Contents

Notation

c: a (vector of) constant(s)

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

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

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

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

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

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 ??.

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

P: a pixel. More formally we see it as a collection of all the observations at the specified location within one season.

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

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 (temperature – threshold)$^+$

Chapter 1

Problem Description

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)	

Figure 1.1: XXX scl classes

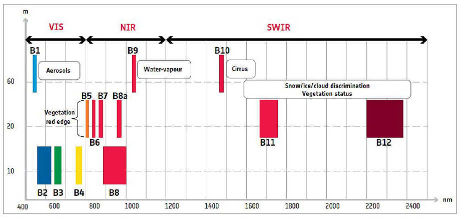

Figure 1.2: XXX scl classes

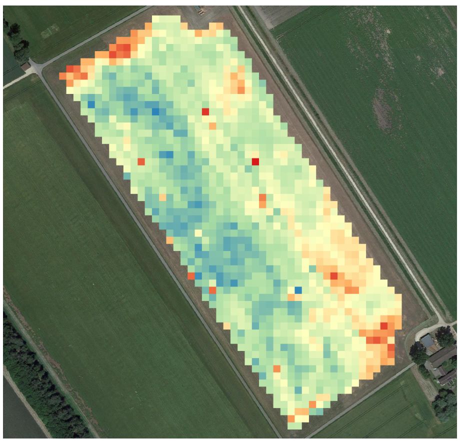
Figure 1.3: XXX yield raster

1.1 DAS vs GDD

XXX

Chapter 2

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.

First, we give a brief overview in table 2.1.

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

Later, 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.

2.1 Setting

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

$$Y_i = m\left(x_i\right) + \varepsilon_i,$$

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

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

Different assumptions on m will lead to the following methods:

2.2 Robustify

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

Even though the procedure is taken from the robust version of the LOESS smoother (c.f. section 2.4.4 and Cleveland (Cleveland)), we discuss it now because we will apply it also to other interpolation methods.

	assumtpions	pros	cons	weights	bounded
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	mostly
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)	mostly
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	mostly
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		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	mostly
Universal Kriging	– function is a realization of a stationary gaussian process	– informative parameters – flexible	– regression to the mean – assumptions clearly not met	yes	mostly

Table 2.1: A short summary of the studied interpolation methods

XXX[1]

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:

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

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

$$w_i^{\text{new}} := w_i^{\text{old}} \, \mathrm{B}\left(\frac{|r_i|}{6 \,\mathrm{mad}\left(r_1, \ldots, r_n\right)}\right) \tag{2.2.0.1}$$

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

Examples of such iterative fits are illustrated in the figures 2.4 2.5, 2.6, 2.4 and 2.7.

2.2.1 XXX Our Adjustment:

Since we usually observe outliers with negative residuals we decide to divide the negative residuals by two before updating the weights. Furthermore, we want to prevent low-weighted observations to corrupt our estimation of scale (the median) and thus we use the weighted median. This can be defined as

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

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

2.3 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.

Optimization Issues

Now we try to minimize the residuals sum of squares over 5 (or 6) parameters. This is a non-convex optimization problem, and thus the algorithm[2] either could not find the global minimum or failed to converge. This was fixed by providing for each parameter reasonable initial values and generous bounds (which match our experience).

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

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

2.3.1 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:

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

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 2.1.

Similar as for the Savitzky-Golay Filter (c.f. section 2.4.3) we reestimate (only once) the parameters by giving less weight to the overestimated observations and more weight to the underestimated observations[3].

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 2.1)

2.3.2 Fourier Approximation

Similar as in section 2.3.1 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 $\Phi = 2\pi \times (t-1)/n$.

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

Pros	Cons
— Assumption of periodicity can be helpful if we are modelling multiyear grow cycles	— Bad behavior in regions with little data (cf. figure 2.1)
— Flexible curve shape	— Hard to interpret estimated parameters
	— Parameter estimation can go wrong. Introducing bounds can help.

