FYS-STK3155

Dag Arne Lydvo
(Dated: October 10, 2020)

In this project I will analyzing three different regression methods, the ordinary least square, Ridge and Lasso. The regression models was scored on two different datasets, one given by an analytical function, the Franke function with some stochastic noise, the other a terrain dataset from an area close to Stavanger, Norway. The findings show a improvement of the performance of the model with regularization for the analytical sourced dataset, but that this benefit was not present in the terrain data.

I. INTRODUCTION

The goal of this project is to analyse three different methods of linear regression using two different resampling methods, bootstraping and cross-validation. The methods of linear regression to be used are Ordinary Least Squared, Ridge and Lasso regression. These methods will be used first on a created dataset given by the Franke function on a set of x and y inputs. Features will be set by a polynomial degree of x and y values as we try to train and predict the Franke function. Second part of the project will be concerning a real dataset of terrain data in the form of a grid x and y with corresponding levels of elevation z.

II. METHOD

A. Datasets

We will be doing experiments with two different datasets. The first generated by the multivariate Franke function given by:

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

To this function we will be adding some stochastic noise given a normal distribution $N(0,1)$. The Franke function with and without noise can be seen in figures below.

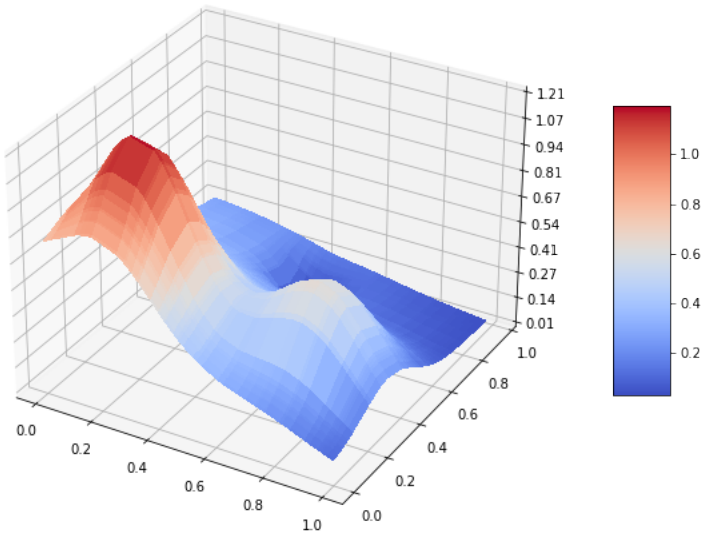

Figure 1. Franke function with no noise

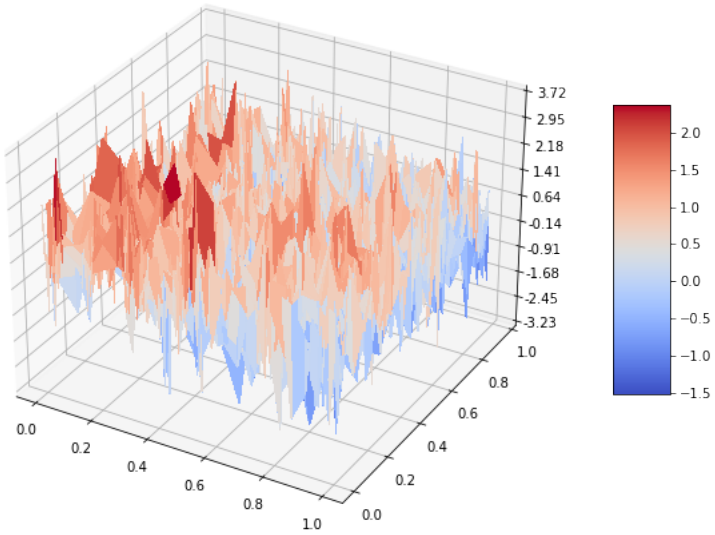

Figure 2. Franke function with stochastic noise N(0,1)

The second dataset will be a grid of digital terrain data from an area near Stavanger, Norway. The dataset is levels of elevations (z) given by a square of x and y values.

