Regression and Resampling methods

FYS-STK4155: Project 1

Trygve Leithe Svalheim
github.com/trygvels/FYS-STK4155

October 7, 2019

Abstract

We look at three different types of regression methods and their ability to fit a polynomial of varying degree to two different types of data. We look at Ordinary Least Squares (OLS), Ridge and Lasso regression and discuss their advantages and hyper parameters. Additionally, we discuss the advantages of resampling methods and the bias-variance trade-off problem. We conclude that OLS is the better method for fitting to the mathematically produced and predictable Franke function data, but that we obtain the lowest error for the terrain data using a regularizing term with Ridge regression.

Contents

1. INTRODUCTION

In this project we explore the problem of regression analysis, resampling and optimal hyperparemeters. Regression is the process modelling a response $\tilde{y}$ to true observed data y from a predictor variable x to learn about the relationship between data. In this work, our focus will be on Linear regression, where the relationship between x and y is strictly linear and can be described by the model

$$\tilde{\mathbf{y}} = \mathbf{X}\beta, \tag{1}$$

and the true response can be described with the addition of an error term ϵ, denoting the modelling error so that

$$\mathbf{y} = \mathbf{X}\beta + \epsilon. \tag{2}$$

Linear regression is a fundamental method of statistical analysis, and allows us to study the relationships hidden in all types of data sets, which is increasingly powerful in a world where data is both abundant and complex. In this study, we look at three common regression methods and assess their capabilities on two different data sets. In addition, we explore the importance resampling methods and the impact of hyper-paremeters and their data dependence.

All relevant code can be found at github.com/trygvels/FYS-STK4155 with all figures and relevant tests to reproduce the data. The code is structured so that it uses a parameter-file for reproducing the presented results. It also allows the user to test the methods against the `Scikit Learn`-alternative.

2. REGRESSION METHODS

In this section we outline the basics of three different regression methods which we later apply to our two data sets. If not otherwise stated, we are following the course lecture notes.[1]

2.1. Ordinary Least Squares (OLS)

As briefly mentioned in the introduction, a Linear regression system revolves around modeling a function $\mathbf{y}(\mathbf{X})$, where the matrix $\mathbf{X}$ containing the predictors is called *design matrix*. Again, we want to find the values of β, that minimizes the error ϵ describing the difference between the predicted and true values $\tilde{\mathbf{y}}$ and $\mathbf{y}$. However, there are several different ways of describing the error, for the first method, the *Ordinary Least Squares (OLS)* method, we chose a *cost function* parameterized by the Euclidean L^2 norm,

$$\begin{aligned} C(\boldsymbol{\beta}) &= \|\mathbf{y} - \tilde{\mathbf{y}}\|_2^2 \\ &= \sum_{i=1}^{n} \left| y_i - \sum_{j=0}^{p} X_{ip}\beta_p \right|^2. \end{aligned} \tag{3}$$

Furthermore we can quantize the full error using the *Mean Squared Error* (MSE), which is simply the ensamble average over the L^2-loss. For this particular error metric, the optimal β parameters can be found by minimising this function. The linear algebra representation of this minimization problem can be described following Hastie, Tibshirani & Friedman[2] by

$$\begin{aligned} \frac{\partial C(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \frac{\partial}{\partial \boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0 \\ \boldsymbol{\beta}_{\text{optimal}} &= \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{y}. \end{aligned} \tag{4}$$

For some datasets, the inversion operation $\left(\mathbf{X}^T\mathbf{X}\right)$ is too costly. With the data sizes we are dealing with this should not lead to any problems. However, in order to generalize our program, we chose to employ the Singular Value Decomposition (SVD) method. Without going into the nits and grits of the derivation of the method, we decompose the matrix X into more computationally tractable matrices;

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T. \tag{5}$$

Using this definition, it can be shown that

$$\begin{aligned} \tilde{\mathbf{y}}_{\text{OLS}} &= \mathbf{X}\boldsymbol{\beta}_{\text{OLS}} \\ &= \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}\left[\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\right)^T \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\right]^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{U}\mathbf{U}^T\mathbf{y}. \end{aligned} \tag{6}$$

The OLS method is thus a simple and straight forward method which is surprisingly powerful even without a regularization term, which is the key difference between it and Ridge regression.

