Regression and Resampling methods

FYS-STK4155: Project 1

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

September 29, 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.

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.

4. RESAMPLING AND BIAS VARIANCE

Flexible models have higher variance, because they tend to overfit.

5. APPLICATION TO REAL DATA

6. COMPARISON

Fitting to the Franke function

7. DISCUSSION

8. SUMMARY REMARKS

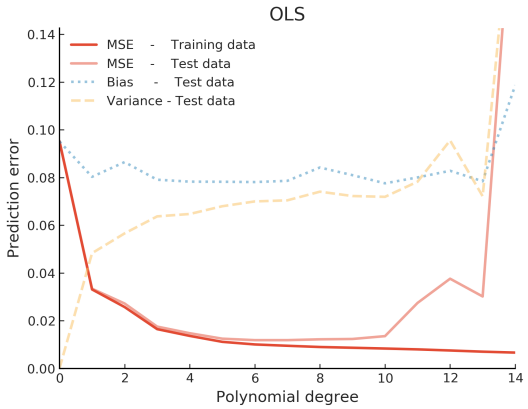


FIG. 1. Stuff

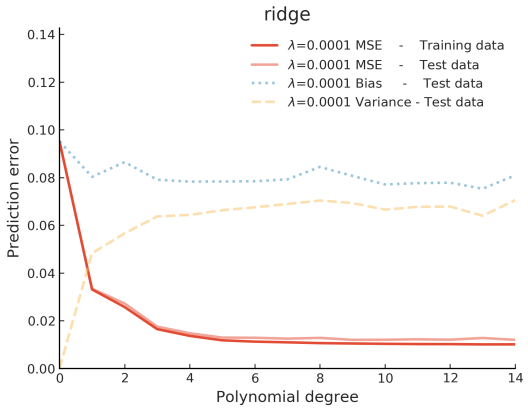


FIG. 2. Stuff

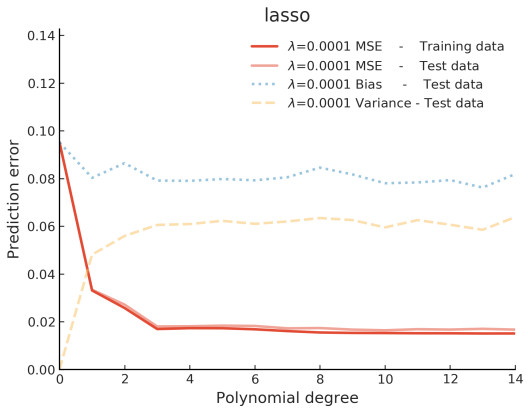
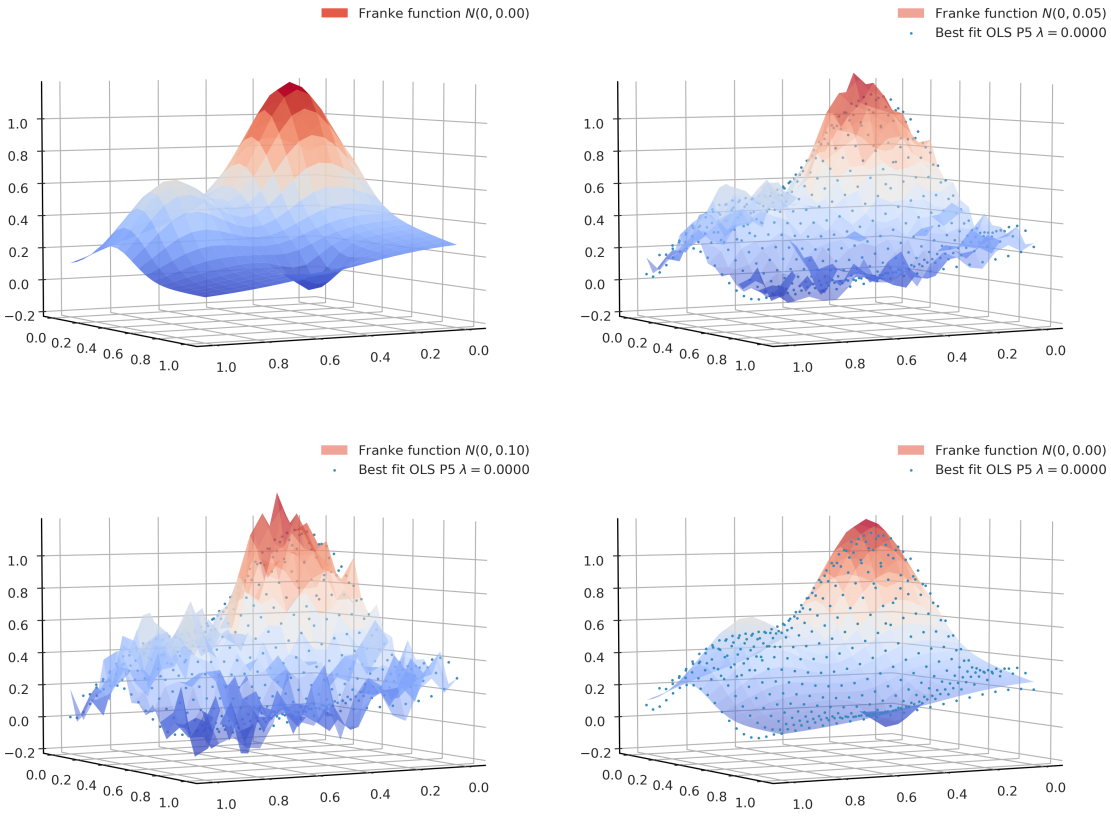
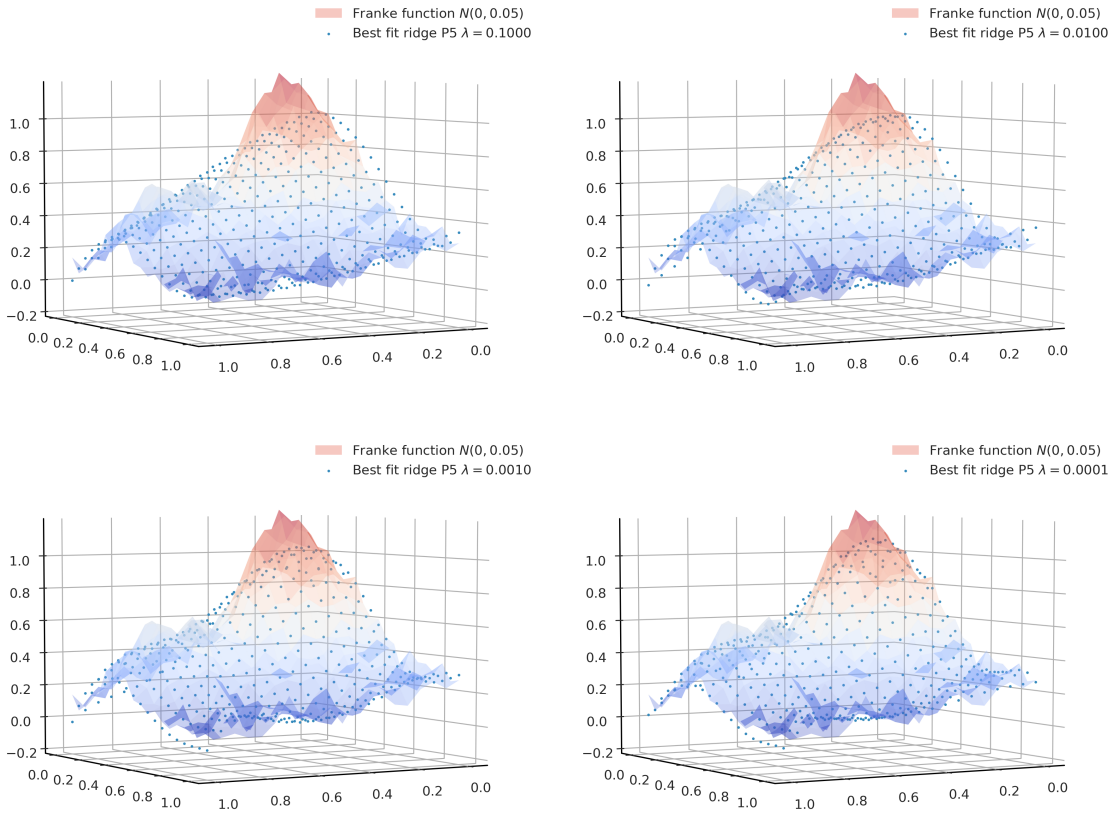


