Report Template
FYS-STK3155/4155 - Project 1

Viktor Bilstad, Frederik Callin Østern, Heine Elias Husdal
University of Oslo
(Dated: September 26, 2025)

The abstract gives the reader a quick overview of what has been done and the most important results. Try to be to the point and state your main findings. It could be structured as follows:

- Short introduction to topic and why its important
- Introduce a challenge or unresolved issue with the topic (that you will try to solve)
- What have you done to solve this
- Main Results
- The implications

In this project we have been investigating some fundamental regression methods consisting of Ordinary Least Squares (OLS), Ridge, and LASSO using custom code. We have applied these methods to approximate Runge's function for input values between -1 and 1, with the intention to gain insight to their advantages, differences and nature in general. The performance of the models is evaluated with Mean Square Error (MSE) and R^2. We have also applied numerical ways of finding optimal parameters θ such as gradient decent with extensions such as stochastic gradient decent, ADAgrad and RMSprop. In the end we touch upon splitting the OLS cost function into a bias and a variance part, and resampling techniques. These methods and techniques are among the main building blocks in machine learning (ML). They also provide the fundamental principles to understand how to model data, optimize parameters, and evaluate performance. This helps us address challenges like overfitting, higher dimensionality and noisy data. All of this is an essential step to further tackle more advanced techniques such as neural networks.

One of the group members read through the first draft to the introduction a bit late. There were ideas for improvement, but we did not have time to coordinate the revisions. Thus, we have two drafts at the moment. Also, note that where we have written (KILDE), we intend to add citations later.

I. INTRODUCTION, DRAFT 1

Regression analysis is a central supervised learning technique in machine learning and statistics, which enables us to make a qualified numerical prediction based on a set of input features. These predictions can be done in a wide range of fields such as economics, weather forecasting and medical diagnostics (KILDER). For instance, when predicting the amount a person spends in a month, the features can consist of age, location of home, number of children, income and so on. Some of these will have a greater impact on the total spendage than others and a good regression model captures the underlying patterns illustrating this. We can train our model on a dataset containing these features, thus predicting the spending of a unknown person x given the features. However, traditional regression often encounter challenges like overfitting, where a too complex model capture noise rather than the more general trend. In the opposite case, a too simple model to accurately capture the patterns in the data is underfitting. How can we optimize our model and their respective parameters to avoid these problems?

To investigate the fundamentals driving these kinds of problems, we have is useful to focus on the regression model itself applied to approximate a simple 1D function while varying the polynomial degree. In this project we have chosen to approximate Runge's function.

II. INTRODUCTION, DRAFT 2

Regression analysis is a central type of supervised learning in machine learning and statistics, which enables us to make a qualified numerical prediction based on a set of input features. These predictions can be done in a wide range of fields such as economics, weather forecasting and medical diagnostics (KILDE). In general, given a labeled training dataset of inputs and outputs, one attempts to construct a model that best fits the output values for a given input. There are several regression methods, most notably Ordinary Least Squares (OLS), Ridge and Lasso regression. These can be used to minimize the mean squared error between the predictions and true output, up to some extra penalty in the Ridge and Lasso cost functions.

In order to find an optimal model, one must choose an appropriate model complexity and *train* the model on a test data subset. If the complexity is too low, the model may be too simple to accurately reflect the complexity of the data and hence to predict the training data output. However, if the complexity is too large, the model may *overfit* on the training data, making it unable to make accurate predictions on unseen data. Bias-variance tradeoff and k-fold cross validation are two resampling-based techniques that allows one to find robust model parameters that provides an appropriate balance between the two extremes of model complexity.

To train the chosen model, one must use an appropriate regression technique, as well as a method for improving model parameters. Gradient descent (GD) methods are a central class of methods in machine learning to iteratively improve model parameters. There are several different different algorithms, including ordinary gradient descent, gradient descent with momentum, Adagrad, RMSprop and ADAM, as well as a stochastic gradient descent version of each of these techniques. In the last decade, the ADAM technique has experienced a particularly high popularity among researchers (KILDE).

In this work, we have practised using the various regression techniques, gradient descent algoirthms and resampling techniques on a simple one-dimensional function. In particular, we considered data taken from the one-dimensional Runge function, with continuous input values in an interval between -1 and 1. The function was approximated using polynomial fits. We used the MSE and R^2 statistic to quantify the error between the optimal predictions and true output for different polynomial degrees and numbers of datapoints. Furthermore, for the OLS, Ridge and Lasso regression techniques, we analyzed the evolution of the mean squared error as a function of the number of iterations of various gradient descent algorithms, comparing how quickly they converged to the analytical solution. Finally, we used the bias-variance tradeoff and k-fold cross validation techniques to analyze possible optimal model complexity and regression techniques. All in all, the goal of the work was to gain insight and intuition into how the various machine learning techniques worked on simple models, so that in the future, one would be better equipped to tackle more complicated problems, including the use of neural networks.

In section 2, we present the theory behind the machine learning techniques and explain how we analyzed them in our work. In section 3, we present the results of our analysis on the Runge function. Finally, in section 4, we draw conclusions from our work and discuss possibilities for future work.