2.2. Ridge regression

A common problem in regression and machine learning is over-fitting. When the complexity of the model increases, the system tries too hard to fit the training data, ultimately increasing the error on the test data. Ridge regression attempts to combat this by introducing an alternative cost function

$$C(\boldsymbol{X}, \boldsymbol{\beta}) = \frac{1}{n}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2. \quad (7)$$

With the addition of a regularization term, quantified by the hyper-parameter λ, we penalize high values of β. This effectively simplifies the solution, making the model less flexible, reducing the variance. The drawback, however, is that this also increases the bias if the *shrinkage parameter* λ is not chosen optimally. As we will later discuss, when increasing the complexity of our model, our system becomes increasingly susceptible to over-fitting the training data. With the implementation of the regularization term, we force the system to converge on a "simple" model, effectively reducing the variance.

2.3. Lasso regression

The last of the trio is Lasso regression. It is similar to Ridge in the way that it adds a regularization term, but more aggressively so. This becomes evident when we once again look at the cost function

$$C(\boldsymbol{X}, \boldsymbol{\beta}) = \frac{1}{n}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1. \quad (8)$$

Here, the regularization comes on the form of L^1, which forces β-values to zero for sufficiently large values of λ. This creates a sparse β-array, as only certain polynimials (in our case) are activated, as opposed to Ridge regression, where high-degree polynomials with very low β-values are still present.

3. DATA

In this project we have tested the three previously discussed regression methods on two different data sets. First, we fit a polynomial to the *Franke function*, a common test function for such analyses. Last, we assess our polynomial regression fits on real world terrain data. In both cases the data is split into a training data set and a test data set, more on this in the next section.

3.4. The Franke Function

The Franke function is a two dimensional function developed to test interpolation techniques. However, its geometric features are well suited for our surface regression analysis as well.[3] Mathematically, the Franke function is expressed as

$$\begin{aligned} f_{\mathrm{F}}(x,y) = &\frac{3}{4}\exp\left\{\frac{-1}{4}\left[(9x-2)^2 + (9y-2)^2\right]\right\} \\ &+ \frac{3}{4}\exp\left\{\frac{-1}{49}(9x+1)^2 + \frac{1}{10}(9y+1)^2\right\} \\ &+ \frac{1}{2}\exp\left\{\frac{-1}{4}\left[(9x-7)^2 + (9y-3)^2\right]\right\} \\ &- \frac{1}{5}\exp\left\{\frac{-1}{4}\left[(9x+4)^2 + (9y-7)^2\right]\right\}. \end{aligned} \quad (9)$$

A surface plot of the Franke function can be seen in figure 2 evaluated over $x, y \in [0, 1]$.

3.5. Terrain data

The other data set employed in this study is taken from the U.S. Department of the Interior Geological Surveys (USGS) EarthExplorer website. The data is gathered from a Shuttle Radar Topography Mission (SRTM), an instrument on board the space shuttle Endeavor, which used radar interferometry to map out terrain. More specifically, the area of land we are using is that of Møsvatn in Telemark visualized in figure 8. In our analysis, we have chosen to down-sample it by averaging to a 60×30 matrix grid, which hopefully still carries distinctly different geometrical properties to that of the Franke function.

4. RESAMPLING AND BIAS VARIANCE

Making predictions can be hard, and even harder with insufficient data. Therefore, effective resampling methods are vital to modern statistics. Resampling is the process of refitting the model to different samples of ones data set. By doing this, we can obtain additional information about our model fits than we would from fitting just once as we can learn something about how our fit varies within our data set. One drawback to this, however, is com-

putational expense, which again, for our study is not an issue.

As previously stated, we split our data set randomly into two main sets; the training data containing 80% of all data and the test data, the remaining 20%. In order to make sure these are representable samples of our data set, as well as studying the variance in our predictions, we apply what is called *K-Fold Cross-validation*, which splits the data into k number of equally sized but mutually exclusive subsets, meaning that when one set is used as test data, the rest of the data is used for training. Once the model is fit to one of the k-fold test data sets, another one of the subsets are chosen, and the rest are again used as training data, So that in the end, all subsets have been used as test data once.

