldots, n-m\}$ we fit a polynomial of degree k by:

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

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

For equidistant points this can efficiently be calculated by

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

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

Adaptation to the NDVI

In the rather famous paper of Chen, Jönsson, Tamura, Gu, Matsushita, and Eklundh (Chen et al.) a "robust" method based on the Savitzky-Golay has been used. The method is based on the assumption that due to atmospheric effects the observed NDVI tends to be underestimated and that it cannot increase too quickly[7]. Their proposed algorithm is:

i.) Remove points which are labeled as cloudy.

ii.) Remove points which would indicate an increase greater than 0.4 within 20 days.

iii.) Linearly interpolate to obtain an equidistant time series X^0.

iv.) Apply the Savitzky-Golay Filter to obtain a new time series X^1.

[7]The latter is argued by the biological impossibility of such fast vegetation changes

v.) Update X^1 by applying again a Savitzky-Golay 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[8] and by repeating this step we approach the "upper NDVI envelope".

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

Extension: Spatial-Temporal-Savitzky-Golay Filter

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

3.4.4 Locally Weighted Regression (LOESS)

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

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

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

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

How does the Robust LOESS differ from 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

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

[9] 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

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

$\lceil \alpha n \rceil < d+1$.[10] 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.

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

3.4.5 B-splines

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

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

where B are basisfunctions and recursively defined by:

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

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

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

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

3.4.6 Natural Smoothing Splines

Let $\mathcal{F}$ be the Sobolev space (the space of functions of which the second derivative is integrable) . Then the unique[12] minimizer

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

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

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

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

Pros	Cons
— 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). — Performs also well if points are not equidistant. — Fixes the Runge's phenomenon (fluctuation of high degree polynomial interpolation).	— Choose λ.

3.5 Tuning parameter estimation

Viele der in sektion 3.3 und 3.4 eingeführten interpolationsmethoden beinhalten einen freien parameter. Um diesen für eine spezifische interpolationmethode auszuwählen werden wir mithilfe OOB schätzung die absoluten residuen schätzen und dann anhand einer statistik den parameter optimieren. Wir klären die vorgehensweise schritt für schritt: - konstruiere eine menge Λ von kanditat-parameter, welche den parameterraum großzügig abdeckt. - betrachte $\mathcal{P}$, eine Menge von pixeln - für jeden parameter $\lambda \in \Lambda$ betrachte die einzelnen pixel und berechne die leave-one-out-cross-validation ($LOOCV$) für den NDVI, mit der spezifischen interpolationsmethode. D.h. $\hat{y}_i$ erhalten wir durch die interpolation von den punkten $\{(x_j, y_j) | j \neq i\}$. Anschließend berechne die absoluten residuen $|y_i - \hat{y}_i|$ und definiere die Menge R_λ als die menge aller residuen (aller pixel in $\mathcal{P}$). - bestimme $\lambda_{optimal} = \arg\min_{\lambda \in \Lambda} \text{Quantile}(90)(R_\lambda)$, wobei wir mit Quantile(90) das 90% quantil beschreiben.

Wir haben uns für Quantile(90) entschieden als optimierungsfunktion, wir zum einem einen anteil von 10% an ausreißern erlauben wollen (korrupte punkte) aber zum anderen in 90% der fälle einen akkuraten fit anstreben.

In der Abbildung 3.5 sehen wir beispielhaft wie die interpolationen aussehen wenn wir zur parameterbestimmung andere quantile benutzen. Zusammengefasst kann man sagen, dass je höher das quantil desto stärker auch die glettung.

3.6 Robustify

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 which have high residuals after the initial (or if we reiterate, the last) fit.