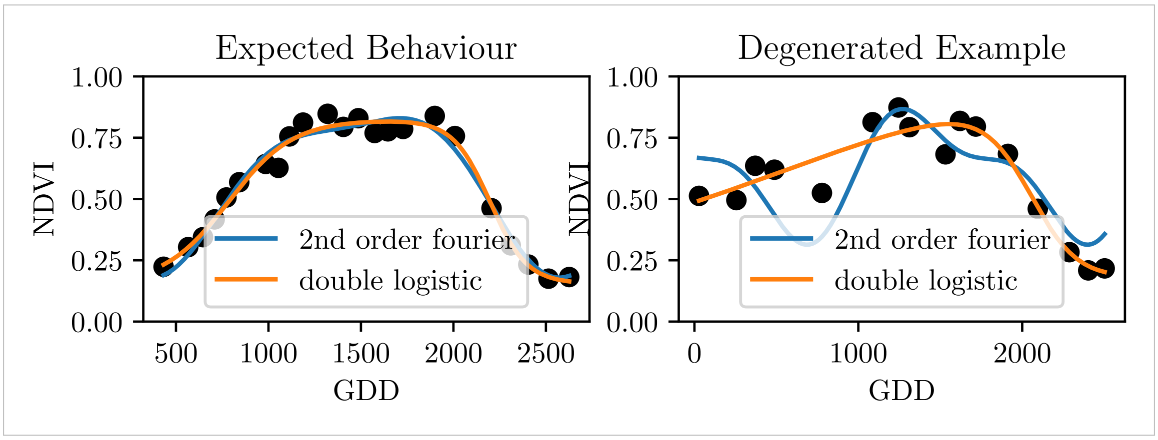

Figure 2.1: Here we observe the nice fitting possibilities of the two parametric methods but notice also some misbehavior

2.4 Non-Parametric Regression

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.

2.4.1 Kernel Regression

As described previously, we would like to estimate

$$\mathbb{E}[Y \mid X = x] = \int_{\mathbb{R}} y f_{Y|X}(y \mid x) dy = \frac{\int_{\mathbb{R}} y f_{X,Y}(x,y) dy}{f_X(x)}, \tag{2.4.1.1}$$

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 2.4.1.1 we arrive at the *Nadaraya-Watson* kernel estimator:

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

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

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 c.f. CompStat 3.2.2)	

2.4.2 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

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: 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 $\gamma(h)$.[4]

We also define the variogram of a Gaussian process as

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

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 2.2.[5]

Pros	Cons
— It is a well-studied method.	— Regression to the mean.
— 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.

[4]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.