This method allows us to study the variance of the β coefficients, and calculate a confidence interval, which is useful in determining the reliability of our model. Furthermore, we test our model fit using a Mean Squared Error (MSE), as defined in our discussion of OLS. We also calculate the coefficient of determination R^2 which is closely related to the MSE, we will focus on MSE for our study. During model assessment and selection; determining the optimal flexibility of our model, we study how the error behaves. This is where the *Bias-Variance tradeoff* comes in. In order to determine the optimal hyper-parameters and model complexity for our system, we need to look at how the MSE responds as a function of model complexity. By decomposing the MSE as

$$E\left[(\boldsymbol{y}-\tilde{\boldsymbol{y}})^2\right] = E\left[(\boldsymbol{y}+\boldsymbol{f}-\boldsymbol{f}-\tilde{\boldsymbol{y}})^2\right] \tag{10}$$

$$=\frac{1}{n}\sum_{i=1}^{n}\left\{\overbrace{E\left[(y_i-f_i)^2\right]}^{=\sigma^2}+E\left[(f_i-\tilde{y}_i)^2\right]\right. \tag{11}$$

$$\left.+2\overbrace{E\left[y_i-f_i\right]}^{=0}E\left[f_i-\tilde{y}_i\right]\right\} \tag{12}$$

$$=\frac{1}{n}\sum_i(f_i-E\left[\tilde{\boldsymbol{y}}\right])^2+\frac{1}{n}\sum_i(\tilde{y}_i-E\left[\tilde{\boldsymbol{y}}\right])^2+\sigma^2, \tag{13}$$

we can identify which terms contribute to the total error. In this equation, the first term is the Bias, the second is the variance the σ is the irreducible error of our model, or the variance in the true mean inherent in our data. In order to split the terms in the last equal sign, we have again used the trick of adding and subtracting.

If we choose a model with low flexibility, in our case a low-degree polynomial, we will under-fit our data, and the our bias will be high, because the distance between our predicted points and the true ones are large. However, if we increase our model complexity too much, our predicted points will match the true training data, but we will be fitting to the noise. This becomes evident when we look at the difference between the test error and the training error (our model is not generalizable). We will discuss the bias-variance in further detail for our three approaches in the results section.

5. RESULTS

In this section we will apply our three regression methods to the two data sets. First, we will discuss their ability to fit the Franke function and their general behaviour and tuning process before we apply them to real world terrain data.

5.6. Franke function data

Ordinary Least Squares

Because the OLS method has no dependence on hyper-parameters other than the number of folds and complexity of the polynomial fit, we will study the interaction between complexity and data noise in the Franke Function. The first thing we did, was to fit a polynomial using the SVD method of OLS on three different sets of data generated by the Franke function. The difference between these was the amount of Gaussian noise applied to the system. By calculating the training and test errors over 15 polynomial complexities, we can look at the OLS' response in terms of Bias-Variance for a dataset with a given noise level. We can hence, on a per noise-level basis determine the model response as seen for a dataset with Gaussian noise $N(0, 0.05)$ in the top of figure 1. Looking at this figure, it becomes obvious that in order to minimize the test set error, we want to choose a polynomial complexity between 5 and 10. Because we want to minimize the variance as well, we chose a polynomial of degree 5 to be the optimal parameter for

this data set (it is also what you asked for in the project description...).

In order to determine the OLS methods response to noise, we apply our fit to several different noise levels, three of which ($N(0, 0.1)$, $N(0, 0.05)$ and $N(0, 0.01)$) are plotted in figure 2. Additionally, we do a an error plot per polynomial degree can be seen in the bottom plot of figure 1, where we can see how the bias, not surprisingly decreases when we reduce the noise, and the amount of overfitting on high polynomial degrees is not as dramatic.

Ridge regression

Next, we look at the effect of an additional regularization term on our model fitting procedure. First off, we note that overfitting and high variance only became a problem at very high model complexities, we therefore do not expect a drastic improvement using the test data that we already got with OLS.

First and foremost, we explore the dependence on the shrinkage parameter. As we can see by the model fits in figure 5, increasing the λ, or shrinkage parameter, the model is simplified, as we penalize high values of beta. This can again be seen in figure 6, where the variance of the betas decrease with the shrinkage parameter. Furthermore, our error plot, figure 3, shows that we no longer get a sharp increase in error for large complexities. However, the overall error is lower using a simple OLS method for this data set, but the ridge method could potentially prove useful in a case where there is danger of over-fitting.