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

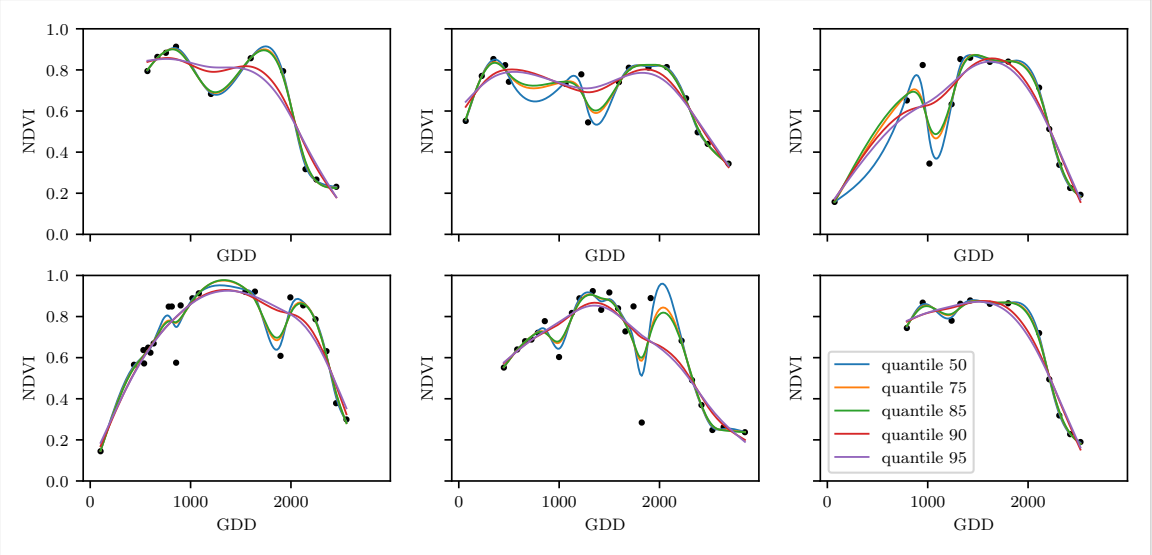


Figure 3.5: Smoothing splines fit with smoothing parameter optimized by minimizing the "..."-quantile of the absolute leave-one-out residuals. Note that the larger the considered quantile is, the smoother the resulting curve becomes.

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

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

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

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

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

and interpolate again using the new weights. We can iterate this reweighting and stop after several steps or when 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 of 50%.[14]

3.6.1 Our Adjustment:

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

[14] The breakdownpoint 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 downweighted more and more in each iteration (because for the next iteration the resiudal of the outlier will be even larger, since we gave less weight to it).

median. This can be defined as

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

for $r, w \in \mathbb{R}^n$. [15]

3.6.2 Examples and Conclusions

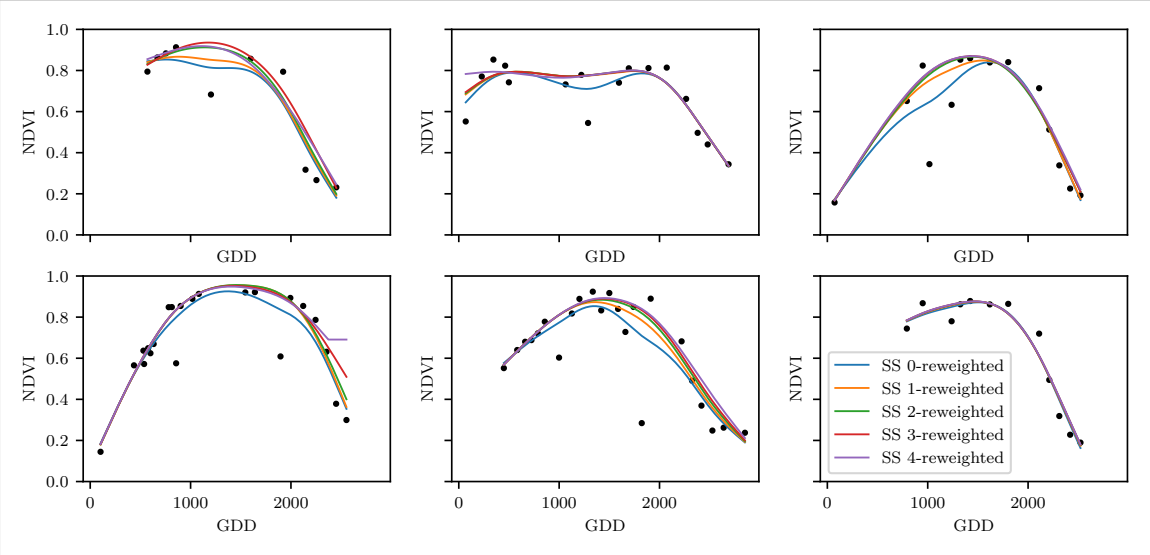


Figure 3.6: Smoothing Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.6) are also displayed

In abbildung 3.6 sehen wir exemplarisch für 6 pixel wie die mit smoothing splines interpolierte NDVI zeitreihe nach $0, 1, 2, 3, 4$ iterationen aussieht (für die analogen abbildungen der andere interpolations methoden verweisen wir auf den Anhang: A.1, A.2, A.3 and A.1).

In der tat beobachten wir, wie die interpolierte zeitreihe nach jeder interation weniger durch ausreißer beinflusst wird. Den grössten unterschied beobachten wir bei der ersten iteration. Ausserdem sehen wir im plot unten links wie die Interpolation dem rechten endpukt mit jeder weiteren iteration "entflieht", obwohl unsere intuition diesen punkt nicht unbedingt als ausreißer identifiziert. Daher werden wir im folgenden immer nur eine iteration durchführen und dannach aufhören.

3.6.3 Upper Envelope Approach - Penalty for negative resiudals

Wenn wir in 3.6.0.1 die negativen residuen durch multiplikation (z.B. faktor 2) künstlich erhöhen, so werden die korrespondierenden punkte in der nächsten iteration weniger gewicht erhalten. Dies ermöglicht uns eine interpolation zu erzeugen, welche einer oberen hülle gleicht. Diese obere hülle kann man sich intuitiv als ein tuch vorstellen, welches man von oben auf die punkte legt.

