FYS-STK4155 - Project 1

Linear Methods for Regression

Eirill Strand Hauge – Anders Julton
(Dated: October 7, 2019)

The aim of this project was to study three linear regression models: OLS, Ridge regression and LASSO regression, with the k-fold cross-validation resampling technique. The methods were tested and compared by fitting polynomials to the well know Franke's function and calculating their R^2 score and Mean Square Error. The bias-variance tradeoff was also studied for all three methods.

When testing the three methods of regression using the Franke function, Ridge and OLS showed the best results for the less complex systems. When using design matrices of high polynomial degree for fitting, OLS showed to be prone to overfitting, while the two other methods remained stable and thus outperforming OLS by far.

The k-fold cross-validation was seen to be a more reliable way to compute scores of the fitting of data, as the scores were highly influenced by the stochastic element of the data.

The three methods were also tested on real terrain data, calculating how well they managed to reproduce the data. With a grid size of 450×225 the OLS method performed consistently better than both Ridge and LASSO up until the complexity, that is the order of the polynomial fit of the model, yielded numerical errors.

I. INTRODUCTION

Linear regression is used in statistics to model how a dependent variable varies as a function of one or more explanatory variables. This relatively simple linear algorithm plays a vital role in machine learning as it is a fundamental algorithm for a machine to learn or fit a function that maps an input to an output, so called supervised machine learning.

In this project we will use three variations of the linear regression algorithm to fit polynomials to the two-dimensional Franke's function [1], given by

$$\begin{aligned} f(x,y) = &\frac{3}{4}\exp\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right) \\ &+ \frac{3}{4}\exp\left(-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}\right) \\ &+ \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right) \\ &- \frac{1}{5}\exp\left(-(9x-4)^2 - (9y-7)^2\right), \end{aligned} \tag{1}$$

with an added stochastic noise, using normal distribution $N(0,1)$. This function is often used when testing fitting algorithms. A plot of the function can be seen in figure (1). Ordinary Least Squares (OLS), Ridge regression and LASSO (least absolute shrinkage and selection operator) regression will be the methods studied by calculating the R^2 score, mean squared error and by implementing the k-fold cross-validation resampling method. We will also study the bias-variance tradeoff for the three methods.

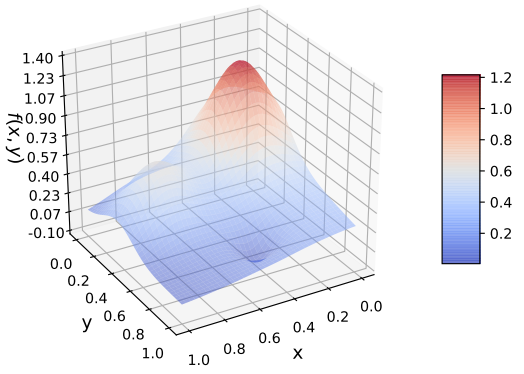

FIG. 1: Plot of Franke's function

Lastly we will use the methods in an attempt to reproduce real topological data (2), gathered from SRTM (Shuttle Radar Topography Mission) footage.

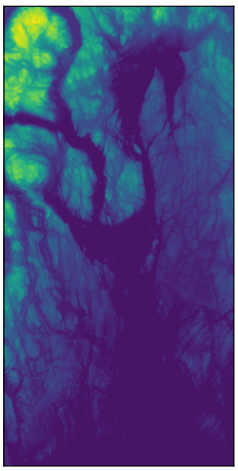

FIG. 2: Terrain data over Oslo, Norway. (Data gathered from `https://earthexplorer.usgs.gov/`

II. METHOD

All three methods described in this section are linear models of regression. Given a function $f(x, y)$, we assume that the function can be approximated linearly. The function is approximated using data z, which may contain noise (ϵ), such that $z = f(x, y)+\epsilon$. The variables (x, y) are transformed into a polynomial design matrix $\mathbf{X}$, which contains all possible combinations of $x^i y^j$ which are of degree smaller or equal to p, given by the set

$$\mathbf{X} = \{x^i y^j : i + j \leq p\}, \tag{2}$$

where i, j, p are positive integers. The aim of all three methods described in this chapter is to find a vector $\boldsymbol{\beta}$, such that

$$\boldsymbol{z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3}$$

where $\boldsymbol{\varepsilon}$ is the error in our approximation.

Each of the three methods takes a design matrix $\mathbf{X}$ of size $n \times m$, where $m = (p + 1)(p + 2)/2$, and a solution vector $\mathbf{z}$ of length n to find a $\boldsymbol{\beta}$ that solves the equation

$$\tilde{\mathbf{z}} = \mathbf{X}\boldsymbol{\beta}, \tag{4}$$

where $\tilde{\mathbf{z}} = \boldsymbol{z} - \boldsymbol{\epsilon}$. Each line in $\mathbf{X}$ will then fulfill equation (3),

$$\tilde{z}_i = \mathbf{X}_i\boldsymbol{\beta}. \tag{5}$$

All three methods aim to minimize expressions containing the residual sum of squares

$$\text{RRS}(\boldsymbol{\beta}) = \sum_{i=1}^{n}(z_i - \tilde{z}_i)^2 = (\mathbf{z} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}), \tag{6}$$

minimizing with respect to $\boldsymbol{\beta}$.

A. Ordinary Least Squares Method

The Ordinary Least Squares method (OLS) will directly minimize the residual sum of squares with respect to $\boldsymbol{\beta}$, giving the expression for OLS:

$$\min_{\boldsymbol{\beta}} ||\mathbf{z} - \mathbf{X}\boldsymbol{\beta}||_2^2, \tag{7}$$

where

$$||\boldsymbol{x}||_2 = \sqrt{\sum_i x_i^2}.$$

By differentiating the expression with respect to $\boldsymbol{\beta}$ we get

$$\frac{\partial \text{RRS}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 2\mathbf{X}^T(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}). \tag{8}$$

Setting the derivative to zero we obtain

$$\begin{aligned}\mathbf{X}^T(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}) &= 0\\ \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} &= \mathbf{X}^T\mathbf{z},\end{aligned}$$

yielding an expression for $\boldsymbol{\beta}$

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z}. \tag{9}$$

Since the matrix $\mathbf{X}^T\mathbf{X}$ might be singular, $\mathbf{X}$ can be factorized using Singular Value decomposition in order to find a pseudo-inverse