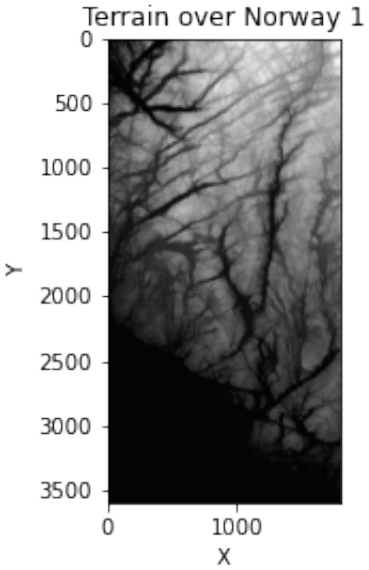

Figure 3. Second dataset: terrain data of a region near Stavanger, Norway

B. Regression methods

1. Ordinary Least Squares

The first method of regression is the ordinary least squares method. We will seek to fit a model to our data set with the parameters β. Given a data set of p features we will try to fit p numbers of β to our p columns of data using matrix inversion. [1] Given an equation for a model of the form:

$$\tilde{y} = X\beta$$

Where X is the matrix containing our features and $\tilde{y}$ is our predicted values. We seek to find the optimal β by minimizing the function

$$C(\beta) = \frac{1}{n}(y - X\beta)^T(y - X\beta)$$

This is given by

$$\frac{\delta C(\beta)}{\delta \beta} = 0 = X^T(y - X\beta)$$

If (X^TX) is invertible the matrix equation can be solved for β:

$$\beta = (X^TX)^{-1}X^Ty$$

2. Ridge regression

Ridge regression is very similar to method described above for OLS. To the matrix equation for β there is added a hyperparmeter λ.

$$\beta = (X^TX + \lambda I)^{-1}X^Ty$$

Where I is the identity matrix. The hyperparameter λ acts as a penalty to very big β and reward values close to 0. This makes the model smaller, also referred to as shrinkage, and helps with over fitting. [3]

3. Lasso regression

C. Bias-Variance trade-off

The mean-squared error can be decomposed to the components of bias and variance.

$$E[(y - \tilde{y})^2] = \frac{1}{n}\sum_{i=0}^{n-1}(y - \bar{y})$$

$$E[(y - \bar{y})^2] = \frac{1}{n}\sum(f - E[\bar{y}])^2 + \frac{1}{n}\sum(\bar{y} - E[\bar{y}])^2 + \sigma^2$$

Given $y = f(x_i) + \epsilon$:

$$\rightarrow E[(f + \epsilon - \bar{y})^2]$$

Through expansion we get:

$$E[(f - \bar{y})^2] + E[\epsilon^2] + E[(2f - \bar{y})\epsilon]$$

$$E[(f - \bar{y})^2] + \sigma$$

$$E[(f - E[\bar{y}] - \bar{y} + E[\bar{y}])^2] + \sigma^2$$

$$E[(f - E[\bar{y}])^2] + E[(\bar{y} - E[\bar{y}])^2] + 2E[(f - E[\bar{y}])(\bar{y} - E[\bar{y}])] + \sigma^2$$

$$(E[\bar{y}] - f)^2 + E[(\bar{y} - E[\bar{y}])^2] - 2(f - E[\bar{y}])E[0] + \sigma^2$$

$$(E[\bar{y}] - f)^2 + E[(\bar{y} - E[\bar{y}])^2] + \sigma^2$$

D. Measuring error

We will be measuring error using two different metrics. The mean squared error (MSE) and the R2 measure of error.

1. *Mean squared error (MSE):*

The mean-squared error is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where Y_i is the known measurement and $\hat{Y}_i$ is the predicted value to be tested. The MSE is equal to the expected value of the squared of the error of the prediction. [2]

2. $R^2 score$

The R^2 score is given by:

$$R^2(\hat{y}, \tilde{\hat{y}}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y})^2}$$

Where $\bar{y}_i$ is the mean value of the predicted values $\hat{y}$.