[15] This adjustment is also necessary to keep the scale estimation meaningful during the iterations.

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

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

Dieses vorgehen beruht der annahme , dass wir den NDVI tendenziell unterschätzen (wie in REF-savitzky-golay). Da wir eine generelle methode entwickeln wollen, welche prinzipiell nicht and den NDVI gekoppelt ist, werden wir diesen Ansatz nicht weiter verfolgen.

3.7 Performance Assecement

Nun wollen wir die verschienden interpolationsmethoden einmal mit und einmal ohne robustifizierung benchmarken. Dafür werden wir die gleiche technik verwenden, wie wir sie bei der parameterbestimmung in sektion . Auf B_λ wenden wir den RMSE und verschiedene quantile an und stellen die Ergebnisse in tabelle 3.2 dar.

3.8 XXX Evaluation

– ss dominate (i.e. have better benchamrk values w.r.t. all considered statistics) b-splines (robustified and non-robustified)

– dl dominate fourier (robustified and non-robustified)

– loess slightly dominates ss, but we prefere ss because of the smoothnes guarantees (compare the figures A.1 and 3.6).

TEMP — Figures

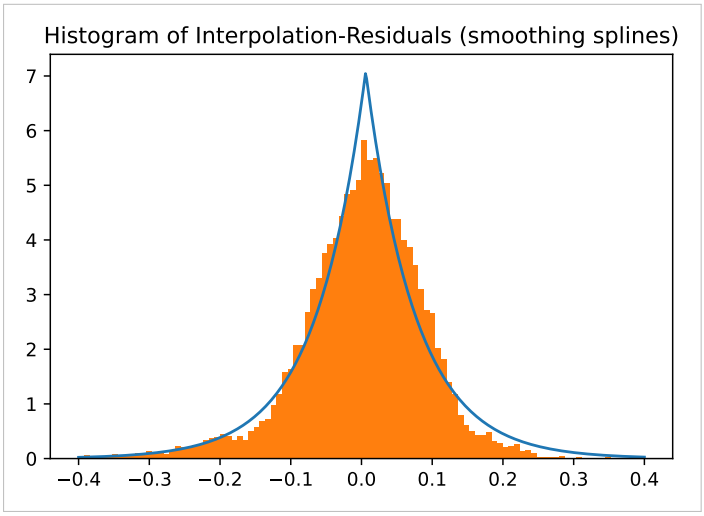


Figure 3.7: XXX caption XXX

Chapter 4

NDVI Correction

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

4.1 Considering other SCL Classes

In figure 4.1 we notice that some blue points[1] follow the interpolated line closely and that they might be useful in improving an interpolation fit.

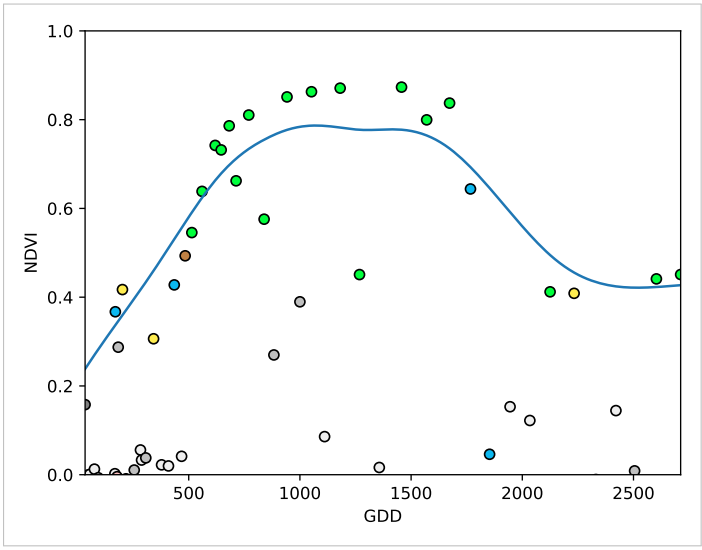


Figure 4.1: A smoothing splines fit considering green and yellow points (SCL45)

To get an impression whether there is some useful information contained in the remaining SCL-classes (all except 4 and 5) 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:

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

Assumption 1. The true NDVI value at time t can be successfully estimated by out-of-bag interpolation using high quality observations. That is the interpolated value (using an 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 hope to recover infomation from SCL-classes other than 4 and 5. For that, we will check for the other SCL-classes if there is a relation between the "true"-NDVI[2] 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.