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

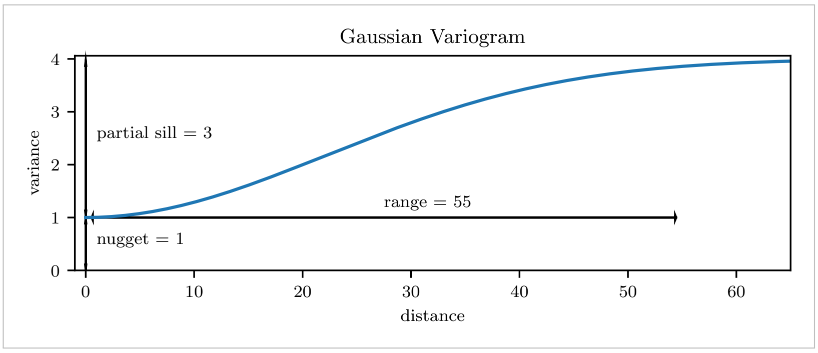

Figure 2.2: Gaussian Variogram with nugget=1, partial sill=3, range=55

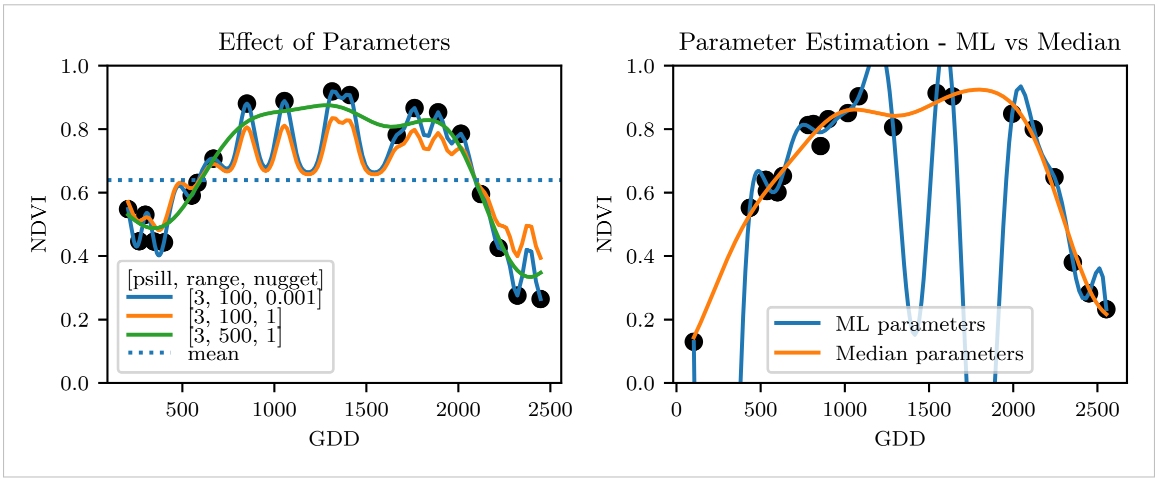

Figure 2.3: On the left, we see how the interpolation change if we increase the nugget and the range parameter. On the right we compare two kriging interpolations, where one takes parameters by numerically maximizing the (which results in a very small nugget) and the other takes the median of many such numerical optimizations.

2.4.3 Savitzky-Golay Filter (SG Filter)

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

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

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

For equidistant points this can efficiently be calculated by

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

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

Adaptation to the NDVI

In a rather famous paper Chen, Jönsson, Tamura, Gu, Matsushita, and Eklundh (Chen et al.) a "robust" method based on the Savitzky-Golay has been used. The method is based on the assumption that due to atmospheric effects the observed NDVI tends to be underestimated and that it cannot increase too quickly[6].

Algorithm:

i.) Remove points which are labeled as cloudy.

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

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

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

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[7] 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[8]. Because we are working with rather high resolution satellite data, and we need the variance in the predictors, we will waive this extension.

2.4.4 Locally Weighted Regression (LOESS)

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

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

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

[8] 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

The Locally Weighted Regression (LOESS) can be understood as a generalization of the Savitzky-Golay Filter (c.f. sec. 2.4.3).

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 $B_{h_i}(x_i)$.[9] 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 $\lceil \alpha n \rceil < d+1$. 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.