III. RESULTS

Starting of with looking at a simple ordinary least squared regression on the Franke function with noise for a grid of 10 x,y values ranging from 0 to 1. The function for the mean-squared error and the R2 seem to be validated by the Scikit-learn function for polynomial degrees of 1 to trough 5.

The model was tested for complexities ranging from 1 to 21 features for 100 trial for each state of complexity. The MSE was measured for the prediction of the test and train set, see plot below.

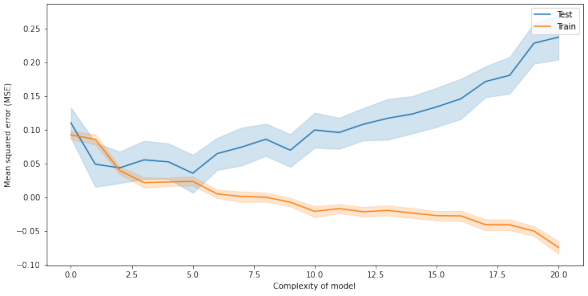

Figure 4. Range and mean of MSE for 100 trials over the complexity of the model

Using bootstrap as a resampling method with 100 resamples we do a bias-variance analysis for different numbers of datapoints.

Next result is from using K-folds cross-validation as a resampling technique. Results are MSE for 21 features with $10x10$ grid which gives 100 datapoints. The MSE score for the bootstrap method for 100 datapoints was 3.2953.

Number of folds	MSE	Scikit-learn MSE
5	25.418494	25.418494
6	1.805152	10.518405
7	4.441928	4.573210
8	2.212151	9.337058
9	4.374121	4.454320
10	10.793917	10.793917

Table I. K-fold cross-validation MSE scores compared to cross-validation scores from the Scikit-learn functionality.

Complexity	MSE	Bias	Variance
21	3.295277	0.955321	2.339956

Table II. Bias-Variance results with bootstrap for 21 features.

Using bootstrapping we analyze the Ridge regression method for different values of lambdas λ over the complexity of the model.

Next result are Ridge regression with use of cross-validation with K-folds ranging from 5 to 10 over the same span of lambdas λ with complexity of the model being 21 features.

Following are results from the experiments using terrain data. Ordinary least squares, Ridge and Lasso regression methods will be analyzed. And the resampling method used is K-fold cross-validation.

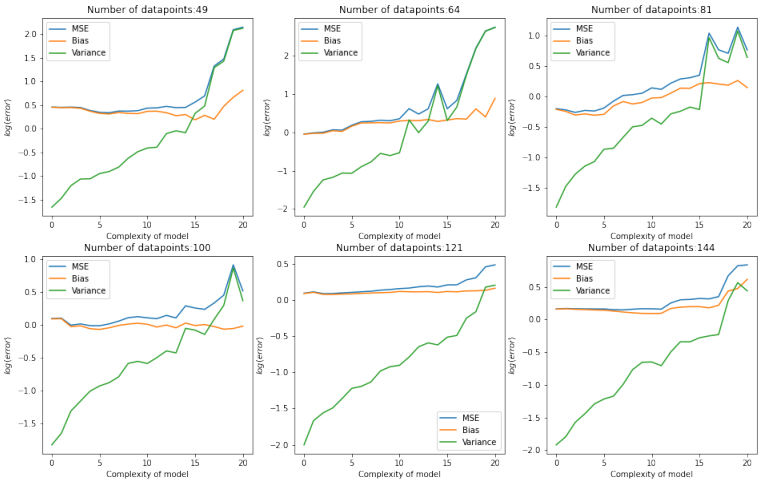

Figure 5. Bias-Variance analysis for different numbers of datapoints, log scaled.

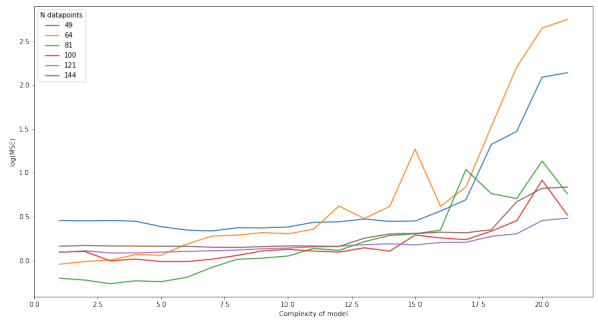

Figure 6. MSE score for different numbers of datapoints, log scaled.