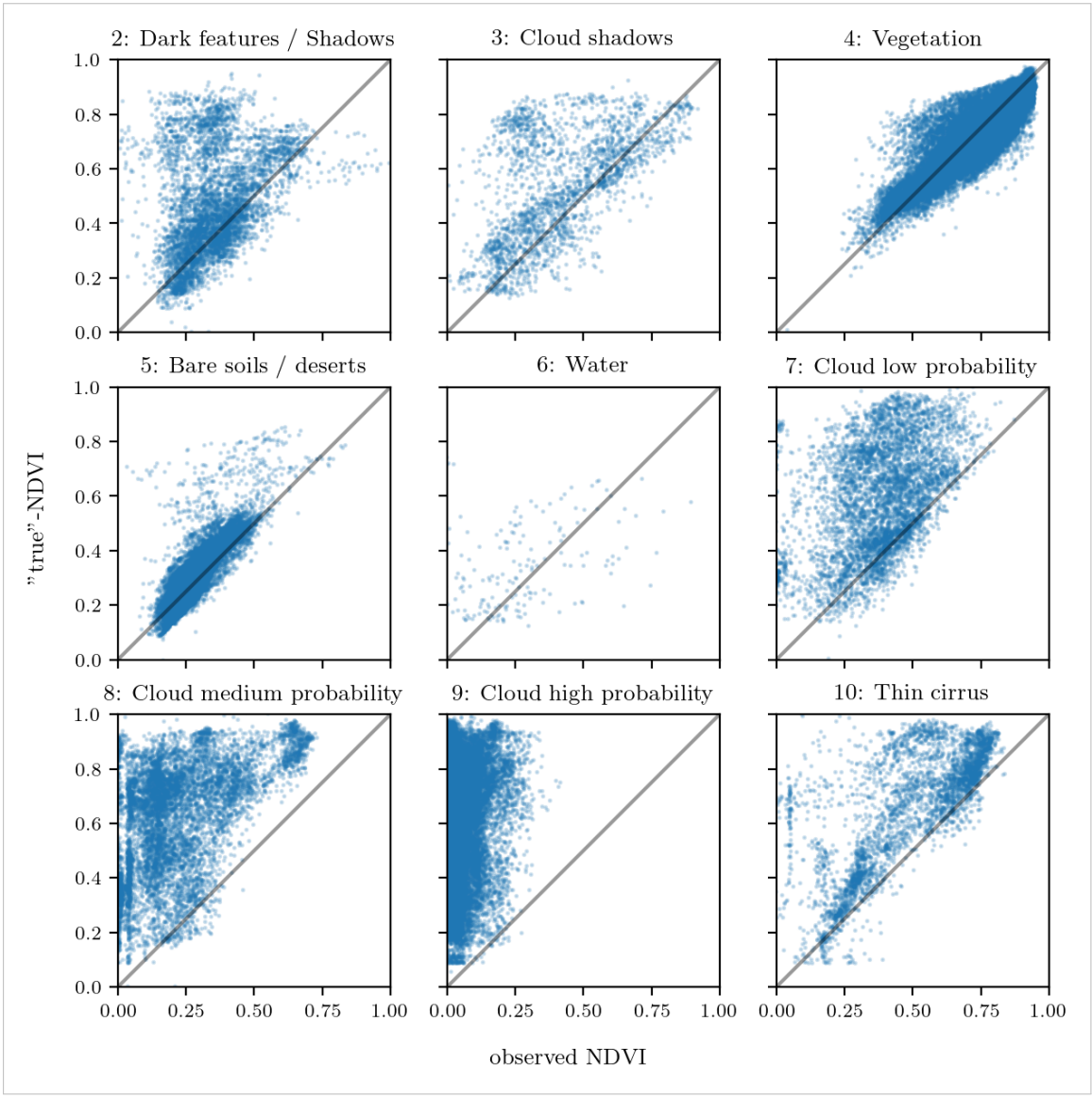


Figure 4.2: For each SCL class, we compare the true NDVI with the observed NDVI. (The true NDVI was estimated with OOB smoothing splines and we used all observations of 10% of the total training pixels.)

[2] i.e. the out-of-bag (OOB) estimate using smoothing splines

It might be tempting to include some of the above 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 shall try to correct the observed NDVI and estimate the uncertainty of each correction.

4.2 Correction

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

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

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

4.2.1 Response and Covariates

For training a NDVI correction model, we need ground-truth (response) and informative covariates. We organize those in a table, where each row corresponds to a P_t (i.e., a pixel at a time t). For the response we will again use the assumption 1. There is no canonical answer to the question 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 for others to understand and use. Therefore, in the subsequent we will only take the spectral data of the satellite and the observed NDVI derived from it as covariates[3].

4.2.2 Correction Methods

In the following, we will introduce different modelling approaches, which we will use to model the relation between the response $y = y_{\text{true OOB NDVI}} \in \mathbb{R}^n$ and the covariates encoded in the design matrix[4] $X \in \mathbb{R}^{n \times p}$. Furthermore, we will use the matlab ':' notation to indicate rows and collumns of a matrix (e.g. $X:, 3$ is the 3rd collumn of X).