Lasso regression

Lastly, we look at Lasso regression. Applying it to the Franke function, we see that its behaviour is very similar to that of Ridge regression, which is expected. Since both methods employ a regularization term to constrain the beta values, we see on the right in figure 6 that each beta converge towards zero, but as opposed to Ridge, the Lasso values actually go to true zero. Furthermore, 5 shows some of the Lasso fits to the Franke function, and how large values of the shrinkage parameter sets all β-coefficients to zero. Additionally, 3 shows that very low λ-values, show the best result, and that we don't gain much from increasing the polynomial degree past 5, however, we note that as with Ridge, we no longer fit to noise for large model complexities as we did for OLS.

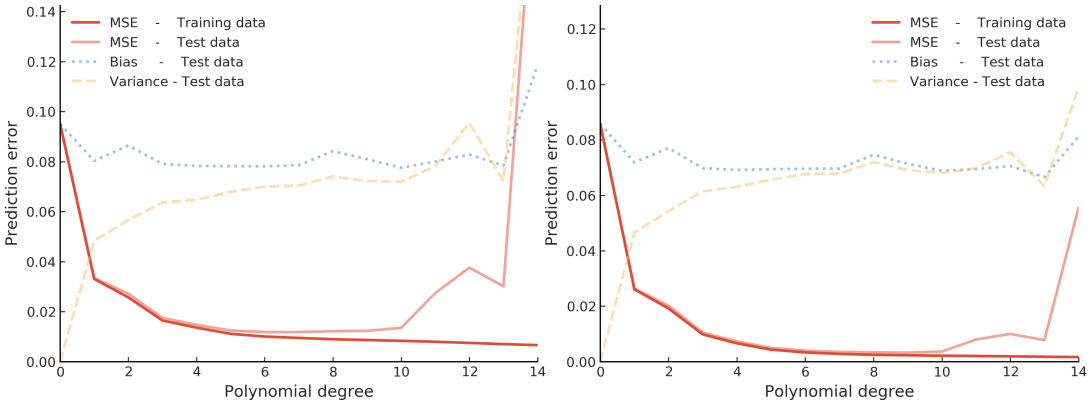
5.7. Terrain data

In this section we apply all our methods to the terrain data. The motivation for applying our model to these data is so that we can properly study the benefits of Ridge and Lasso regression. As we saw in the previous analysis, OLS won out, because we never require high enough polynomial degree so that overfitting becomes a significant problem. However, the terrain data has less predictable slopes and higher noise levels, as can be seen in figure 8.

For this study, we have applied our methods for many different polynomial degrees and λ values, only a few of which are presented here. First, we looked at the error response per λ value, which revealed that we have no significant decrease for Ridge nor Lasso past $\lambda = 10^{-5}$. The error over polynomial degrees using this value of the shrinkage parameter can be seen in figure 7. As we can see, OLS has a positive trend, but breaks down at degrees higher than 8, and is beaten by Ridge regression! Although Lasso never collapses like the other two, it seems to converge on a higher MSE for very high polynomial degrees. The figure suggests that decreasing the λ value further could allow for a more complex model fit for Lasso, but this had minimal effect. Another visualization of the behaviour of our models at a polynomial complexity of degree 10 can be seen in figure 8, where we see how OLS decides to party instead.