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T. \tag{10}$$

where $\mathbf{U}$ is an $n \times m$ orthonormal matrix, $\mathbf{D}$ is a $m \times m$ diagonal matrix with non-negative real (singular) values on the diagonal, and $\mathbf{V}$ is an $m \times m$ unitary matrix. Using SVD, equation (9) can be rewritten as

$$\begin{aligned}\boldsymbol{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z}\\ &= ((\mathbf{U}\mathbf{D}\mathbf{V}^T)^T\mathbf{U}\mathbf{D}\mathbf{V}^T)^{-1}(\mathbf{U}\mathbf{D}\mathbf{V}^T)^T\mathbf{z}\\ &= (\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{z}\\ &= (\mathbf{V}\mathbf{D}^2\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{z}\\ &= \mathbf{V}\mathbf{D}^{-2}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{z}\\ &= \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^T\mathbf{z}.\end{aligned}$$

The two next regression methods, Ridge and LASSO regression, build on the principles of OLS. In the same way as OLS, Rigde and LASSO aim to minimize the residual sum of squares. These methods are called shrinkage methods, as they tend to produce regression coefficients with lower variability compared to OLS.

The shrinkage methods minimize a penalized expression for the residual sum of squares.

We will start by explaining the Ridge regression method.

B. Ridge Regression

The regression method minimizes the residual sum of squares with respect to $\boldsymbol{\beta}$, penalized by the norm of $\boldsymbol{\beta}$ times a constant value $\lambda_R > 0$. The expression for Ridge regression becomes:

$$\min_{\boldsymbol{\beta}} \left(||\mathbf{z} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda_R ||\boldsymbol{\beta}||_2^2 \right). \tag{11}$$

As the second term is proportional to the norm of $\boldsymbol{\beta}$, minimizing this expression will force regression coefficients to shrink compared to OLS.

By differentiating the expression with respect to $\boldsymbol{\beta}$ we get

$$\frac{\partial \text{RRS}(\boldsymbol{\beta}) + \lambda_R ||\boldsymbol{\beta}||_2^2}{\partial \boldsymbol{\beta}} = 2\mathbf{X}^T(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda_R \boldsymbol{\beta}. \tag{12}$$

Setting the derivative to zero we obtain

$$\begin{aligned} \mathbf{X}^T(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}) + \lambda_R \boldsymbol{\beta} &= 0 \\ \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + \lambda_R \mathbf{I}\boldsymbol{\beta} &= \mathbf{X}^T\mathbf{z}, \end{aligned}$$

yielding an expression for $\boldsymbol{\beta}$

$$\boldsymbol{\beta}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda_R \mathbf{I})^{-1}\mathbf{X}^T\mathbf{z}. \tag{13}$$

Now, adding a positive constant to the diagonal of the symmetric matrix $\mathbf{X}^T\mathbf{X}$ will ensure that the matrix is non-singular. The matrix $(\mathbf{X}^T\mathbf{X} + \lambda_R\mathbf{I})$ will therefore always be invertible.

The next method, the LASSO regression is another shrinkage method.

C. LASSO Regression

LASSO stands for "least absolute shrinkage and selection operator", and minimizes the following expression:

$$\min_{\boldsymbol{\beta}} \left(||\mathbf{z} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda_L ||\boldsymbol{\beta}||_1 \right), \tag{14}$$

where

$$||\boldsymbol{x}||_1 = \sum_i |x_i|. \tag{15}$$

Equation (15) makes the solutions nonlinear in the z_i [2], so there is no closed form expression for $\boldsymbol{\beta}^{\text{lasso}}$.

D. Bias-Variance tradeoff

We will start by looking at the mean squared error

$$\text{MSE}(\boldsymbol{\beta}) = \frac{\text{RRS}(\boldsymbol{\beta})}{n} = \frac{1}{n}\sum_{i=1}^{n}(z_i - \tilde{z}_i)^2. \tag{16}$$

and rename it in terms of an expectation value

$$\text{MSE}(\boldsymbol{\beta}) = \mathbb{E}\left[(\mathbf{z} - \tilde{\mathbf{z}})^2\right]. \tag{17}$$

Taking a closer look at the expectation value

$$\begin{aligned} \mathbb{E}\left[(\mathbf{z} - \tilde{\mathbf{z}})^2\right] &= \mathbb{E}\left[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}] + \mathbb{E}[\tilde{\mathbf{z}}] - \tilde{\mathbf{z}})^2\right] \\ &= \mathbb{E}\Big[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}])^2 + 2\left(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}]\right)\left(\mathbb{E}[\tilde{\mathbf{z}}] - \tilde{\mathbf{z}}\right) \\ &\quad + \left(\mathbb{E}[\tilde{\mathbf{z}}] - \tilde{\mathbf{z}}\right)^2 \Big], \end{aligned}$$

such that we get an expression consisting of three terms

$$\begin{aligned} \mathbb{E}\left[(\mathbf{z} - \tilde{\mathbf{z}})^2\right] &= \mathbb{E}\left[(\tilde{\mathbf{z}} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] + \mathbb{E}\left[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] \\ &\quad + 2\mathbb{E}\Big[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}])\left(\mathbb{E}[\tilde{\mathbf{z}}] - \tilde{\mathbf{z}}\right) \Big]. \end{aligned} \tag{18}$$

We recognize that the first term is equivalent to

$$\mathbb{E}\left[(\tilde{\mathbf{z}} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] = \text{Var}(\tilde{\mathbf{z}}), \tag{19}$$

the variance of $\tilde{\mathbf{z}}$.