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

Ordinary Least Squaers (*OLS*)

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

[3] We do not mention the intercept explicitly, but it will also be included.
[4] This is the Matrix which contains of all covariates.
[5] Strictly speaking since SCL-classes are dummy variables

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

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

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

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

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

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

LASSO

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

$$\hat{\beta}_\lambda = \underset{\beta \in \mathbb{R}^p}{\arg\min} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 = \underset{\beta \in \mathbb{R}^p \text{ and } \|\beta\|_1 < \lambda}{\arg\min} \|y - X\beta\|_2^2.^{7} \tag{4.2.2.1}$$

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 continious and convex.

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

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 $RMSE_\lambda$ [8] (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 4.2.1 and their second degree of interactions.[9]

Random Forest (*RF*)

To define a random Forest introduced by Breiman (Breiman) 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

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

[8] The cross-validatet Root Mean Square Error is the mean of the RMSE's obtained for each fold (using the model trained on the remaining folds). We use the following definition of the $RMSE$: $\sqrt{\sum_{i=1}^{n}(y - \hat{y})^2/n}$

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

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

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[10] decreasing the RMSE[11] each time. If the set we want to split contains less then a certain amount of training points we stop.

To build a *Random Forest* we will bootstrap-aggregate[12] many such Trees[13]. The prediction of the Random Forest for a new point x is then the mean of the predictions from all the Trees.

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

Multivariate Adaptive Regression Splines (*MARS*)

REFFriedman (Friedman)

A MARS model 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.

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

[11] 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).

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

[13] 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.

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 canditate functions-pairs

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

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 generalized cross validation score (GCV).[14]

To reduce computational complexity, we follow the recommendation from REFleaps wrapper (leaps wrapper) and restrict h in equation (4.2.2.2) 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.

Pros	Cons
— 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).

General Additive Model (GAM)

GAMs as described in Hastie and Tibshirani (Hastie and Tibshirani) 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:

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

To estimate the non-parametric functions we can use smoothing splines (ref sec. 3.4.6). 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 simultanously we will use a strategy named backfitting. We basically cycle through

[14] This means that we perform an iterative procedure to reduce the number of functions in $\mathcal{M}$. For every function h in $\mathcal{M}$ we compute the model using $\mathcal{M}$
$\{h\}$. We discard the function which – when excluding from $\mathcal{M}$ – leads to the best GCV score.

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

the indicies $1, \dots p$ and refit $\hat{g}_j$ each time. The following illustrates the procedure:

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

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

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

4.2.3 Uncertainty Estimation

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

We do this by replacing the response 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 $\hat{v}$.

4.2.4 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 unreliable observations. Thus, we define the weightfunction:

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

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.

4.3 Resulting Interpolation Stragegies

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

i.) OOB Interpolation (+ robustify?)

ii.) Correction

iii.) Uncertainty estimation

iv.) Interpolation (+ robustify?)

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

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.

As it is not feasible to try every possible combination, we make the following restrictions of 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.

4.4 Evaluation Method

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

Definition 4.4.0.1. *(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*[17]*. We define the RYEA as the relative RMSE in yield estimation. Formally expressed:*

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

4.4.1 Idea

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.

4.4.2 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. For this we will use the following statistics:

— Maximum slope

[17] We will use the matrixes derived in section 4.4.2

- Minimum slope
- Integral[18] over all
- Peak (i.e. maximal NDVI)
- Peak GDD (i.e. value at which the peak is attained)
- Integral[18] up to the peak
- Integral[18] after peak
- Integral[18] from 0-685 GDD
- Integral[18] from 685-1075 GDD

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

As a result, we obtain for each interpolation strategy a matrix in which each row corresponds to a pixel and contains both the yield and the characterizing statistics. Using this matrix, we train a random forest[19] for yield estimation, and compute the integrated OOB estimates[20] $\hat{y}$. Finally, for each interpoaltion strategy, we calculate the RYEA. The results are shown in table 5.1.

[18] We will only consider the integral of the function $max(0, NDVI - 0.3)$, where 0.3 is assumend to be a minimal NDVI value. REF

[19] The choice of the modelling approach does not matter to 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.