FIG. 3. Stuff



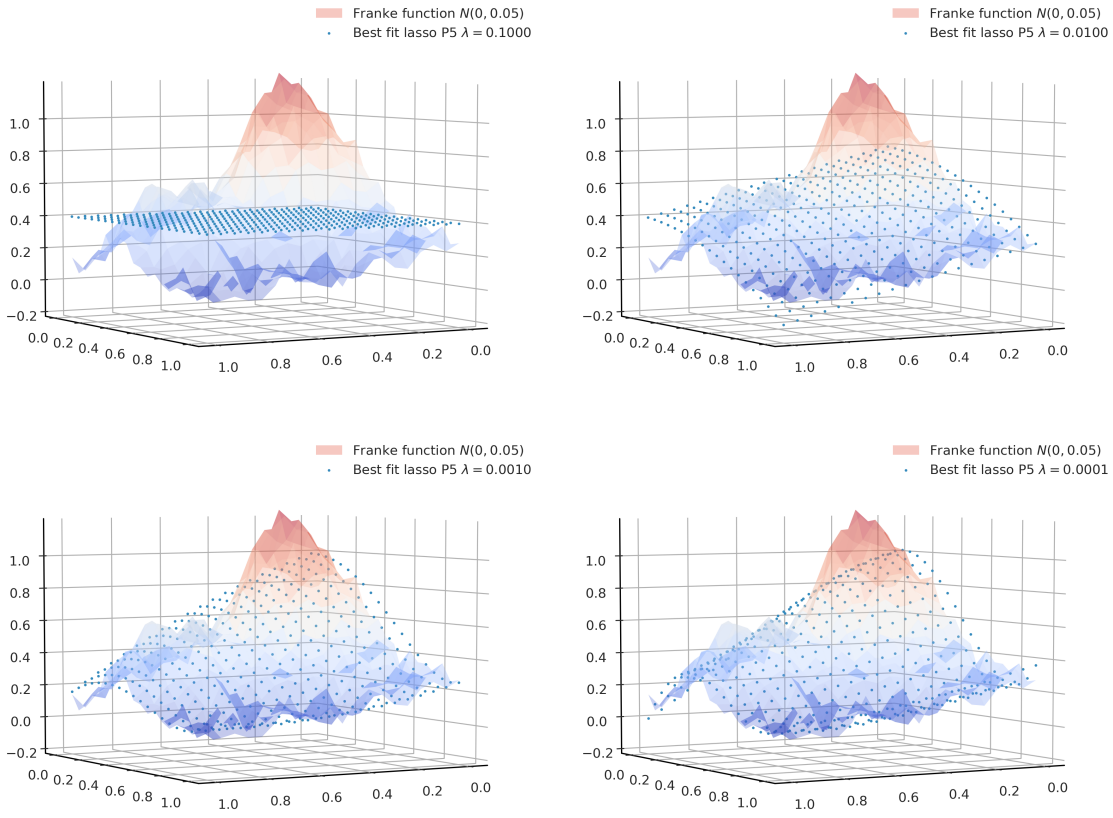
$p = 4$ $p = 5$

FIG. 4. OLS Regression.



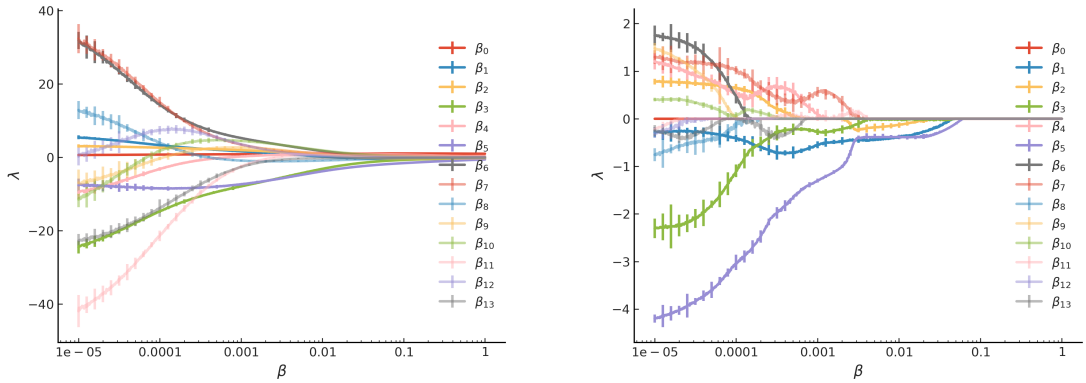
$\lambda = 0.001$ $\lambda = 0.0001$

FIG. 5. Ridge Regression.



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

FIG. 6. Lasso Regression.



Ridge Lasso

FIG. 7. 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.