Considering the next term, using that $\mathbf{z} = \mathbf{f} + \boldsymbol{\epsilon}$, we get

$$\begin{aligned} \mathbb{E}\left[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] &= \mathbb{E}\left[\mathbf{z}^2 - 2\mathbb{E}[\tilde{\mathbf{z}}]\mathbf{z} + \mathbb{E}[\tilde{\mathbf{z}}]^2\right] \\ &= \mathbb{E}\left[(\mathbf{f} + \boldsymbol{\epsilon})^2 - 2\mathbb{E}[\tilde{\mathbf{z}}](\mathbf{f} + \boldsymbol{\epsilon}) + \mathbb{E}[\tilde{\mathbf{z}}]^2\right] \\ &= \mathbb{E}\left[(\mathbf{f} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] + 2(\mathbf{f} - \mathbb{E}[\tilde{\mathbf{z}}])\mathbb{E}\left[\boldsymbol{\epsilon}\right] + \mathbb{E}\left[\boldsymbol{\epsilon}^2\right]. \end{aligned}$$

Where we use that $\mathbf{f}$ and $\mathbb{E}[\tilde{\mathbf{z}}]$ are not stochastic. Since $\boldsymbol{\epsilon}$ is assumed to have a Gaussian distribution around zero, we know that

$$\mathbb{E}\left[\boldsymbol{\epsilon}\right] = 0, \tag{20}$$

and that

$$\mathbb{E}\left[\boldsymbol{\epsilon}^2\right] = \sigma^2, \tag{21}$$

where σ^2 is the standard deviation of $\boldsymbol{\epsilon}$. The second term can therefore be written as

$$\mathbb{E}\left[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] = \mathbb{E}\left[(\mathbf{f} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] + \sigma^2. \tag{22}$$

We further recognize that

$$\mathbb{E}\left[(\mathbf{f} - \mathbb{E}[\tilde{\mathbf{z}}])^2\right] = \text{Bias}^2(\tilde{\mathbf{z}}). \tag{23}$$

Moving on to the third term

$$\begin{aligned} &\mathbb{E}\Big[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}])\left(\mathbb{E}[\tilde{\mathbf{z}}] - \tilde{\mathbf{z}}\right) \Big] \\ &= \mathbb{E}[\tilde{\mathbf{z}}]\mathbb{E}[\tilde{\mathbf{z}}] - \mathbb{E}[\tilde{\mathbf{z}}\mathbf{z}] - \mathbb{E}[\tilde{\mathbf{z}}]\mathbb{E}[\tilde{\mathbf{z}}] + \mathbb{E}[\mathbf{z}]\mathbb{E}[\tilde{\mathbf{z}}] \\ &= -\mathbf{f}\mathbb{E}[\tilde{\mathbf{z}}] - \mathbb{E}[\tilde{\mathbf{z}}\boldsymbol{\epsilon}] + \mathbf{f}\mathbb{E}[\tilde{\mathbf{z}}] + \mathbb{E}[\epsilon]\mathbb{E}[\tilde{\mathbf{z}}] \\ &= -\mathbb{E}[\tilde{\mathbf{z}}\boldsymbol{\epsilon}], \end{aligned}$$

where $\tilde{\mathbf{z}}$ and $\boldsymbol{\epsilon}$ are independent variables, and can therefore be separated yielding $\mathbb{E}[\tilde{\mathbf{z}}\boldsymbol{\epsilon}] = \mathbb{E}[\tilde{\mathbf{z}}]\mathbb{E}[\boldsymbol{\epsilon}]$. We then have that the third term is zero,

$$2\mathbb{E}\Big[(\mathbf{z} - \mathbb{E}[\tilde{\mathbf{z}}])\left(\mathbb{E}[\tilde{\mathbf{z}}] - \tilde{\mathbf{z}}\right) \Big] = 0. \tag{24}$$

We can then rewrite the mean squared error as

$$\begin{aligned}\mathrm{MSE}(\boldsymbol{\beta}) &= \mathbb{E}\left[((\mathbf{z}-\tilde{\mathbf{z}})^2\right] \\ &= \mathbb{E}\left[(\tilde{\mathbf{z}}-\mathbb{E}[\tilde{\mathbf{z}}])^2\right] + \mathbb{E}\left[(\mathbf{f}-\mathbb{E}[\tilde{\mathbf{z}}])^2\right] + \sigma^2. \quad (25)\\ &= \mathrm{Var}(\tilde{\mathbf{z}}) + \mathrm{Bias}^2(\tilde{\mathbf{z}}) + \sigma^2.\end{aligned}$$

The variance is a measure of how far the values of $\tilde{\mathbf{z}}$ is spread from the average of the values. A high variance can cause overfitting, that is the algorithm fits to the training data instead of the function that created the data, and will generally reduce the prediction strength of the algorithm. This can happen if the order of polynomial of the algorithm is higher than the function being fitted, or if the number of data points are too low.

The bias is a measure of the error from the assumptions made when fitting the output. A high bias can cause underfitting, that is the algorithm misses relations between the data, and can happen when using a lower order polynomial to fit the data than the order of the polynomial for the function being fitted, or if the number of data points are too high.

The bias-variance tradeoff is therefore trading lower bias for higher variance, and vice versa.

Overfitting can be detected by splitting your data into training and test data. If, for instance, after tuning your algorithm on the training data the accuracy is much greater on the training data than on the test data, most likely the algorithm is fitted to the training data instead of the actual data and overfitting has occurred.

E. Resampling and cross-validation

To better test the validity of our fitting algorithms, we will implement the resampling method k-fold cross-validation. The method can be summarized as followed:

1. Randomly shuffle the training data.
2. Split the data into k groups/folds, preferably of equal size.
3. Retain one fold as the test data, the $k-1$ remaining folds as training data.
4. Fit the model on the training data and evaluate it on the test data, retaining the score.
5. Repeat until each fold has served as the test data.
6. Take the average the score, which will be your final estimate.

A diagram of the procedure can be seen in figure (3).

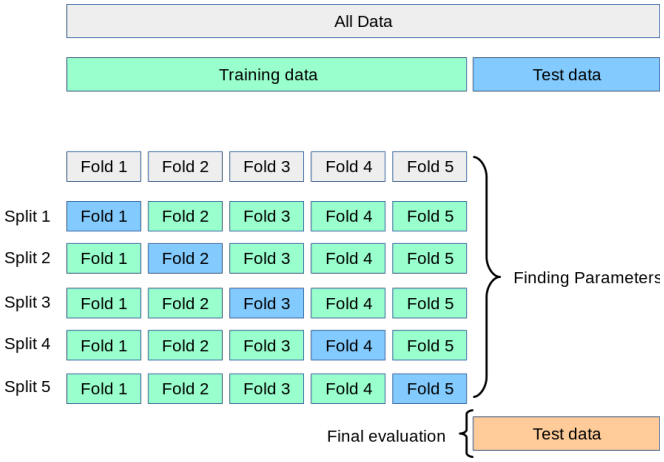

FIG. 3: A schematic representation of k-fold cross-validation (Copied from [5]).