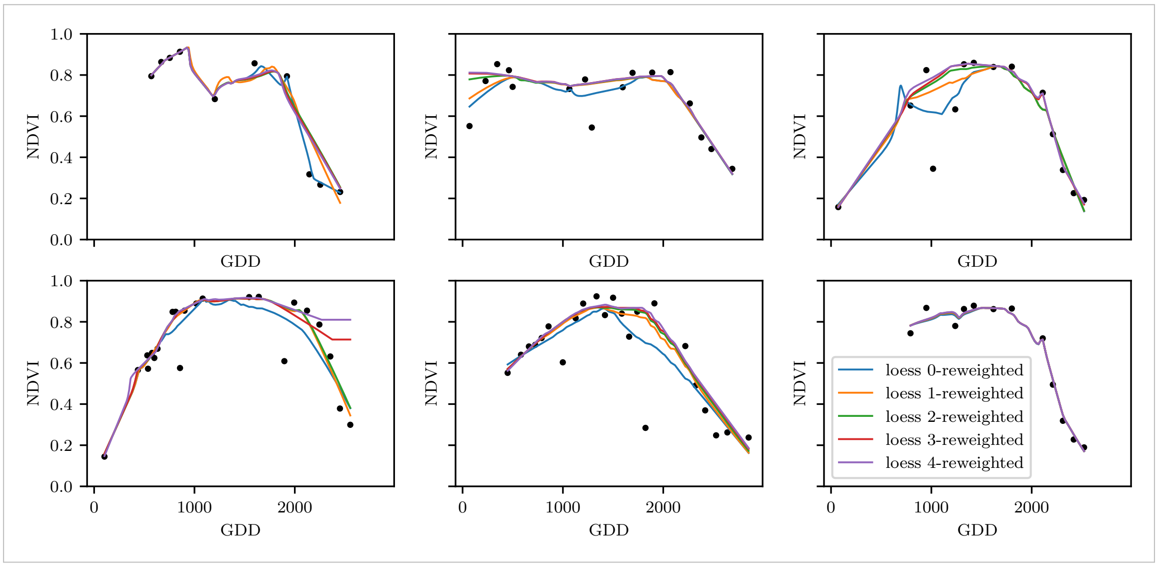

Figure 2.4: The LOESS smoother fitted to different (scl45-)NDVI time series. Iterations of a robustifing refit (as indicated in section 2.2) are also displayed

[9] 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.

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

2.4.5 B-splines

from Lyche and Morken (Lyche and Morken)

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

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

Smoothing: We can relax the constraint that we have to perfectly interpolate. Thus, we use the minimum number of knots[10] such that: $\sum_{i=1}^{n}(w(y_i - \hat{y}_i))^2 \leq s$

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	

2.4.6 Natural Smoothing Splines

Let $\mathcal{F}$ be the Sobolev space (the space of functions of which the second derivative is integrable). Then the unique[11] minimizer

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

is a natural[12] 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.

TEMP — Figures

[10] SciPy uses FITPACK and DFITPACK, the documentation suggests that smoothness is achieved by reducing the number knots used

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

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

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

	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

Table 2.2: 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.

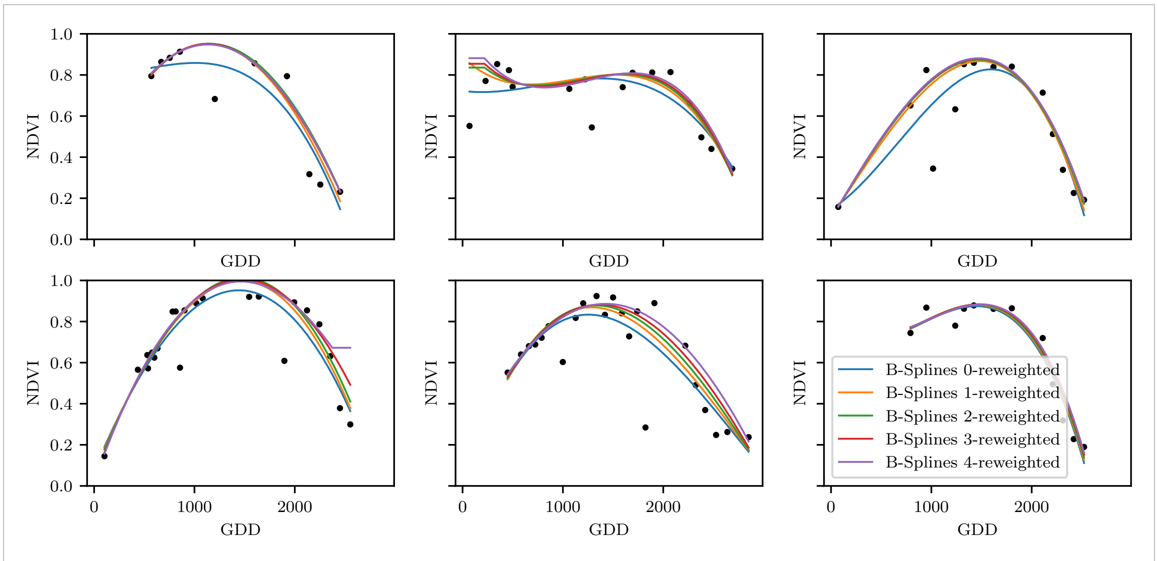

Figure 2.5: B-Splines fitted to different (scl45-)NDVI time series. Iterations of a robustifing refit (as indicated in section 2.2) are also displayed

