Project 1 (Paste into overleaf later)

**Abstract**

The aim of this project has been to study linear regression models by comparing the performance of Ordinary Least Squares (OLS), Ridge-, and Lasso Regression when applied to terrain data from an area in Norway. %Which area

These models are implemented in python, making use of libraries such as Numpy, MatPlotLib and Scikit-Learn. Initially, 4/5s of the data was used as a training set while the remaining 1/5 was used as a test set for validation. The terrain data was split using …..

This initial analysis showed mean squared errors (MSEs) of …. Meaning that …. performed better for this data set than the other regression methods.

Comparing the simple splitting of the data to errors retrieved after applying bootstrap- and k-fold cross-validation resampling methods revealed that … provided a better training of the data. Overall, …. Was the best model for the terrain data, with …. order

Polynomial and MSE = ....

%TODO: Fill in the data results

**Introduction**

When searching for the best way to model a data set, some assumptions must be made. Central to linear regression methods is the assumption that there exists some continuous function f(x) that describes the data (and that we can somehow find an approximation of this function. %Cite something

This project compares three linear regression methods: Ordinary Least Squares (OLS) - the simplest - and then Ridge- and Lasso Regression, which both introduce a dampening %shortening??

term: lambda.

This chapter outlines the mathematical theory behind these methods, while chpt. \ref{method} goes through the implementation details. The python code used for this project can be seen in the appendices. Appendix \ref{franke} contains the code used for writing and testing the functions used in the analysis. Calculations of MSE were first done for an identity matrix with lambda = 0 (for Ridge and Lasso), which results in a perfect fit. This was done to ensure the functions returned the expected results. Then, all functions were tested for a 3d surface using Franke’s function (eq. \ref{franke}). The results from the analysis of this surface are discussed in chpt. \ref{Developing and testing}.

Finally, chapt. \ref{Results} shows how the functions performed when applied to terrain data from …. , Norway. %Add the correct name for the data

Here, the same code was used, but with some minor variations:

* Scaling of the data:
  + No scaling was added for the Franke data, as the set was already fairly uniform. The MSE and R2-scores seemed reasonable without any further scaling or pre-processing of the data. The terrain data, on the other hand, has greater variations in z (reflecting the height above sea level) %right?! Making it difficult to interpret the MSE scores. Other than that, there are few sudden changes in the terrain data, so the ….. scaler was chosen %\cite!
* Orders of magnitude when testing polynomials:
  + For the real data, the calculations for higher order polynomials became too large to handle for my poor laptop %Write this properly
  + This limits the analysis, but hopefully the results still give insight into the behaviours of the different regression methods.

The code, where any changes were made, can be viewed in appendix \ref{appendix2}. Apart from these changes, the same code as for the Franke data was used.

**Theory – Linear Regression:**

**Error functions:**  
 As seen from eq 1, where y are the actual values, and y-tilde are the predicted values, MSE goes to zero for a perfect fit.

The R2-score seen in eq.2, where y\_overline is the mean of y, weights the difference between real and predicted value by the distance from the mean. Here, a perfect fit will result in an R2-score equal to 1.

%Something about which one is useful for what? MSE is susceptible to sizes of the data set, while r2 always has a max of 1. But how does r2 behave for great outliers – they become less important, right?

**Franke:**

The Franke function seen in eq. 3 is often used for testing models for surfaces. %cite something here

**Ordinary Least Squares:**

This is a simple and often-used method for making predictions. Eq. 2.1 from hastie et al p. 12. Least squares seeks to find the coefficients beta that minimises the residual sum of squares /cite hastie et al p. 12.

%Citation for the equation/derivation! (possibly hastie et al p. 24)

* What is y-hat in eq. 8? (the predicted outcome, I think)

**Ridge:**

* TODO: Move “this code was used” into the introduction.

Cite p. 61 in Hastie et al:

Ridge Regression is a shrinkage method (as discussed in hastie et al)(?). This is a way of limiting the importance of certain features (i.e. shrinking them).

Eq. 3.41 from hastie et al shows the Ridge equation for optimal coefficients. As they explain, lambda > 0 controls the shrinking, penalizing large coefficients, as these ….. %why?

**Lasso:**

Similar to Ridge, Lasso Regression is also a shrinkage method, but here the coefficients are reduced faster. (eq. 3.51 in hastie et al., p. 68). As well as all the way to zero?

**Bias-Variance trade-off**

From the franke analysis (3.5):

Regression models must balance bias, error, and variance, or as Belkin et. al. put it: "[find] the ’sweet

spot’ between underfitting and overfitting" [6]. The point is to find a model that is complex enough

to provide good predictions, without also adding in patterns from random noise in the training data.

