Regression and Resampling methods

FYS-STK4155: Project 1

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

October 3, 2019

Abstract

Contents

1. INTRODUCTION

In this project we explore the problem of regression analysis, resampling and optimal hyperparemeters. Regression is the process modelling a response y through the parameterization of a predictor variable x ???. This is a powerful tool, as it gives us the ability to minimize the difference between observed data y and predicted data $\tilde{y}$ and solve for an optimal predictor. 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 responsecan 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 hyperparemeters and their data dependence.

2. REGRESSION METHODS

In this section we outline the basics of three different regression methods which we later apply to our two data sets.

2.1. Ordinary Least Squares (OLS)

As briefly mentioned in the introduction, a Linear regression system revolves around modeling a function $\mathbf{yX}$, 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 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 details of the method, it works by decomposing the matrix X into more computationally tractable matrixes;

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

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 overfitting. 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, quantized by the hyperparameter 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

which penalizes high values of β according to the hyperparamter counteracting the effect of overfitting. As we will later discuss, when increasing the complexity of our model, our system becomes increasingly susceptible to overfitting 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.

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. 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 **??** 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 Geolocial Surveys (USGS) EarthExokirer website. The data is gathered from a Shuttle Radar Topography Mission (SRTM), which ??. More specifically, the area of land we are using is that of Møsvatn in Telemark visualized in figure **??**.

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 prosess 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. One drawback to this, however, is computational 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. 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 hyperparameters and model complexity for our system, we need to look at how the MSE. If we choose a model with low flexibility, in our case a low-degree polynomial, we will underfit 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 hyperparameters 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.

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 4, 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 usefull in a case where there is danger of overfitting.

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 regular-

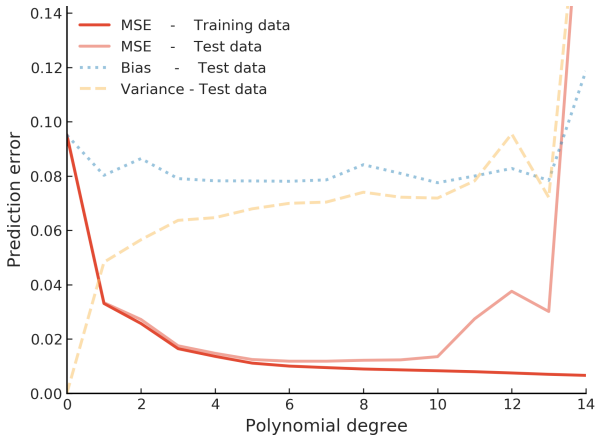
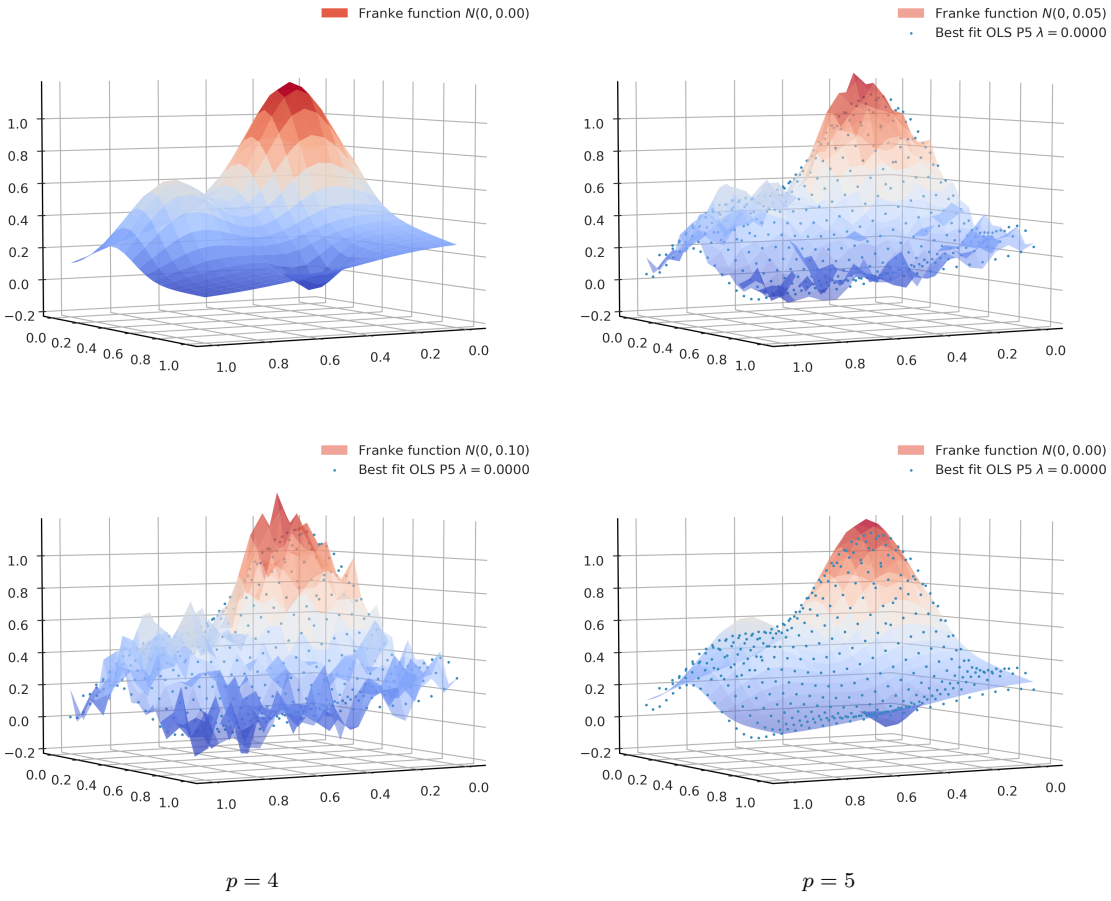