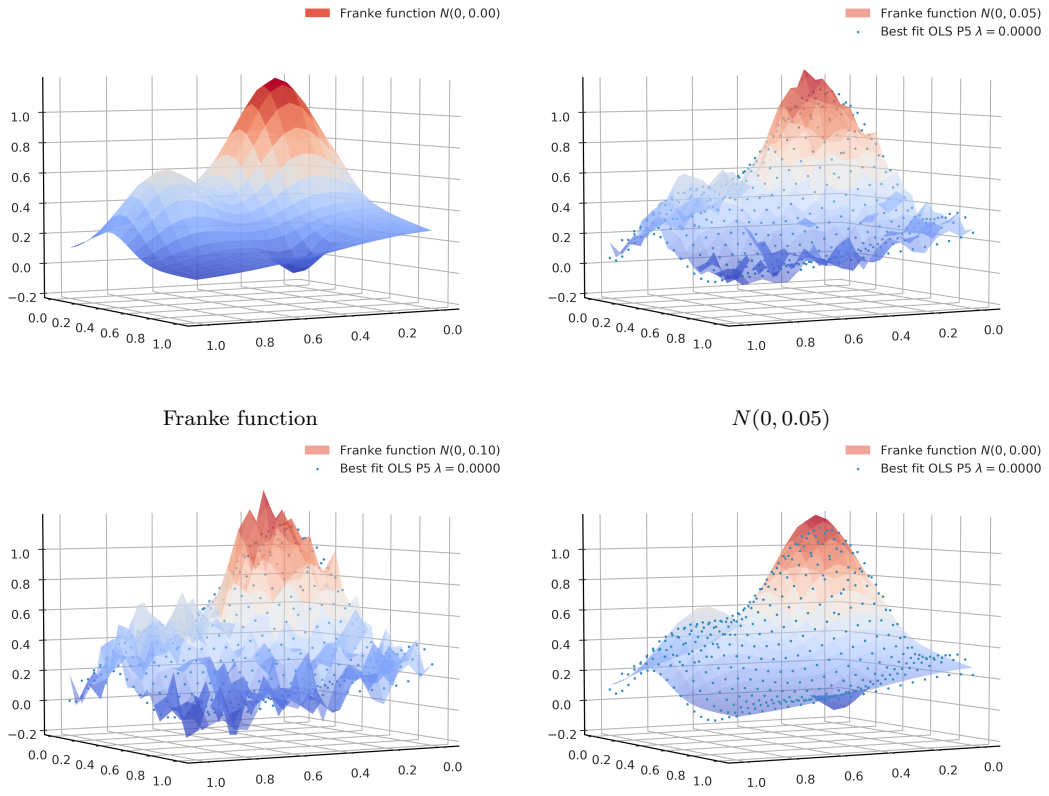
6. SUMMARY REMARKS

In this project we have looked at three different regression methods, applied to two different types of data. First, we looked at how our methods respond to a nicely behaved Franke function, before we applied it to the jagged and unpredictable behaviour of Møsvatn. We conclude that for the simple Franke case, the OLS comes out as the winner by far. The additions of regularizing terms does not seem to have a major impact even considering very high noise levels. However, when applied to the terrain data, OLS loses to Ridge regression, which is amble to constrain the β values to reasonable values for longer. Lasso, however, is too crude



$N(0, 0.1)$ $N(0, 0.05)$

FIG. 1. MSE error, bias and variance when using OLS to fit two different datesets with different noise levels. The model is fitted using a polynomial ranging from degree 0 to 15. Not surprisingly, the bias is reduced for the data with lower noise, and we get less overfitting.



Franke function $N(0, 0.05)$

$N(0, 0.1)$ No noise

FIG. 2. The Franke function surface and OLS Regression applied to it with varying degrees of noise. Even with high noise, OLS does a good job of modelling the Franke function.

and strict on its β's, resulting in under-performing predictions and inability to adapt to the real world, depression and insecurity.

REFERENCES

[1] Morten Hjorth-Jensen. *Lectures Notes in FYS-STK4155. Data Analysis and Machine Learning: Linear Regression and more Advanced Regression Analysis*. Sept. 2019.

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

[3] Richard Franke. *A critical comparison of some methods for interpolation of scattered data*. Tech. rep. Monterey, California: Naval Postgraduate School., 1979.