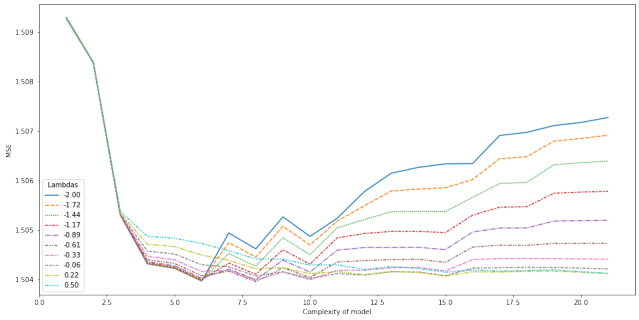

Figure 7. Ridge regression:MSE score for different values of λ, using bootstrap.

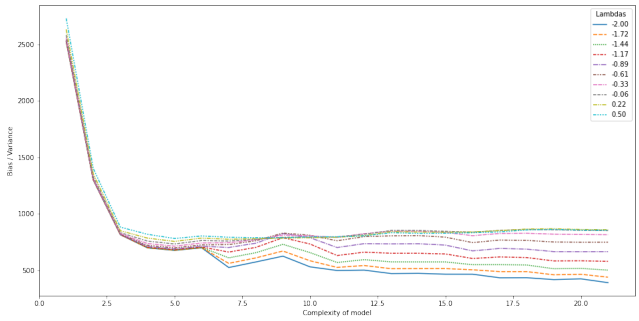

Figure 8. Ridge regression:The ratio of Bias over Variance for different values of λ.

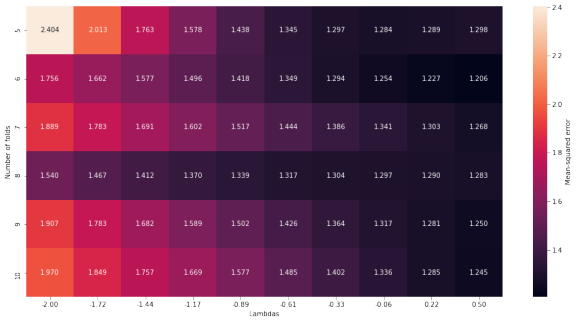

Figure 9. Ridge regression: Cross-validation scores for number of folds and lambdas λ

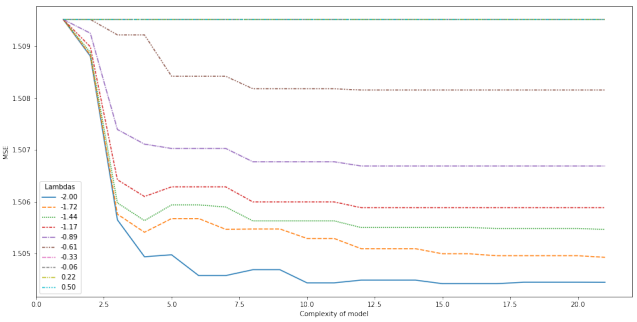

Figure 10. Lasso regression: MSE scores using bootstrap resampling for a range of lambdas λ