FIG. 1. MSE error, bias and variance when using OLS to fit two different datesets with different noise levels. The model is fitted using a polynomial raning from degree 0 to 15.

ization 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. Adittionally, 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

6. SUMMARY REMARKS



$p = 4$ $p = 5$

FIG. 2. OLS Regression.

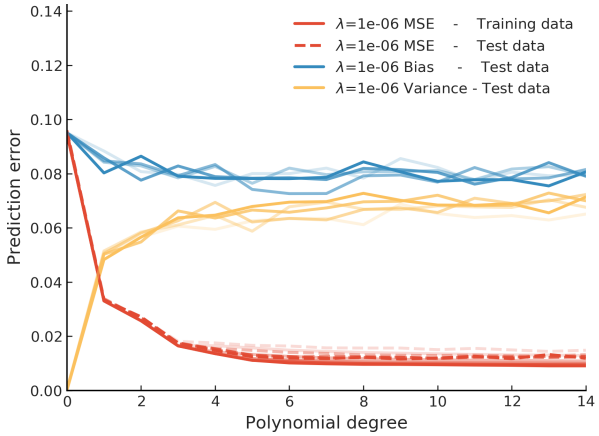
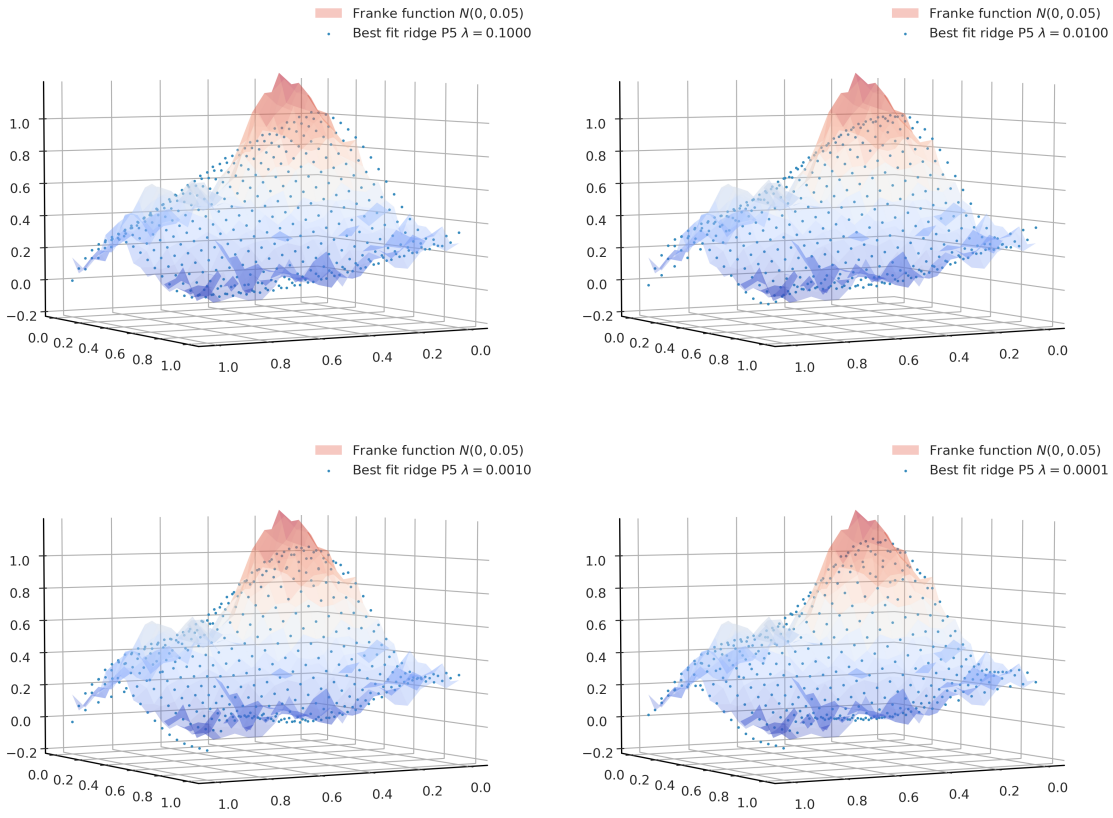
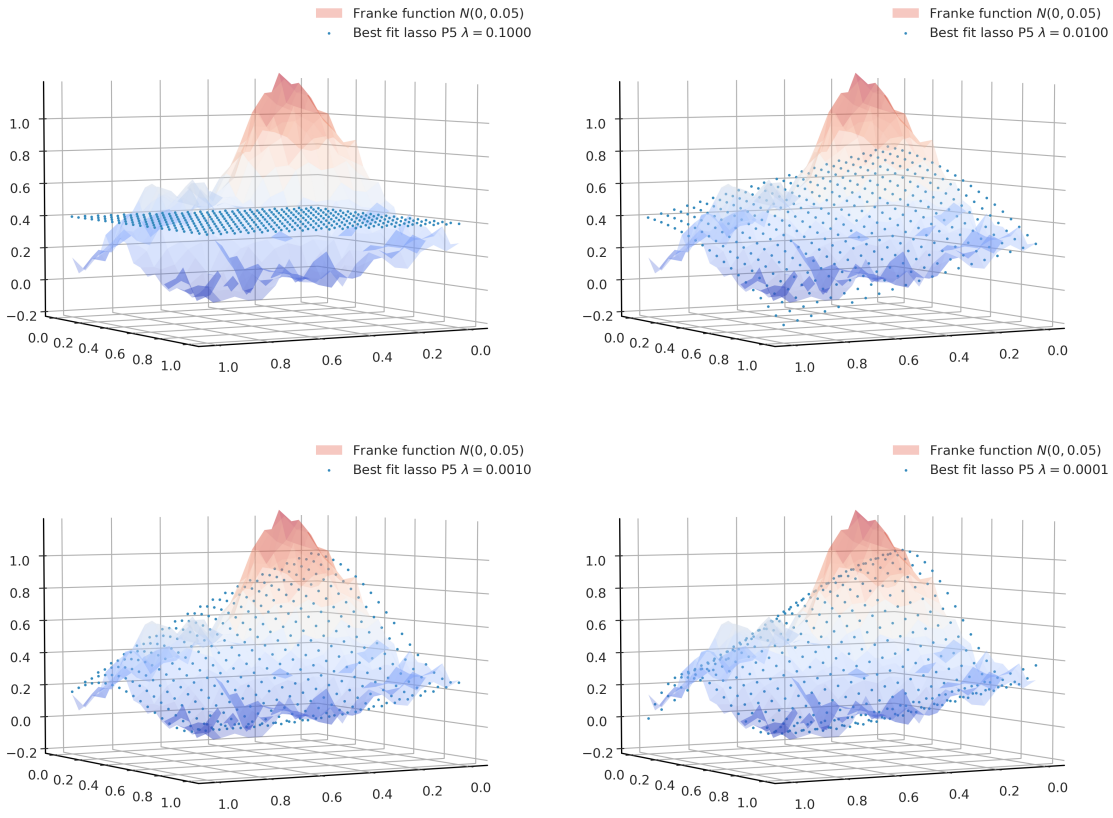