F. Statistics

In this project we will study the regression methods by calculating their MSE scores, defined earlier, and R^2 score defined by

$$R^2(z,\tilde{z}) = 1 - \frac{\sum_{i=0}^{n-1}(z_i-\tilde{z}_i)^2}{\sum_{i=0}^{n-1}(z_i-\bar{\tilde{z}})^2}, \quad (26)$$

where $\bar{\tilde{z}}$ is the mean value of $\tilde{z}$.

We will also calculate the 95 % confidence interval of the β_i values from the variance-covariance matrix of $\boldsymbol{\beta}$, For the OLS method, the variance-covariance matrix is given by

$$\mathrm{Var}(\boldsymbol{\beta}^{\mathrm{OLS}}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2, \quad (27)$$

while for Ridge regression the variance-covariance matrix is given by [6]

$$\mathrm{Var}(\boldsymbol{\beta}^{\mathrm{Ridge}}) = (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\sigma^2, \quad (28)$$

where the variance σ^2 is given by

$$\sigma^2 = \frac{1}{n-m-1}\sum_{i=1}^{n}(z_i-\tilde{z}_i)^2. \quad (29)$$

The 95 % confidence interval is then given by [2]

$$\beta_j \pm 1.96\cdot\sqrt{[\mathrm{Var}(\boldsymbol{\beta})]_{ii}}\sigma. \quad (30)$$

There is no consensus on a statistically valid method for calculating standard errors of the LASSO parameters, so no attempt will be made to calculate the confidence interval for LASSO regression. [4]

III. IMPLEMENTATION

The full implementation can be found at `https://github.com/andersjulton/FYS-STK4155/tree/master/Project_1`.

For the implementation the Python programming language was chosen, as there are several available packages convenient for this project. The numpy library matrices were used, as well as several linear algebra methods. The numpy library was also used to perform singular value decomposition and to provide random numbers in order to create data sets for the Franke's function.

The methods for regression were implemented in a class hierarchy. An abstract superclass was implemented to contain most of the supporting methods, such as the method for computing the design matrix, resampling methods and methods for statistical analysis.

While the classes for OLS regression and ridge regression depend only on the linear algebra methods from numpy library, the class performing LASSO regression uses the implementation of scikit-learn. [5]

All figures were produced using the Matplotlib library. [3]

IV. RESULTS

A. Study Using the Franke's Function

A study of the OLS method as a function of the size of train data is shown in figures (4) and (5). The polynomial degree of the design matrix was kept at 5, and all values shown on the plots were calculated as the average of ten runs. The scores were calculated using separate test data.

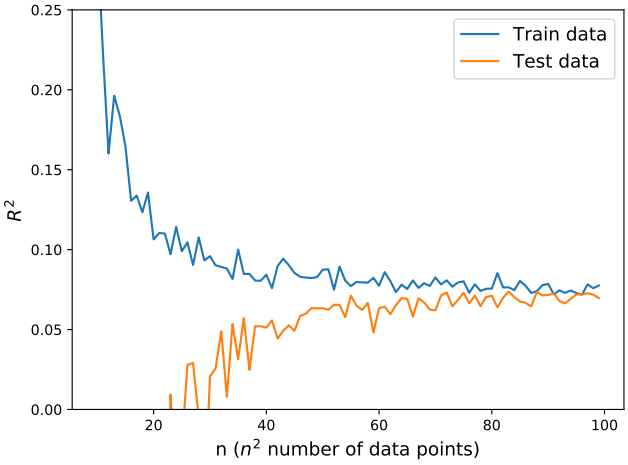

FIG. 4: Showing R^2 of test data as a function of grid size $(n \times n)$ of train data.

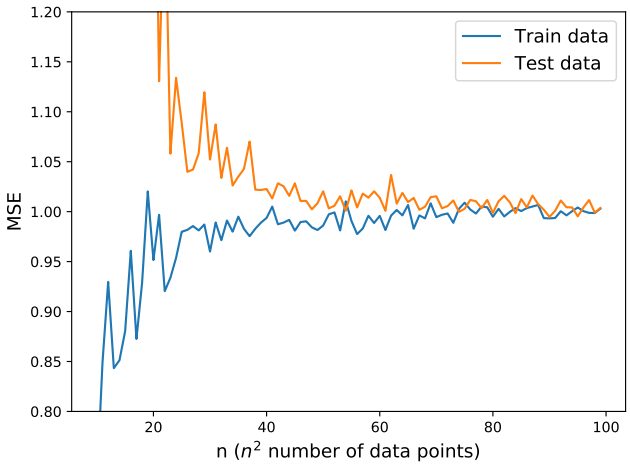

FIG. 5: Showing MSE of test data as a function of grid size $(n \times n)$ of train data.

A study of the OLS method as a function of the polynomial degree of the design matrix is shown in figure (6). The figure compares MSE score of the test data and train data. The number of data points used was 81×81.

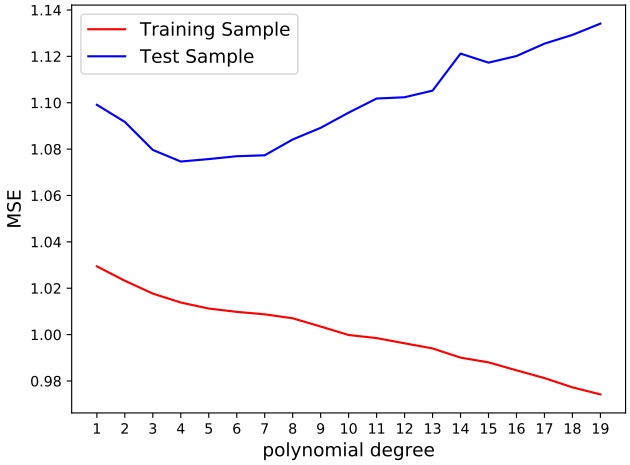

FIG. 6: Plot of test and training error as function of polynomial degree, using OLS.

To show the effect of implementing k-fold cross-validation, a run was done with 50×50 data points, $p = 5$ and $k = 10$ using OLS. MSE scores calculated for each iteration are shown in figure (7). The line in the figure shows the mean MSE value of all the folds in the k-fold cross validation.