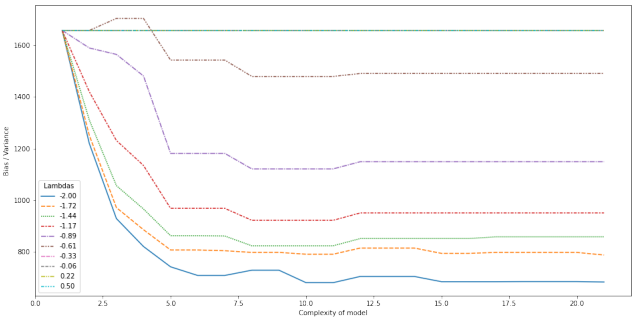

Figure 11. Lasso regression: Bias-Variance ratio for a range of lambdas

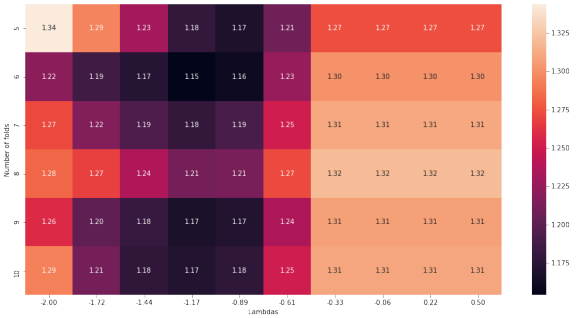

Figure 12. Lasso regression: Cross-validation scores for number of folds and lambdas λ

IV. DISCUSSION

Starting with the ordinary least squared method we see that a comparison of the predictions on the test and train data produces a MSE score that diverges as the complexity of the model increases, this is expected. Using the bootstrap method for resampling with 100 resamples we analyse Bias-Variance trade-off for an array of different number of datapoints. We see that the Variance component of the MSE increases with the complexity of the model for all experiments shown in figure 5. What is noticable is that gap between bias and variance at high complexeties shrinks as the number of data points increases. At 144 datapoints the variance component is less than the bias at a complexity of 21 features.

Looking at only the mean-squared error we see that

		MSE
Lambda	Method	
-	OLS	48.033013
-2.00	Ridge	108.002575
-0.89	Ridge	111.307910
-0.61	Ridge	111.449049
-1.72	Ridge	112.574759
-1.17	Ridge	112.587732
-0.33	Ridge	113.472991
-1.44	Ridge	113.620977
-0.06	Ridge	117.225305
-1.72	Lasso	118.071532
-2.00	Lasso	119.029316
-1.44	Lasso	119.761310
-1.17	Lasso	121.478469
0.22	Ridge	123.422281
-0.89	Lasso	125.740256
0.50	Ridge	133.470841
-0.61	Lasso	148.378082
-0.33	Lasso	173.472939
-0.06	Lasso	189.608095
0.22	Lasso	220.195714
0.50	Lasso	245.139363

Table III. Terrain data: MSE scores using cross-validation, sorted lowest to highest.

	Ridge						Lasso					
N folds	5	6	7	8	9	10	5	6	7	8	9	10
-0.06	167.700000	149.300000	137.200000	139.500000	135.500000	117.200000	233.700000	199.100000	213.300000	212.900000	189.600000	194.500000
-0.33	167.600000	145.600000	136.200000	141.000000	135.900000	113.500000	221.400000	186.900000	189.500000	191.200000	174.700000	173.500000
-0.61	169.900000	146.200000	137.600000	143.600000	139.200000	111.400000	191.800000	175.900000	164.700000	163.600000	156.000000	148.400000
-0.89	175.400000	154.200000	142.700000	147.500000	145.500000	111.300000	176.800000	152.700000	141.000000	139.900000	139.200000	125.700000
-1.17	188.300000	170.900000	151.600000	152.500000	153.700000	112.600000	170.400000	153.900000	138.400000	135.400000	137.700000	121.500000
-1.44	208.600000	191.900000	161.400000	156.800000	160.700000	113.600000	186.600000	162.300000	144.400000	145.500000	140.800000	119.800000
-1.72	228.000000	209.200000	167.600000	157.700000	163.400000	112.600000	185.200000	168.400000	143.300000	147.400000	149.400000	118.100000
-2.00	238.900000	217.000000	168.000000	153.600000	159.400000	108.000000	186.300000	190.700000	160.300000	158.800000	162.400000	119.000000
0.22	171.900000	156.100000	141.300000	141.000000	138.400000	123.400000	276.200000	234.200000	236.100000	234.400000	220.200000	221.200000
0.50	180.300000	166.300000	150.000000	148.200000	146.000000	133.500000	379.300000	299.200000	275.400000	265.000000	248.800000	245.100000

Table IV. Terrain data: MSE scores using cross-validation for ridge and lasso regression.