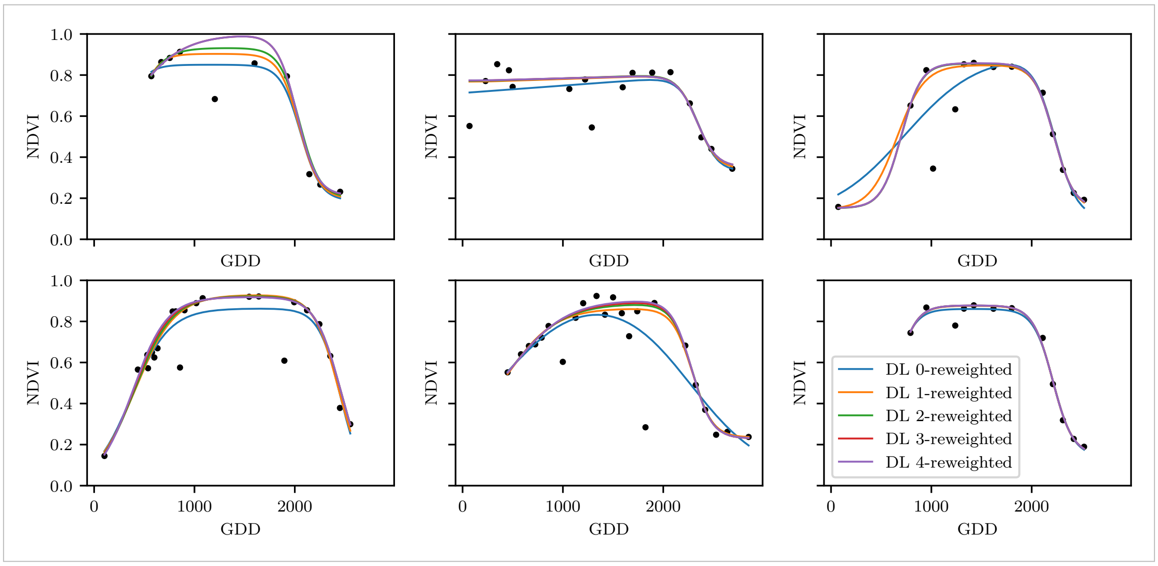

Figure 2.6: A Double Logistic curve fitted to different (scl45-)NDVI time series. Iterations of a robustifing refit (as indicated in section 2.2) are also displayed