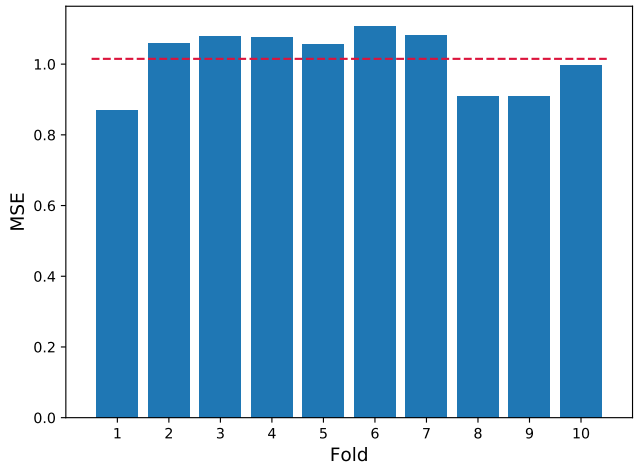

FIG. 7: MSE scores with OLS for each iteration in k-fold cross-validation.

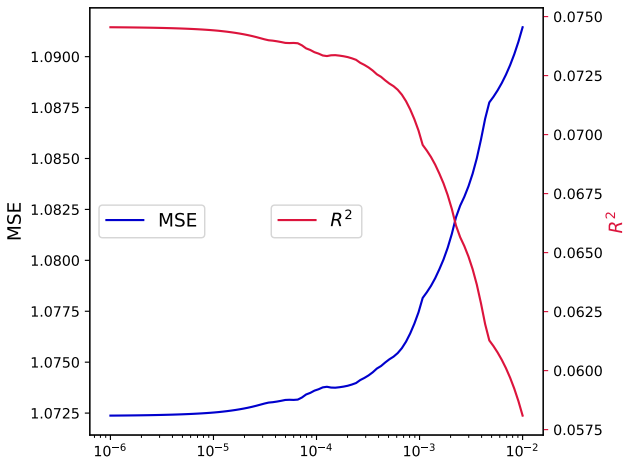

FIG. 9: MSE and R^2 for LASSO with $p = 5$ for various values of λ.

To study the effect of varying the hyperparameter λ a run was done with both Ridge and LASSO regression with 81×81 data points and $p = 5$, calculating MSE and R^2 scores. The results can be seen in figure (8) and (9) for Ridge and LASSO respectively. The runs were tested against test data with added noise.

In order to find suitable λ-values for the Ridge and the LASSO regression methods, the MSE scores was studied as a function of λ. The MSE scores, using separate test data without added noise, were calculated for $\lambda \in [10^{-7}, 10^{-2}]$. The λ providing the lowest MSE-value is shown on the figures below as function of the polynomial degree of the design matrix. Figure (10) shows the λ-values best suited for the Ridge regression, while figure (11) show the λ-values best suited for the LASSO regression. The number of data points used to train was held at 100×100.

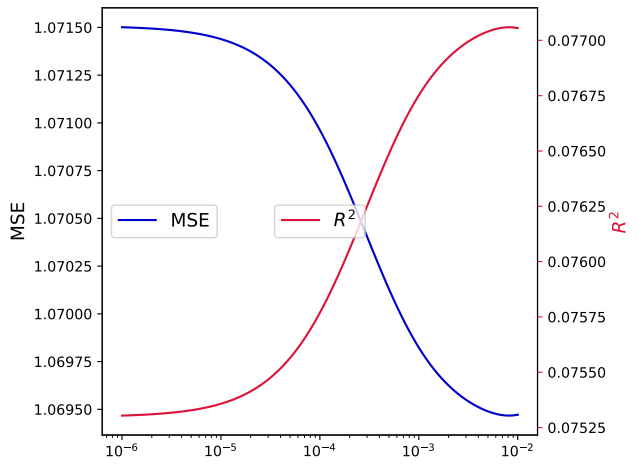

FIG. 8: MSE and R^2 for Ridge with $p = 5$ for various values of λ.

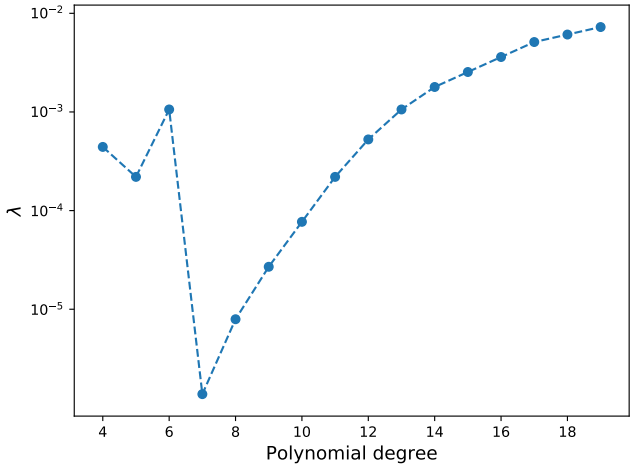

FIG. 10: The λ-values corresponding to a minima on MSE when the Ridge regression method was used. Shown as a function of the polynomial degree of the design matrix.

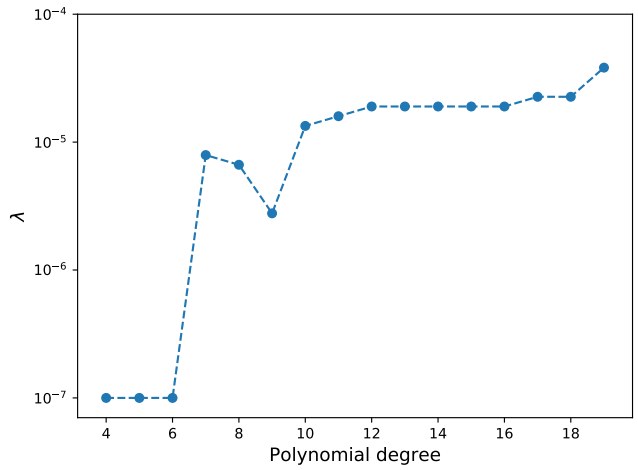

FIG. 11: The λ-values corresponding to a minima on MSE when the LASSO regression method was used. Shown as a function of the polynomial degree of the design matrix.