	OLS
N folds	
5	200.969800
6	177.621580
7	101.594888
8	75.400293
9	70.649340
10	48.033013

Table V. Terrain data: MSE scores using cross-validation for ordinary least squares regression.

low numbers of datapoints produces higher errors generaly and especially at higher complexities. 121 datapoints seems to be producing the lowest error at high complexities and seems also to be the most stable across complexities. Comparing the two experiments with the highest number of datapoints 121 and 144, at 144 the bias component starts to increase more at higher complexities.

The K-fold cross-validation were run on 100 datapoints and a complexity of 21 features. The calculated MSE does not match completely with the Scikit-learn calculations. With experiments ranging from 5 to 10 folds there are some that match very accurately and some that are off by a lot, especially for 6, and 8 folds. This seems somewhat strange, could be some error in the KFold-Split function. The best calculated score is from using 6 folds with a MSE of 1,8, but with a Scikit-learn score of 10.5. A quite big gap. 7 and 8 folds score about 4.4 to 4.5 across both scoring calculations, but not as good as the bootstrap score of 3.29.

Ridge regression using bootstrap was run with 100 data points and analyzed over a range of complexities of 21 features. Here there seems to be a strong dependency on the lambda parameter, especially as the complexities increases, see Figure 7. Running 10 different experiments with a range of lambdas from log(-2) to log(0.5) shows and increasing higher error for the low end of the lambda range as complexity increases. As figure 8 shows the bias over variance ratio decreases with complexity for the lower range of lambdas, while for lambdas of log(0.5) and log(0.22) it stays comparatively stable. For cross-validation scores the best scores are consistent with the bootstrap methods in that the higher MSE scores are found at the lower range of lambdas. The best score of 1.206 are found at lambda of log(0.5) and using 6 folds for cross-validation.

The Lasso regression experiments yields MSE scores in very much the same range as the ridge experiments. Though there seems not be any increase in error as complexities increase as seen for the ridge regression. This resulting in that the lower range of lambdas, especially log(-2) performing quite well as seen in figure 10. Figure 11 showing the bias-variance ration shows a graph very similar to that of figure 10. The bias component of the error decreases with complexity and the variance not increasing significantly as in the case of ridge regression. The cross-validation shows somewhat the same dependency on lambdas as the bootstrap experiments do, where the higher end of the lambda range perform worse. But the best scores are to be found around $\lambda = log(-1.17)$. The cross-validation method produces MSE scores somewhat below those of the bootstrap method.

For the experiments using the terrain data from near Stavanger, Norway I choose a square of 20X20 from the terrain data resulting in a dataset of 400 data points. The experiments first errors a lot higher than for the Franke function, see Table III. The best score by far is produced by the Ordinary least square method with an MSE of 48, quite a bit lower than number 2 which is Ridge regression with $\lambda = log(-2)$. Other observations is that the ridge regression seems to perform a bit better than the Lasso regression. As seen in Table V, the increase in number of folds produces better scores, from a score of 200 for 5 folds down to 48 at 10.

V. CONCLUSION

From this experiment we see that for the analytical function, the Franke function, the benefits or regu-

larization as the Ridge and Lasso regression methods produces better performance for the models than the OLS method. Bootstrap as a resampling method also seems to perform better than K-fold cross-validation. But I do suspect there is some mistake in my code for the cross-validation. Lasso regression seems again to produce a reduction in error as complexity increases without cost as seen for the ridge regression.

For the terrain data there are a lot higher errors observed and this real data seems to be harder to fit than the analytical function, something I would expect. There seems not to be benefits to regularization for the terrain data as was seen in the experiments for Franke function. The best performing method was the Ordinary least square method. With Ridge performing slightly better than Lasso. Higher number of folds for cross-validation seems also to produce lower errors across all regression methods.

REFERENCES

[1] https://compphysics.github.io/MachineLearning/doc/Projects
bs.html
[2] https://en.wikipedia.org/wiki/Mean$_squared_error$/
[3]https://www.quora.com/How-and-why-does-ridge-regression-help-with-overfitting