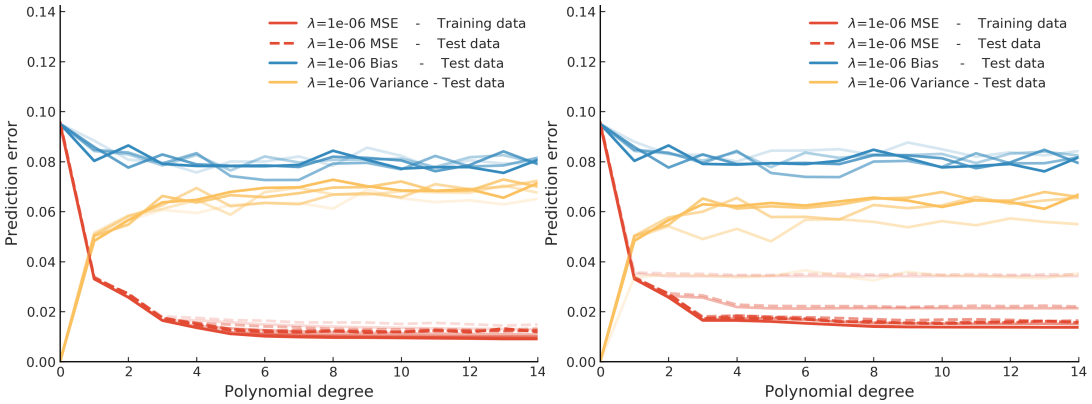


FIG. 3. MSE error, bias and variance when using Ridge and Lasso regression with different λ parameters on the Franke function with Gaussian noise $N(0, 0.1)$. Plotted using $\lambda = [10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$ over 15 polynomial degrees.

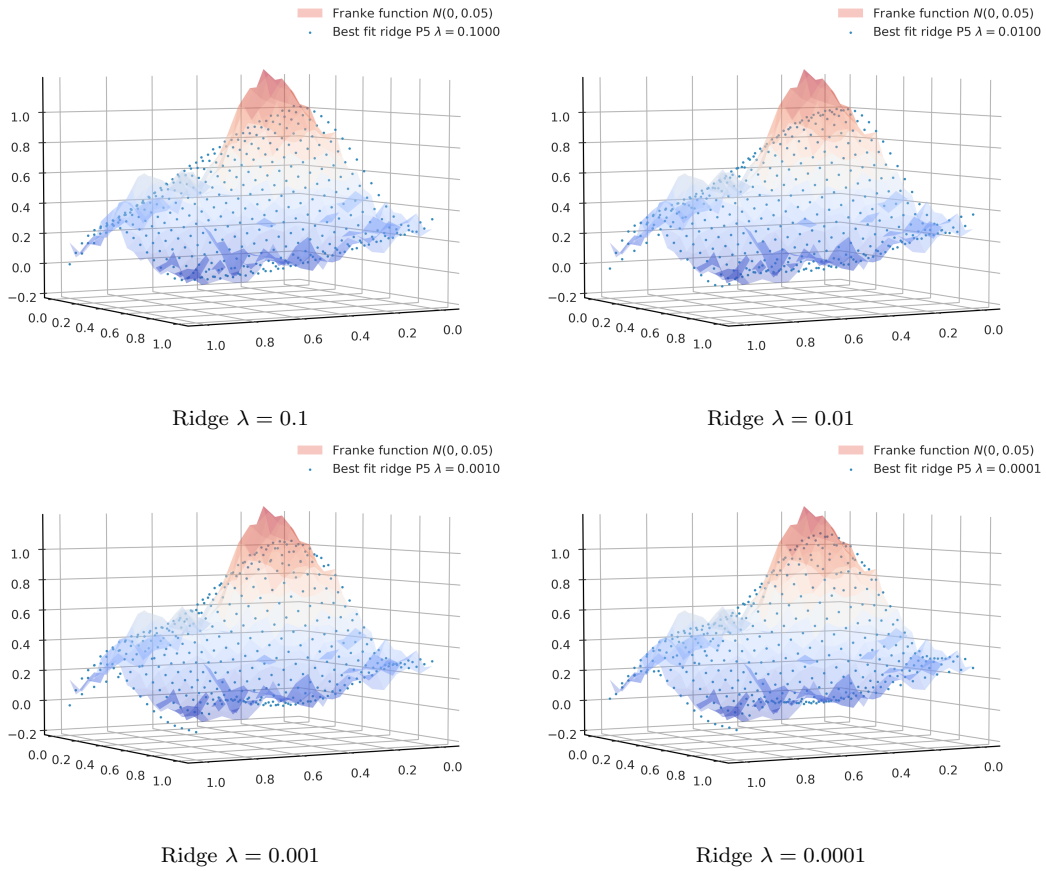
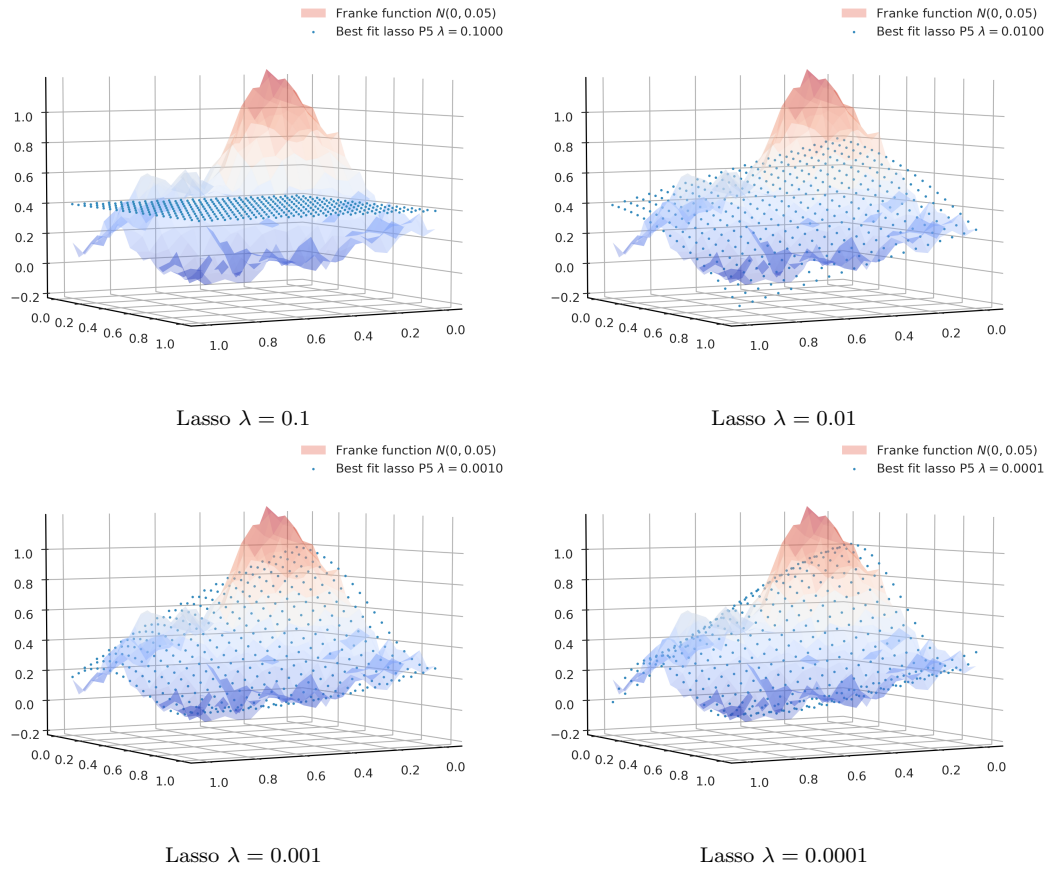