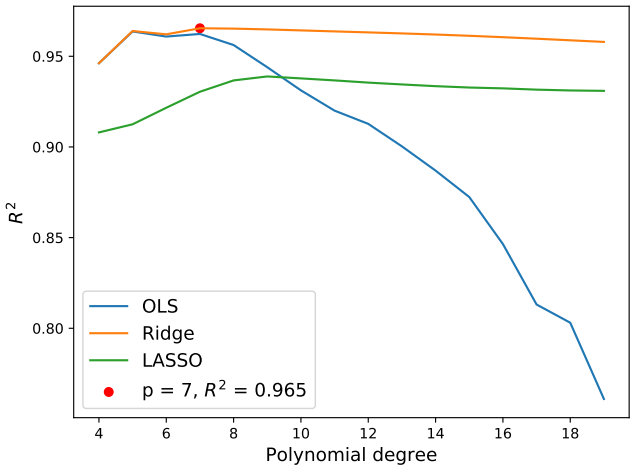

FIG. 13: R^2 scores for OLS, Ridge and LASSO. For Ridge and LASSO the best found λ for each polynomial degree was used.

Using these λ-values, the MSE and R^2 scores were calculated on test data without noise. The MSE score is shown in figure (12) and the R^2 score is shown in figure (13). The number of data points used to train was held at 100×100.

The variance and bias squared was illustrated using all three methods of regression. The grid size of the train data was held at 81×81 and the polynomial degree of the design matrix was set to five. The result of this study using OLS is shown in figure (14). The results using Ridge regression in figure (15) and LASSO regression in figure (16) were computed for a few selected λ values.

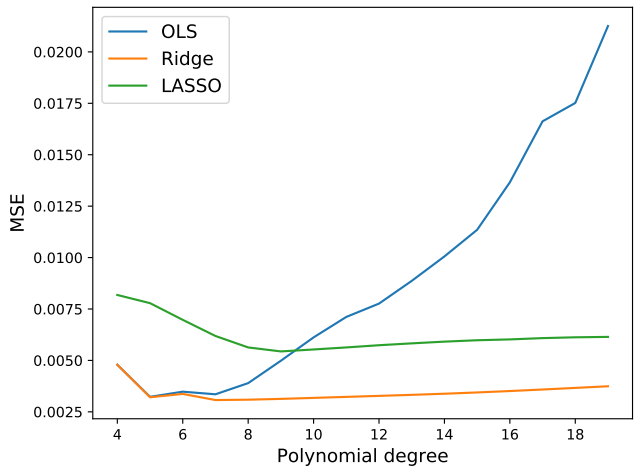

FIG. 12: MSE scores for OLS, Ridge and LASSO. For Ridge and LASSO the best found λ for each polynomial degree was used.

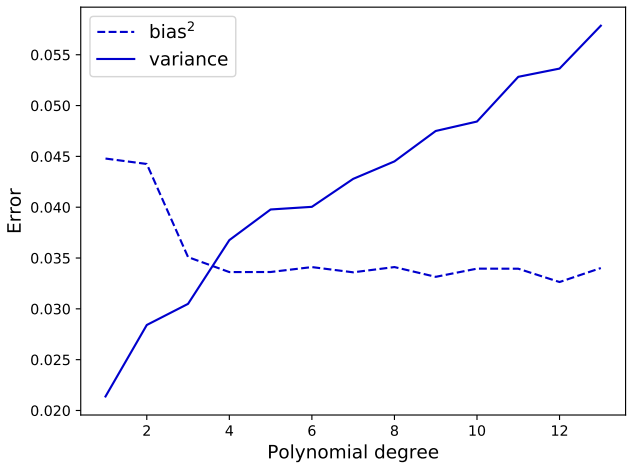

FIG. 14: Plot of Bias^2 and variance for OLS as a function of polynomial degree.

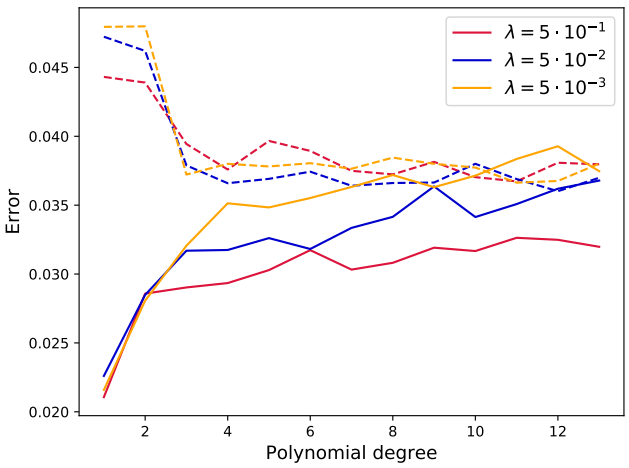

FIG. 15: Plot of Bias^2 (dashed) and variance (solid) for Ridge as a function of polynomial degree.

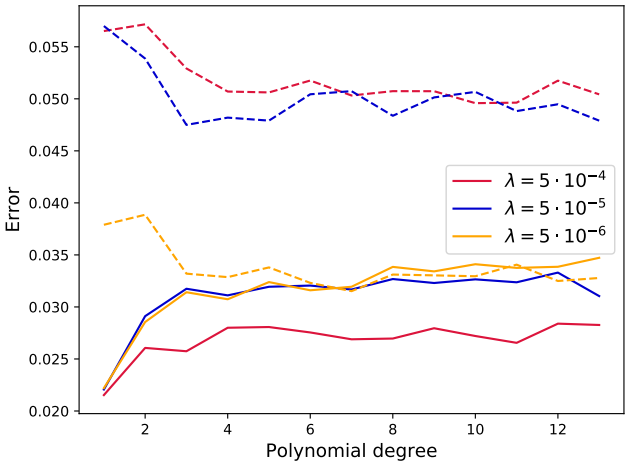

FIG. 16: Plot of Bias^2 (dashed) and variance (solid) for LASSO as a function of polynomial degree.

The figures below show the 95% confidence interval of the β-values. The calculations were made using a design matrix with 80×80 data points and $p = 5$. The confidence interval was calculated both with and without noise in the Franke's function. Figure (17) shows the confidence interval of β when the OLS method was used for the fitting of the data. Figure (18) shows the confidence interval of β of Ridge regression using $\lambda = 7.4 \times 10^{-4}$. Figure (19) shows the standard deviations for the β parameters for OLS and Ridge with $\lambda = 7.4 \times 10^{-4}$.