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

Chapter 5

Results

5.1 XXX small recap from "Interpolation Methods"

5.2 Robustification and NDVI-Correction

Table 5.1: XXX RMSE of yield prediction

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

Chapter 6

Discussion

High RMSE in ...: How much can we expect to get? We have multiple sources of uncertainty in the data:
1. Uncertainty in Yield data collected by the combine harverster
2. Uncertainty in Yield data through rasterization
3. Uncertainty in satellite images through "measurement errors" introduced via clouds and other atmospheric effects
4. Uncertainty introduced by interpolating (especially when long data-gaps are present)

Bootstrap in NDVI correction

Müssen wir test und trainingsdaten strikt nach jahr trennen? Zwar könnten wir damit berwerten, ob unser modell ein allgemeines muster gelernt hat oder nur die gegebenen jahre gelernt hat. Wir haben jedoch an keiner stelle (bis der evaluation) irgendwelche groundtrouth benutzt. Anstadt dessen haben wir den "true"-NDVI mit der Annahme 1 per OOB geschätzt. Somit haben wir bootstrap-mäßig an den haaren aus den dem problem gezogen. Daher folgern wir, dass wir unsere methode auf einen neuen (vergleichbaren) datensatz anwenden können und die Korrektur wiedermals über diesen bootsrap lösen können.

6.1 NDVI Correction

6.1.1 Shall We Use Additional Covariates?

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

In section 4.2.1 we have only used the spectral data (and the observational NDVI calculated from them) as covariates. Since we have the weather data available (cf. 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 weather in our study region to be rather homogeneous [1]. 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

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

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].

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

[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

Outlook

7.1 Data

— Method how data has been extrapolated to the grid could possibly be improved

— For computational reasons we mostly considered all years and split the data (on the pixel level) randomly into a train/test set. A cross Validation with leaving one year out would be

7.2 Interpolation

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

7.3 NDVI Correction

— try different link functions in section 4.2.4 between estimated absolute resiudals and weights

7.4 NDVI Correction + +

In sektion

— NDVI Correction can be applied to all sorts of land observed via. satellites (without the need of ground truth data)

— The idea of NDVI Correction could be applied to other spectral indices like the Green Leaf Area Index.

— Yield is not the only target variable of interest. Other variables like protein content could also be used in section ... for the method evaluation.

Bibliography

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

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

Breiman, L. Random Forests. *45*(1), 5–32.

Brockmann, M., T. Gasser, and E. Herrmann. Locally Adaptive Bandwidth Choice for Kernel Regression Estimators. *88*(424), 1302–1309.

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

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

Cleveland, W. S. Robust Locally Weighted Regression and Smoothing Scatterplots. *74*(368), 829–836.

Friedman, J. H. Multivariate Adaptive Regression Splines. *19*(1), 1–67.

Hastie, T. and R. Tibshirani. Generalized Additive Models: Some Applications. *82*(398), 371–386.

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

leaps wrapper, S. M. D. f. m. b. T. H. a. R. T. U. A. M. F. u. w. T. L. Earth: Multivariate Adaptive Regression Splines.

Lyche, T. and K. Morken. Spline Methods.

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

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

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

Tibshirani, R. Regression shrinkage and selection via the lasso: A retrospective. *73*(3), 273–282.

Appendix A

XXX Appendix

Interpolation

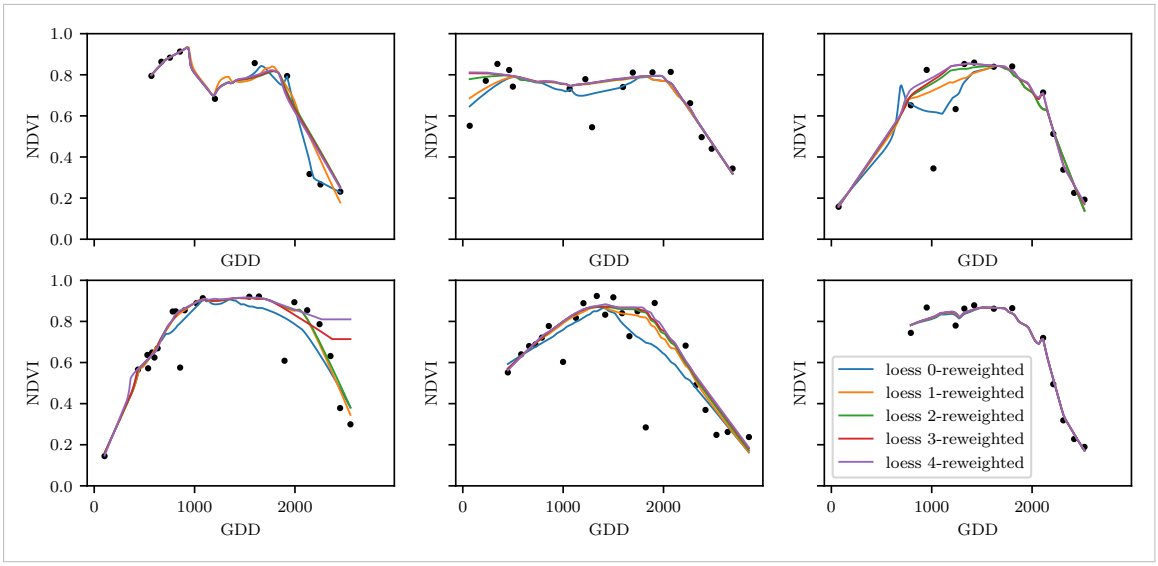


Figure A.1: The LOESS smoother fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.6) are also displayed