**Real data**

**Results:**

**Just add and describe the graphs, without much further comment or analysis:**

**MSE vs degree:**

Mse and R2 show similar behavior: MSE is almost 1 for the simplest polynomial, then improves to 0.8 for the 2nd order polynomial. The fit keeps improving for higher complexity, but only slightly.

**Lasso vs Ridge:**

The error gets worse for higher lambdas. For Lasso, the fit gets suddenly worse as coefficients trend towards zero, while the change is more gradual for the Ridge fit.

**Bias-variance trade-off**

Error and bias follow each other closely, while variance shows an extremely slow increase for higher complexity. There is a significant improvement in error from low complexity to high.

**Kfolds:**

Some spread between initial values for different k-folds. Ridge shows an initial improvement, before performing worse as lambda continues to increase. For lasso, the error gets sharply worse as lambda increases.

**Discussion:**

1. **Comment on mse and r2 results comment on scaling (ref this?** [**https://scikit-learn.org/stable/autoexamples/pre**](https://scikit-learn.org/stable/autoexamples/pre)

The scaling was added in order to get MSE values of a reasonable size. Because MSE depends on the size of the parameters, large parameters give large MSE even for errors that would normally be considered small when compared to the initial data value. For this reason, scaling by …. Was done, ensuring data values between … and … as seen in fig. …..

As seen in the fig- of MSE vs complexity, this scaling resulted in mses that are possible to interpret at values between 1 and 0.

Unlike what we saw for the Franke data, the errors keep improving for higher complexity for both test and training sets. There is also a similar behaviour in the R2-score %suggesting no great outliers???

Again, this was the same for the Franke data.

For low complexity, almost no correct features from the data are captured. There is an improvement at the 2nd order polynomial, but the fit is still not great at MSE = 0.8. Although the error improves for higher complexity, there is no equally sharp improvement (as the one seen from 1-2) suggesting that adding further complexity may not be the best solution.

1. **Ridge: analyse mse for different lambda compare with results from a**
2. **Lasso: Same as for b Compare the three methods and discuss which is the better fit**

Due to limited data power, degree = 3 was chosen for comparing lasso and ridge errors. As expected from the theory discussed in chpt. …. There is a sharp change in error for Lasso as certain coefficients drop suddenly to zero, while there is a more gradual change in the Ridge errors. Both models perform best for lambda close to zero. As coefficients are shortened %?

To a greater degree the error becomes noticebly worse. This suggests that the features dropped were important to the model, and should be included. Again, similar behaviour is seen for the R2-score.  
As for the Franke data, this suggests OLS performs better than both Ridge and Lasso for the terrain data.

**e) Discuss bias-variance trade off for the data set (as fcn of complexity) discuss maybe the**

**training set split wrt bootstrap resampling method**

The error seen here, however, is far better than what was seen from the simple split into test and training sets used in the comparison between ols, ridge and lasso above.

This analysis was not run for very high polynomial degrees due to lack of computation power, but it is expected that the values continue to drop slightly before again increasing into the u-shape described by … .

An important caveat here, however, is that the terrain data was simplified somewhat before running the bias-variance analysis. It is possible the data became less complex and a lower degree polynomial was now enough to capture the important features of the data.

**f) k-folds cross validation resampling Evaluate mse compare mse to bootstrap**

**main question: Evaluate which model fits the data best - critical evaluation of the results -**

**applicability to the type of data discussed here (assumes a continuous function - terrain surface is**

**continuous)**

OLS shows a significant drop for all k-folds after the 2nd order polynomial.

Ridge and Lasso now show different behaviours, were the Ridge folds show a drop in errors at lambda = 1, while lasso performs worse for larger lambdas, as was seen for the simple data split too.

Even though Ridge now shows an improvement for a certain lambda, the error is still worse than from OLS. This suggests OLS is still the method that best described the data,

TODO: Ensure titles are short enough so they don’t overlap on top of document

TODO: Try to run the beta coefficient graph again? Try coefficient vs lambda

**TODO: Consider moving some of the discussion to the results bit as descriptions of the results**

**TODO: Consider running through the first mse ridge etc plots again with the same data set used later to get a proper comparison.**

**Conclusion:**

Ordinary Least Squares was the best method to describe the terrain data, seen by the fact that ridge and lasso behaved best for the smaller lambda values. The results from the bias-variance and k-folds validation resamplings suggest that the initial split into test and training sets may have resulted in an uneven split between significant data points. However, this result is somewhat uncertain as the terrain data was simplified in order to perform these validations.

If I were to repeat this experiment, I would pre-process the data better, ensuring that the same data set could be run through all steps and validation processes. Just running through the ridge vs lasso vs ols comparison with the simplified data set would also provide useful insight here.