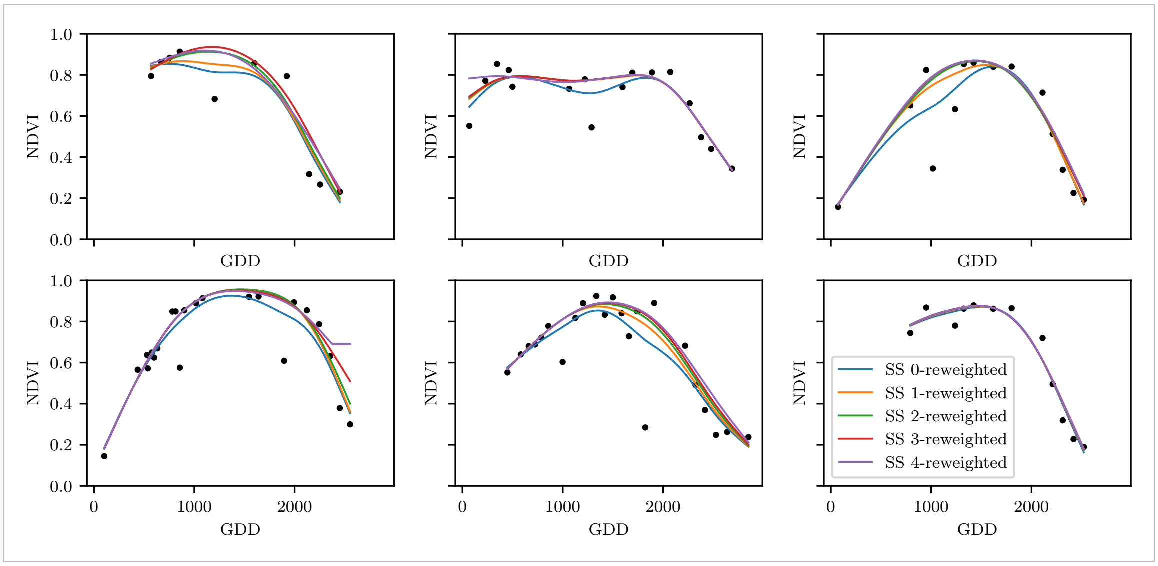

Figure 2.7: Smoothing Splines fitted to different (scl45-)NDVI time series. Iterations of a robustifing refit (as indicated in section 2.2) are also displayed

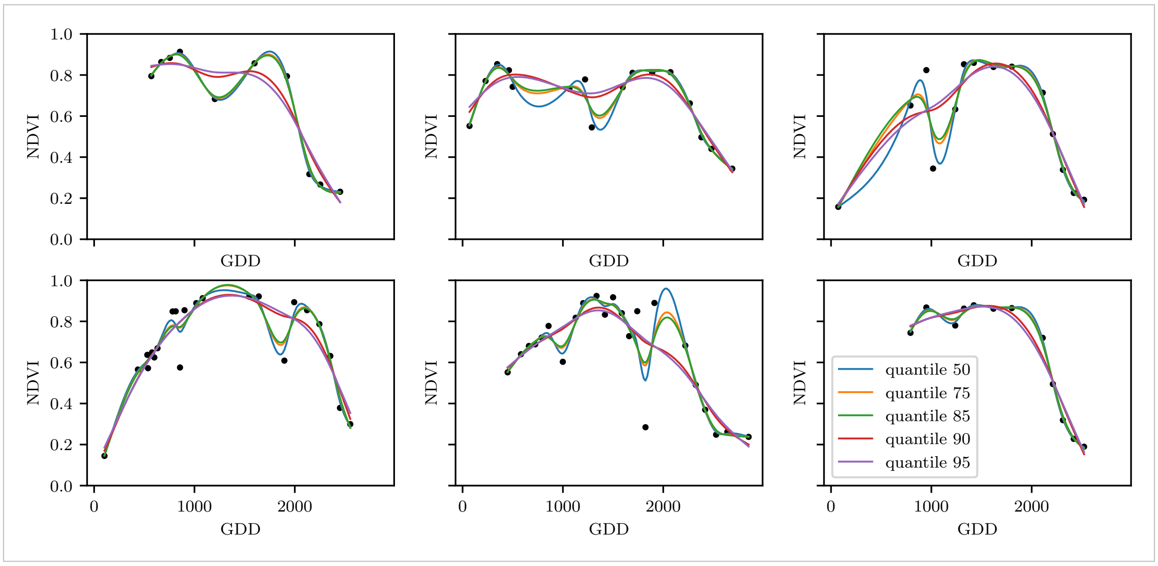

Figure 2.8: Smoothing splines fit with smoothing parameter optimized by minimizing the "…"-quantile of the absolute leave-one-out residuals. Note that the larger the considered quantile is, the smoother the resulting curve becomes.

