Exercises week 39
FYS-STK3155 - Project 1

Ådne Rolstad
University of Oslo
(Dated: September 24, 2025)

We have studied various regression models, including ordinary least squares (OLS), Ridge regression and LASSO regression. Regression models fit polynomials to given target data. Such models are important, as they provide an intuitive and practical understanding of some of the fundamental ideas in machine learning. We wanted to study how the mean squared error (MSE) of the models depend on the model complexity and the model parameters. To understand this, we made our own implementation of OLS, Ridge and LASSO, as well as implementing a gradient descent method. In order to study the Bias-Variance trade-off, we included the resampling methods bootstrapping and cross-validation. We fitted regression models to Runge's function, and showed that OLS gives the best model. We also show that a polynomial of degree 8 gives the smallest MSE in OLS. While studying the Bias-Variance tradeoff, we show that the bias decreases when the model complexity increases, while the variance increases. This implies that the more complex models are in an area of overfitting.

I. INTRODUCTION

Linear regression models are used to approximate a continuous relationship between inputs and a target variable. Such models are simple and offers interpretable coefficients that explain how the inputs affect the output. In some cases, linear regression models can out-perform more complicated non-linear models. Different linear regression models, such as ordinary least squares (OLS), Ridge and LASSO, can yield different parameters due to regularization, so it is necessary to compare the performance of the different models. We evaluate performance with the mean squared error (MSE) and the R^2 score [1].

Many linear regression models, such as OLS and Ridge, have analytical expression, but models such as LASSO does not. For such cases, it is necessary to use numerical methods such as gradient descent to compute the model parameters. There are different gradient descent methods, either with a fixed or updated learning rate [2].

To obtain a more reliable picture of model quality, we use resampling methods such as Bootstrap and Cross-Validation. By repeatedly fitting models on resampled data, we can estimate variability, examine the bias-variance trade-off, and choose a model complexity) that gives the best fit while avoiding overfitting [2].

First, data are generated from Runge's function. The data is scaled and split into training and test data. We implement a standard OLS algorithm and perform a standard OLS regression analysis using polynomials of degree up to 15. We fit the model on Runge's function and evaluate the MSE and R^2 as functions of the model complexity.

We then implement a Ridge algorithm and perform a Ridge regression analysis on the same function. We analyse the MSE and R^2 and how these values depend on the regularization parameter λ. A fixed learning rate gradient descent method is then implemented. We perform the same OLS and Ridge analysis before comparing the results with the closed-form results.

Next, we implement gradient descent methods with updated learning rates. These methods include momentum, ADAgrad, RMSprop and ADAM. We compare the results from the different methods. We implement a LASSO regression model and perform the same analysis as before. The results are then compared with the results from OLS and Ridge.

A stochastic gradient descent method is then implemented, and the results are compared with the results from the other gradient descent methods. We include LASSO and Ridge regression in this analysis. The last step is the Bias-Variance analysis. We implement a bootstrap method and perform an OLS analysis using the bootstrap resampling technique. The variance, bias and MSE are plotted, and we discuss the Bias-Variance trade-off.

Next, we implement a k-fold cross validation method and perform the same analysis using this resampling technique. The results are compared with the bootstrap results.

Finally, we perform the bias-variance analysis on Ridge and LASSO, using both bootstrap and Cross-Validation, and compare the results with the OLS results.

In section II we describe the methods and algorithms used in the report. In section III we present and discuss the results from the various analyses. Section IV contains a conclusion and summary of the results and discussion.

II. METHODS

A. Method 1/X

- Describe the methods and algorithms, including the motivation for using them and their applicability to the problem
- Derive central equations when appropriate, the text is the most important part, not the equations.

B. Implementation

- Explain how you implemented the methods and also say something about the structure of your algorithm and present very central parts of your code, not more than 10 lines
- You should plug in some calculations to demonstrate your code, such as selected runs used to validate and verify your results. A reader needs to understand that your code reproduces selected benchmarks and reproduces previous results, either numerical and/or well-known closed form expressions.

C. Use of AI tools

- Describe how AI tools like ChatGPT were used in the production of the code and report.

III. RESULTS AND DISCUSSION

As shown in figure 1, the bias decreases while the variance increases as the model complexity increases.

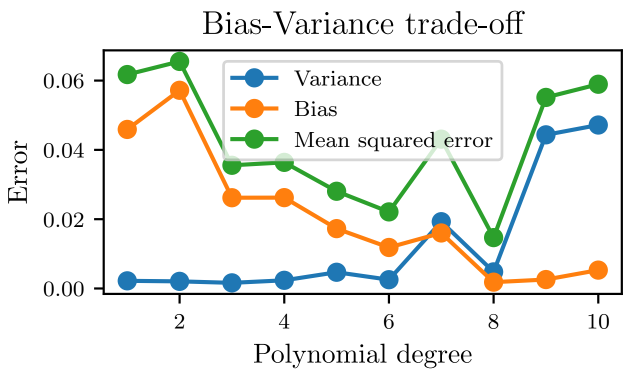

Figure 1: A plot showing the bias-variance trade-off

Figure 2 shows how the MSE changes with different values of lambda and different polynomial degrees in Ridge regression.

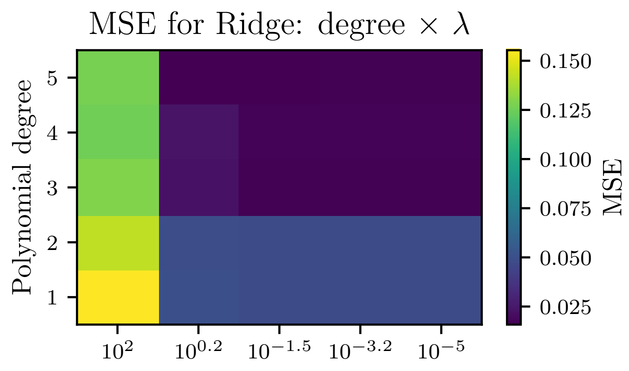

Figure 2: Ridge regression MSE for different lambda values and polynomial degrees

IV. CONCLUSION

- State your main findings and interpretations
- Try to discuss the pros and cons of the methods and possible improvements
- State limitations of the study
- Try as far as possible to present perspectives for future work

[1] T. Hastie, R.Tibshirani, and J.Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics* (Springer, New York, 2009), URL `https://link.springer.com/book/10.1007%2F978-0-387-84858-7`.
[2] M. Hjorth-Jensen, *Computational Physics Lecture Notes 2015* (Department of Physics, University of Oslo, Norway, 2015), URL `https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf`.