FIG. 3. MSE error, bias and variance when using Ridge and Lasso regression with different λ parameters on the Franke function with gaussian noise $N(0, 0.1)$. The λ values used are 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3} and 10^{-2} plotted with decreesing opacity. The model is fitted using a polynomial raning from degree 0 to 15.



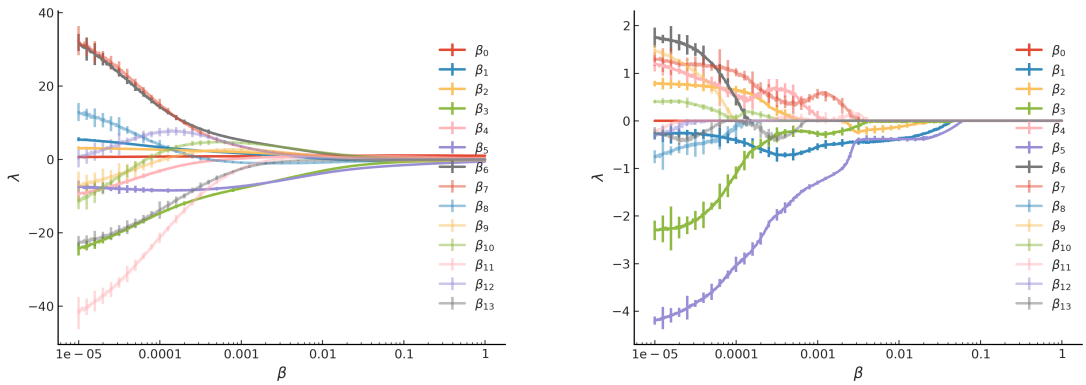
$\lambda = 0.001$ $\lambda = 0.0001$

FIG. 4. Ridge Regression.



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

FIG. 5. Lasso Regression.



Ridge Lasso

FIG. 6. Values for β for different values of the shrinkage parameter λ using Ridge and LASSO regression. Confidence per parameter is calculated across all K-Folds, error bars 1 σ confidence.