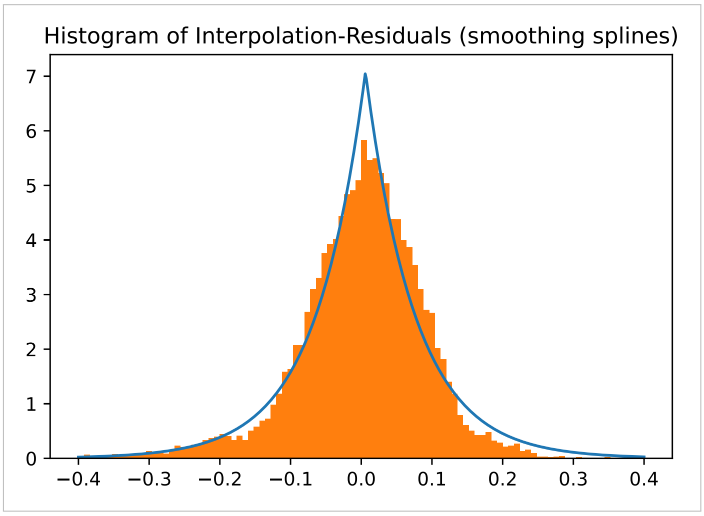

Figure 2.9: XXX caption XXX

Chapter 3

NDVI Correction / Improve NDVI Data

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 (c.f. table ??). In this chapter we would like to improve the observed NDVI values by using more information than just the two bands used to calculate the NDVI (B4 and B8).

3.1 Considering other SCL Classes

In figure 3.1 we see for example that some blue points[1] follow the interpolated line closely and that they might be useful in improving an interpolation fit.

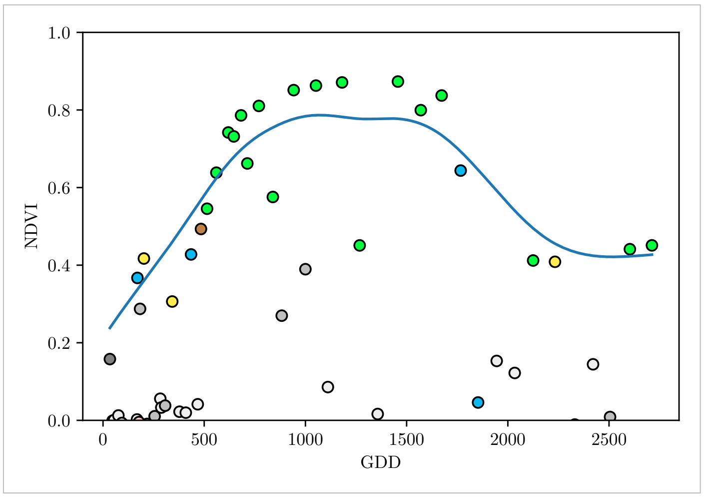

Figure 3.1: A smoothing splines fit considering green and yellow points (scl-classes 4 and 5)

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

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

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

Definition 3.1.0.1. *XXXAssumption (true NDVI) 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 XXX) considering the points $P^{scl45} \setminus P_t$. In the following, we will call this estimate the "true"-NDVI*

shall pair every observed NDVI value with its out-of-bag-estimate. Then for each category we collect all pairs and create a scatter plot in fig 3.2XXXXXXXXXXXXXX

i.) For each pixel and for each observation (every scl-class):
estimate the NDVI value (via out-of-the-box interpolation[2])

ii.)