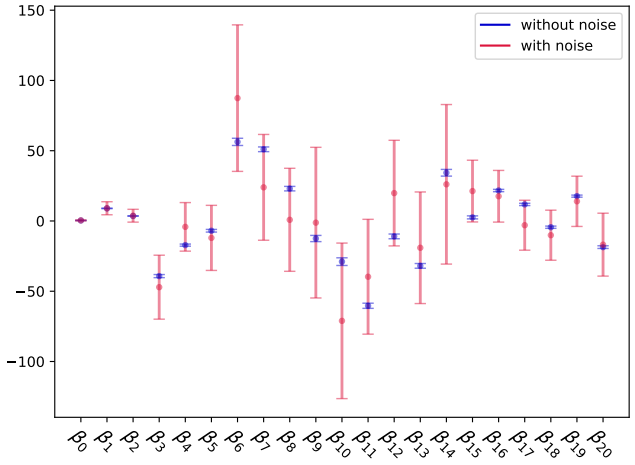

FIG. 17: Confidence interval of β when the OLS method was used.

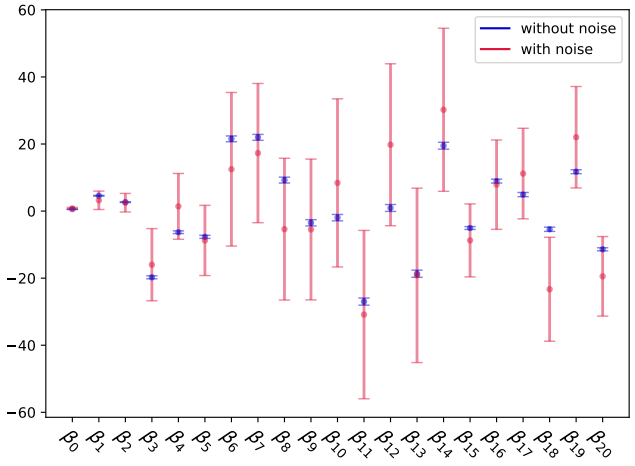

FIG. 18: Confidence interval of β when the Ridge regression method was used.

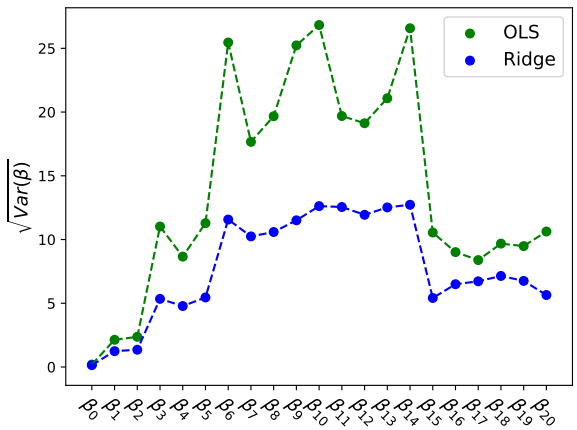

FIG. 19: Plot of $\sqrt{\mathrm{Var}(\beta)}$ for OLS and Ridge.

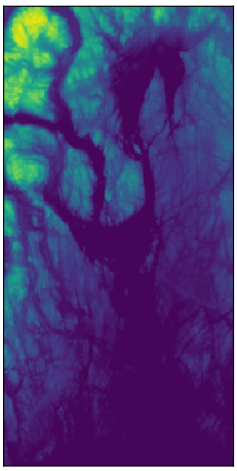

FIG. 21: Downsized terrain data with 450×225 data points.

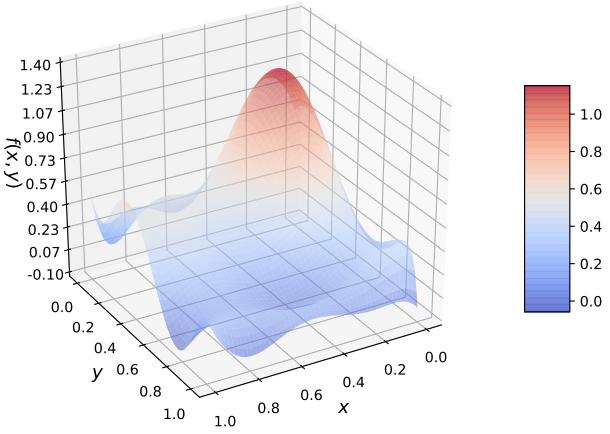

FIG. 20: Plot of model fit of Franke's function using Ridge regression.

B. Study Using Terrain Data

The original terrain data as seen in figure (2) contains 3601×1801 data points. For our study the data was downsized by a factor 8 by removing one column and one row, splitting the data in 8×8 grids and using the average values of the grids to form a new 450×225 image. Figure (21) shows the downsized image.

R^2 scores were calculated as a function of polynomial degree for all three methods, where the results are shown in figure (22). The λ-values used in the calculations was found by similar method as when studying the Franke's function, although most of the values was trending towards zero.

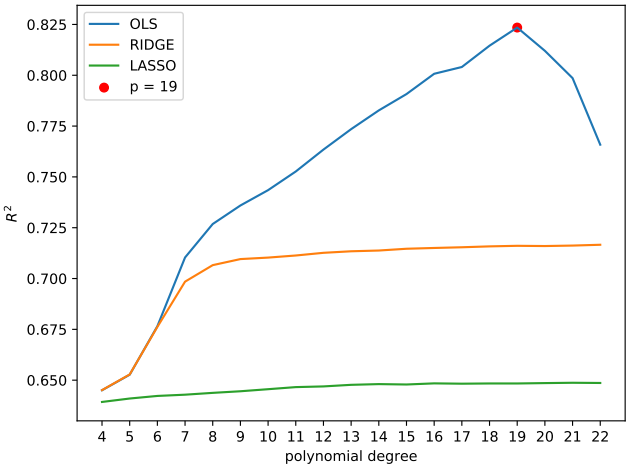

FIG. 22: R^2 scores from fitting terrain data with OLS, Ridge and LASSO. The polynomial degree that yielded best R^2 score for OLS is shown.

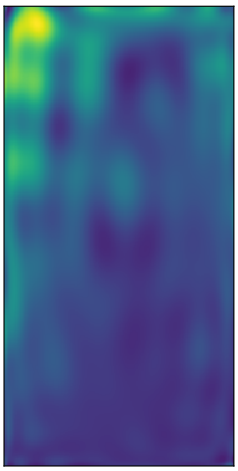

FIG. 23: Reproduction of terrain data using OLS with $p = 19$.