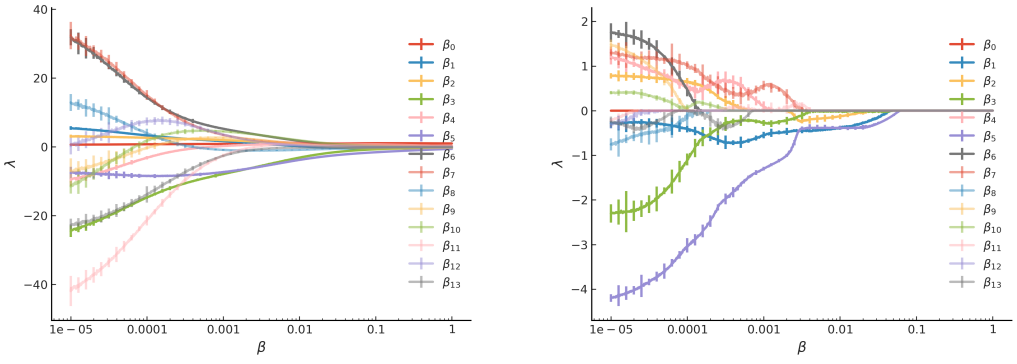
FIG. 4. Ridge Regression for four different values of the λ parameter. For this analysis, we obtain the best fits with small values of λ, which brings us closer to OLS.



Lasso $\lambda = 0.1$ Lasso $\lambda = 0.01$

Lasso $\lambda = 0.001$ Lasso $\lambda = 0.0001$

FIG. 5. Lasso Regression for four values of λ. Even for small λs, the fits are very crude.