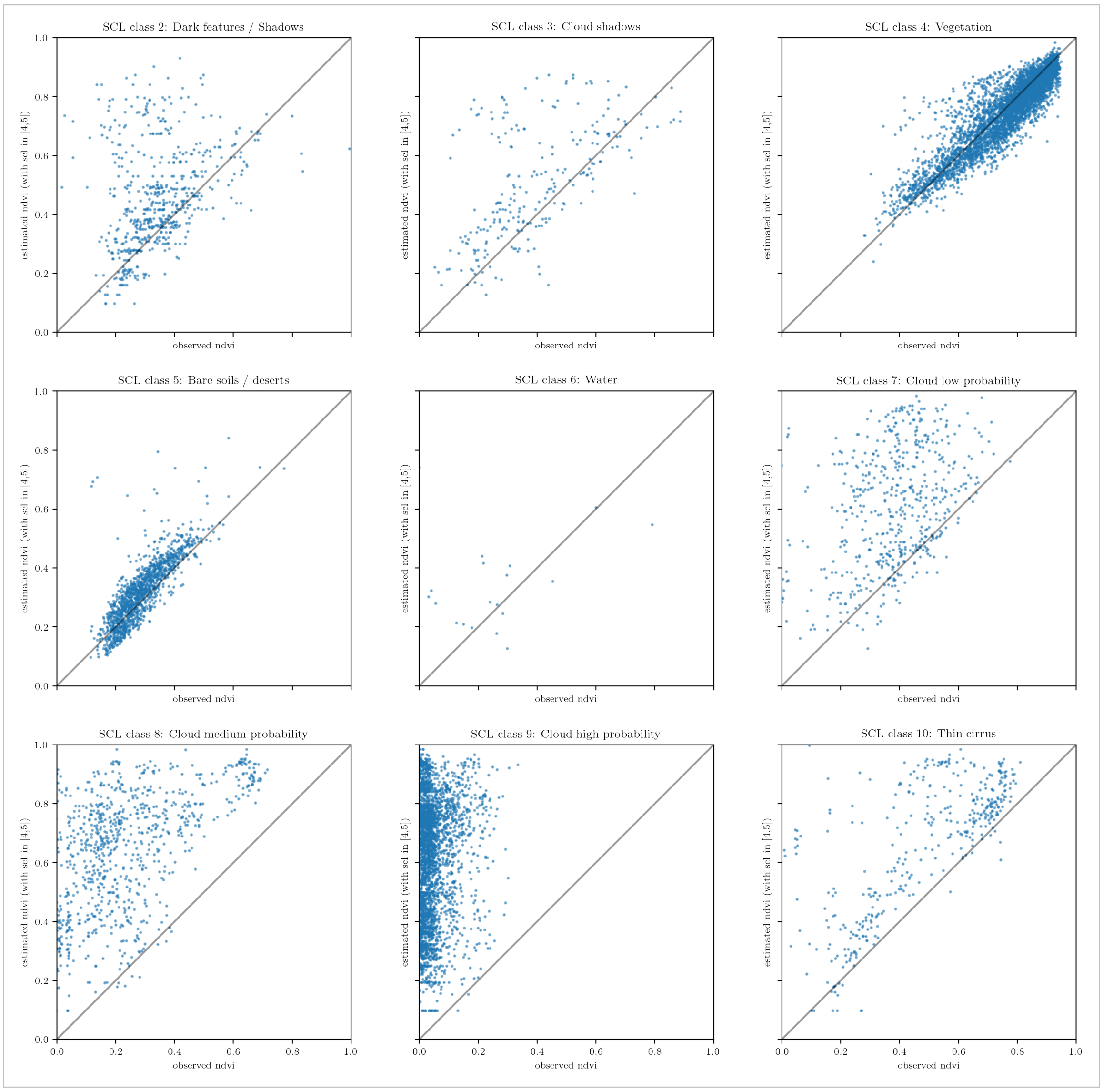

Figure 3.2: XXX caption XXX

[2] That is, we use all observations (with scl-class 4 or 5) but the current one.

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.

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.

Cappellari, M., R. M. McDermid, K. Alatalo, L. Blitz, M. Bois, F. Bournaud, M. Bureau, A. F. Crocker, R. L. Davies, T. A. Davis, P. T. de Zeeuw, P.-A. Duc, E. Emsellem, S. Khochfar, D. Krajnović, H. Kuntschner, R. Morganti, T. Naab, T. Oosterloo, M. Sarzi, N. Scott, P. Serra, A.-M. Weijmans, and L. M. Young. The ATLAS3D project - XX. Mass-size and mass-sigma distributions of early-type galaxies: Bulge fraction drives kinematics, mass-to-light ratio, molecular gas fraction and stellar initial mass function. *432*, 1862–1893.

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*, 332–344.

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

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

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.

Appendix A

Hi Mom