A.1 NDVI correction

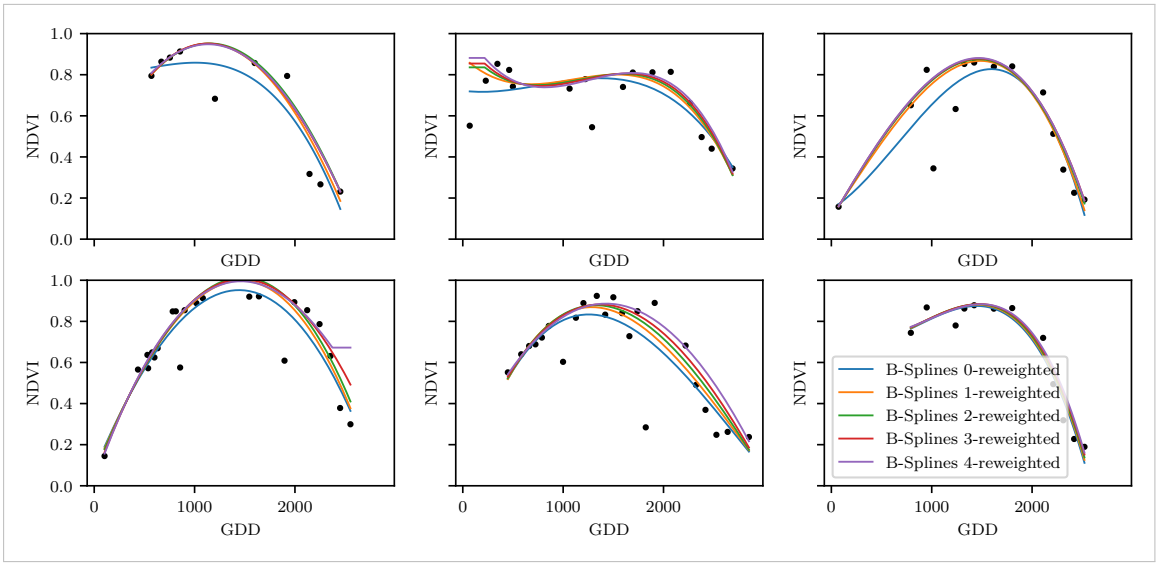


Figure A.2: B-Splines fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.6) are also displayed

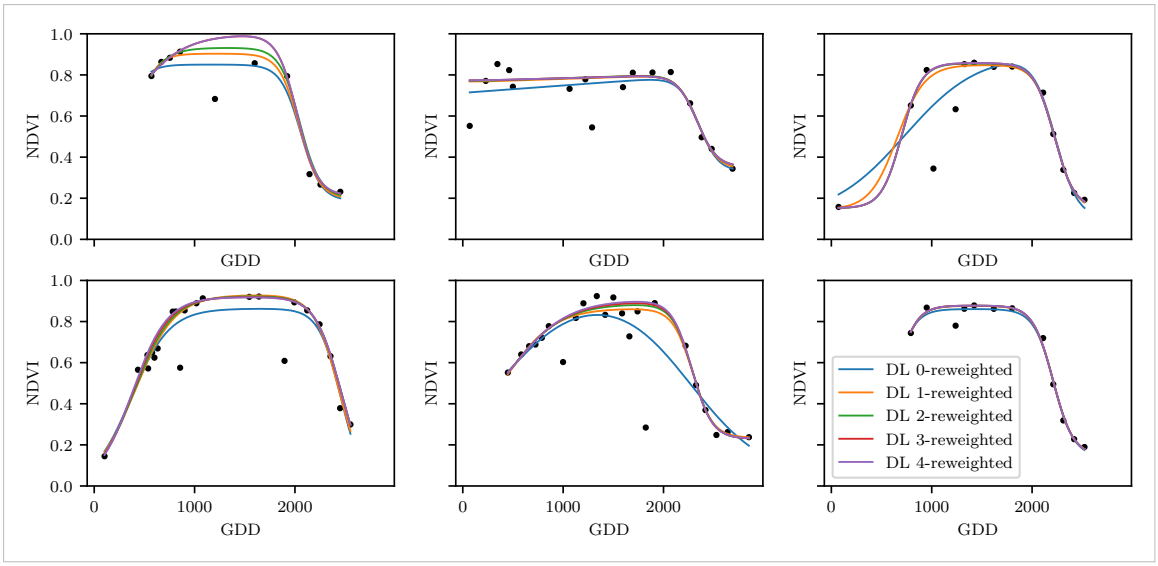
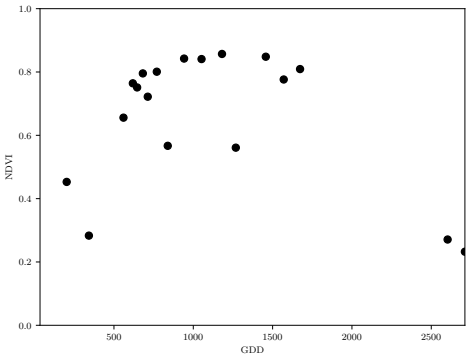
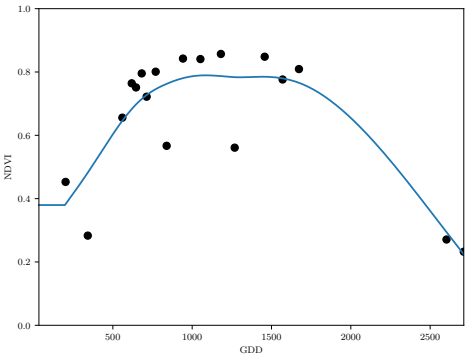