V. DISCUSSION

A. Study Using the Franke's Function

The study of OLS as a function of the size of training data, illustrated in figures (5) and (4), showed overfitting when the data set was smaller than 40×40. This can be seen as the train data show far better MSE scores than the test data. The scores for MSE and R^2 stabilized as the grid size surpassed 40×40. The MSE score stabilized around MSE $= 1$. This is equivalent to the noise of the data set ($\sigma^2 = 1$), showing that the regression is reproducing the Franke's function within the limitations of the noise. The R^2 score shown in figure (4) show the same trend as the MSE score, stabilizing around $R^2 = 0.07$. As an ideal R^2 score should be $R^2 = 1$, this seems like very poor performance. However, comparing this value with results on figure (13), where no noise was used in the test data, it apparent that this score was highly influenced by the noise added to function. The R^2 score at polynomial degree 5 was $R^2 > 0.95$ when no noise was added to the test data.

Signs of overfitting was apparent when the polynomial degree of the design matrix was increased. This can be observed in figure (6). The MSE score for the train data declined (improved) as the polynomial degree increased. The MSE score of the test data was consistently higher than the MSE of the train data. The MSE score of the test data followed the same rate of decline in MSE as the train data while the polynomial degree was low ($p < 5$). The trend of the MSE score of the test data shifted around $5 \leq p < 7$. The MSE steadily increased for higher polynomial degrees. The increasing MSE value of the test sample was due to overfitting.

It was observed throughout the project that the stochastic element of the computations had a significant impact. The results of the simulations depended heavily on the randomly generated data sets, as well as the added noise to the Franke's function. The study of the k-fold cross validation shown in figure (7), show how the MSE score varied depending on the fold. The mean value of the MSE score was close to reproducing the noise added to the function.

For $p = 5$, both Ridge and LASSO showed better prediction for smaller λ values, as illustrated in figures (8) and (9). The best choice for λ-values based on MSE score was observed to increase as the polynomial degree of the design matrix increased. Figure (10) shows this trend to be clear for the Ridge regression method, and figure (11) show the same trend, although weaker, for the LASSO regression method. This trend is not consistent for either methods, as the scores are highly dependent on both the number of data points and the random noise added to the function. The λ-values can be observed to be generally lower for the LASSO regression method compared to the Ridge regression method.

The Ridge and LASSO regression methods were shown to be less prone to overfitting than the OLS method. This is illustrated in figures (12) and (13). The performance of the OLS method sharply declined when the polynomial degree of the design matrix exceeded $p = 7$, while the two other methods showed a more stable behavior. While OLS showed signs of overfitting, the steep decline in performance when $p > 20$ may not be overfitting, but could be caused by numerical errors.

A general trend of increasing variance and reduced bias^2 as a function of polynomial degree was observed in the studies shown in figures (14), (15) and (16). The bias - variance tradeoff was most visible for the OLS method. The performance of Ridge using different λ values showed consistently lower bias when using larger λ values, while a similar pattern in the bias^2 was slightly less discernible. For the LASSO regression method, the lowest λ value $\lambda = 5 \cdot 10^{-6}$ outperformed the two other λ values. While the variance was on the same level as $\lambda = 5 \cdot 10^{-5}$ and higher than for $\lambda = 5 \cdot 10^{-4}$, the bias^2 was significantly lower for $\lambda = 5 \cdot 10^{-6}$ independent of the polynomial degree.

It was shown in figures (17) and (18) that the confidence intervals of the β values increased greatly when noise was added to the function. In figure (19) we see the variance of the β values in Ridge regression is consistently smaller than for OLS, as expected.

B. Study Using Terrain Data

Using the regression methods on the real terrain data showed that OLS performed better than both Ridge and LASSO regression for all but the lowest degree of polynomials. Here the performance of OLS and Ridge was essentially equal, while LASSO performed worse than the two others. At around $p = 6$ OLS outperformed also Ridge, with a peak performance at $p = 19$. The low performance of Ridge could be a consequence of the method used to determining the best λ-value. In theory

Ridge should perform as well as OLS since the hyperparamater in Ridge would go to zero if OLS was the better method. In our calculations the smallest value for λ that was tested was 10^{-7}. By examining even lower values, we expect Ridge to perform on par with OLS. Another explanation could be the different representations of $\boldsymbol{\beta}$ for OLS and Ridge. In OLS we used SVD (Singular Value Decomposition) to avoid problems with singular matrices, while in Ridge we used the design matrix directly, so numerical errors in the two methods could cause deviations. Using SVD on Ridge as well might be worth examining in future projects regarding these regression methods.

The sharp drop in the performance of OLS at $p = 19$ is most likely due to numerical errors due to the size of the design matrix and not overfitting. In future projects it would be interesting to study this peak as a function of data points.

The low performance of the LASSO method is somewhat harder to understand as the external package from SKlearn was used. A separate run with smaller λ-values (down to 10^{-9}) and with max iterations increased from 1000 to 5000 was done, but did not yield any significant increase in performance for LASSO.

Figure (23) shows the terrain data reproduced using OLS with the polynomial degree that yielded the best R^2 score, $p = 19$. While not a perfect reproduction, one can clearly see the some of the major features from the original image depicted in the reproduced image.

VI. CONCLUSION

The β values, and thus results in general, for the same function/problem depend greatly on the train data provided. The evaluation of performance using k-fold cross validation therefore provides a more reliable result.

OLS and Ridge performed quite similarly well when lower polynomial degrees was used ($p < 8$) for fitting the Franke function. The LASSO method was yielding somewhat poorer results for the lower polynomial degrees.

OLS method was more vulnerable to overfitting due to high polynomial degree of the design matrix. The penalized regression methods, Ridge and LASSO, were shown to be more stable, outperforming OLS by far as the polynomial degree increased.

For the terrain data, OLS was consistently better than both Ridge and LASSO regression, with Ridge being consistently better than LASSO. Even when tuning the LASSO method with expanded search for better λ-values and increasing iterations, the method did not yield noticeable better results.

[1] Richard Franke. A critical comparison of some methods for interpolation of scattered data, 1979.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[3] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[4] Minjung Kyung, Jeff Gill, Malay Ghosh, and George Casella. Penalized regression, standard errors, and bayesian lassos. *Bayesian Anal.*, 5(2):369–411, 06 2010.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] Vedide Uslu, Erol Egrioglu, and Eren Bas. Finding optimal value for the shrinkage parameter in ridge regression via particle swarm optimization. *American Journal of Intelligent Systems 2014*, 2014:142–147, 04 2014.