Ridge regression Ridge regression

FIG. 6. Values for the 14 first β-values using a 5th degree polynomial fit for different values of the shrinkage parameter λ using Ridge and Lasso regression. Confidence per parameter is calculated across all K-Folds, and plotted as error bars.

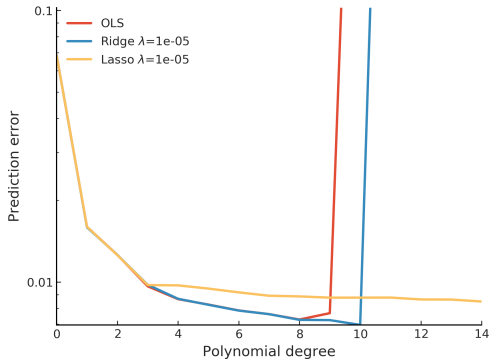


FIG. 7. The MSE error for the terrain fits. We note how OLS collapses at polynomial degree 8, and is beaten by Ridge regression which carries on to degree 10. Although lasso regression does not collapse in the way Ridge and OLS does, it seems to converge on a higher MSE for very high complexities.

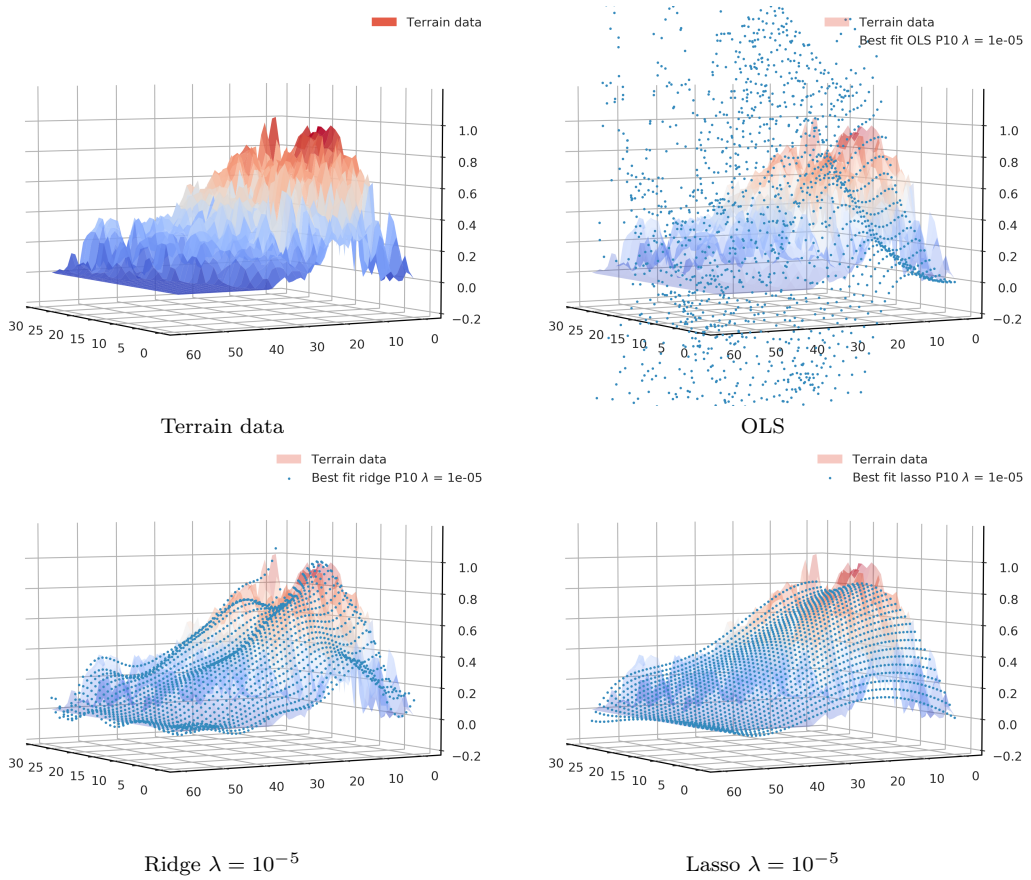


FIG. 8. Polynomial fits on terrain data for a degree 10 polynomial. As we can see, this time, the OLS method collapses above degree 8, while Ridge regression makes a good fit and Lasso struggles to capture the details.