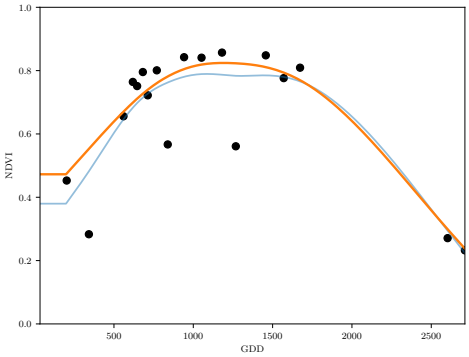
Figure A.3: A Double Logistic curve fitted to different (SCL45) NDVI time series. Iterations of a robustifing refit (as indicated in section 3.6) are also displayed



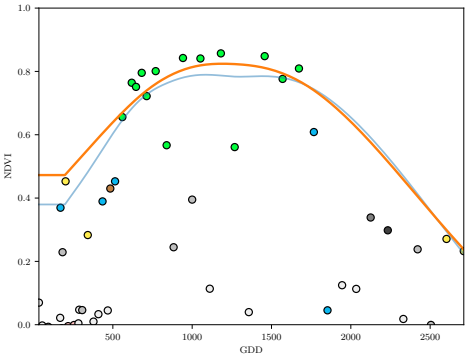
(a) NDVI time series with scl45



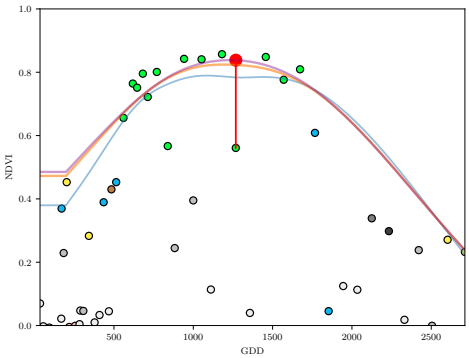
(b) Interpolation via Smoothing Splines



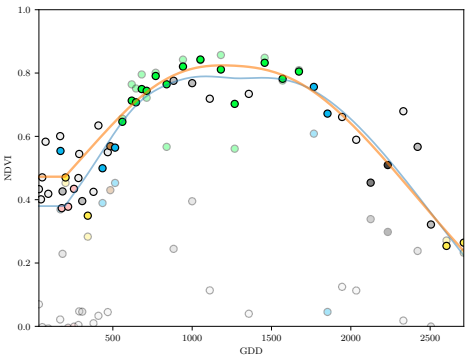
(c) Robustly reweighted fit



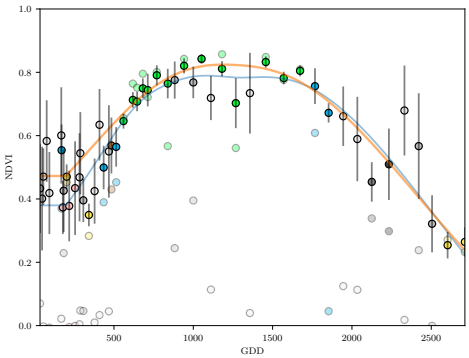
(d) Now also consider other SCL-classes



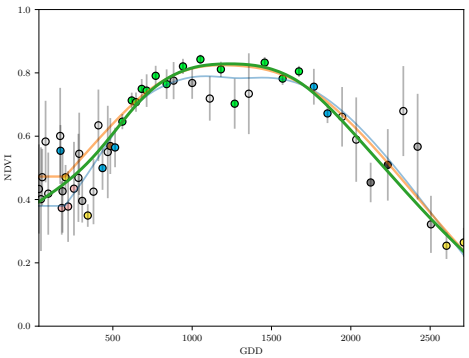
(e) OOB estim. for each point using SCL45



(f) Correct NDVI



(g) Estimate absolute errors



(h) Robust interpolation on weights derived from uncertainties.

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