When you write the introduction you should focus on the following aspects:

- Motivate the reader, the first part of the introduction gives always a motivation and tries to give the overarching ideas. Citing some central ideas or problems in the literature is a good idea here. [1][2][3, 4]
- What you have done, with a focus on choice of problem and method, and why these were chosen.
- The structure of the report, how it is organized. List the sections, and very briefly describe what is in them and how they fit together.

III. METHODS

The regression techniques were analyzed in two ways. First, we implemented and tested various gradient descent algorithms in order to investigate the convergence of these methods when training polynomial fits to the Runge function. Then we used resampling techniques in order to find the optimal model complexity, which is given by the degree of the polynomial fit. Together, these results were used to discuss which of the three regression methods were best to fit the Runge function.

A. Regression techniques

Consider a dataset consisting of a vector $\boldsymbol{x}$ with n input data values and a corresponding vector $\boldsymbol{y}$ of output data. Suppose the data can be modelled by the relation

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

for some non-stochastic continuous function $\boldsymbol{f}$. Here, $\boldsymbol{\epsilon}$ is a vector of independent and identically normally distributed error terms with zero mean and some standard deviation $\sigma > 0$. In regression, we use the dataset to fit a model to the output values [5]. For linear regression, in particular, the model approximation $\tilde{\boldsymbol{y}}$ of the output values is given by a linear function of the input,

$$\tilde{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\theta}$$

Here, $\boldsymbol{X}$ is the *design matrix* corresponding to the input data [5]. The columns in this matrix is some function of the input values in $\boldsymbol{x}$. For a polynomial fit (which we focused on in our work), the j-th column in $\boldsymbol{X}$ consists of the j-th powers of the input values in $\boldsymbol{x}$. In other words, $X_{i,j} = {x_i}^j$.

The vector $\boldsymbol{\theta}$ consists of a certain number p of parameters. These parameters must be trained in order to minimize the error between the predicted and actual output values. Different regression techniques quantify the error using different cost functions.

1. *OLS regression*

In the Ordinary Least Squares (OLS) regression method, the cost function of the model parameters that must be minimized is the usual mean squared error (MSE) given by

$$C(\boldsymbol{\theta}) = \frac{1}{n}(\boldsymbol{y} - \tilde{\boldsymbol{y}})^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^2$$

There is an analytical expression for the optimal parameters where $C(\boldsymbol{\theta})$ has its minimum. It is given by

$$\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

2. OLS regression

B. 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.

C. 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.

D. Use of AI tools

-We asked ChatGPT how to use the \ref command in a LaTeX document. This helped us to refer to figures in the report.

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

IV. RESULTS AND DISCUSSION

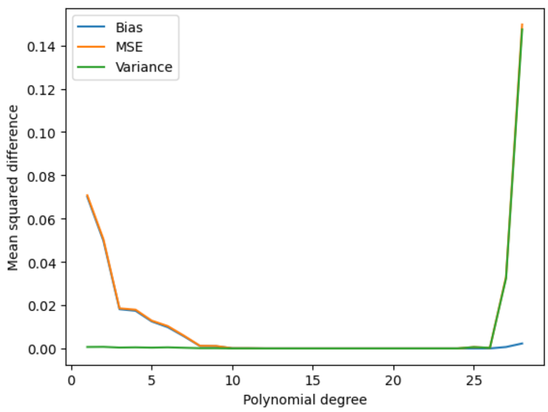

Figure 1: Mean square error (MSE), bias, and variance plotted as a function of polynomial degree. In the middle plateau (degree 10-25) MSE, bias, and variance is low, indicating a range of useful polynomial degrees to fit the data.

Figure 1 shows the bias, variance and MSE as a function of the degree of the polynomial fit.

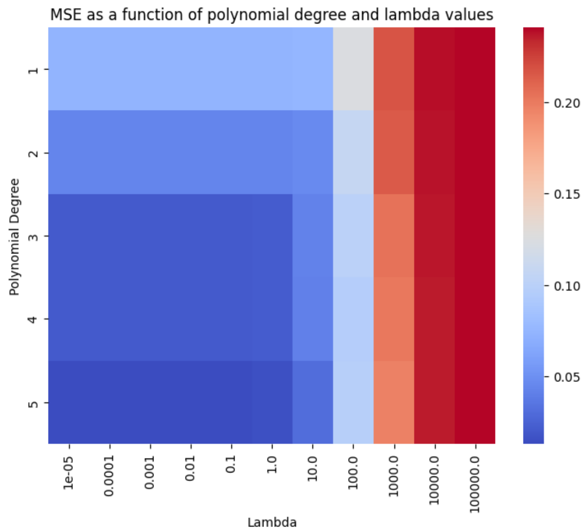

Figure 2: Heatmap showing optimal parameters for minimizing Mean Square Error (MSE).

Figure 2 shows a useful way to find good parameters when fitting a model with Rigde. [4] [6]

- Present your results
- Give a critical discussion of your work and place it in the correct context.
- Relate your work to other calculations/studies
- An eventual reader should be able to reproduce your calculations if she/he wants to do so. All input variables should be properly explained.
- Make sure that figures3 and tables contain enough information in their captions, axis labels etc. so that an eventual reader can gain a good impression of your work by studying figures and tables only.

V. 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

Figure 3: My dog.

[1] 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`.

[2] H. J. T. Zhang Yi, Yan Fu, Computers and Mathematics with Applications **47**, 1155 (2004), URL `https://doi.org/10.1016/S0898-1221(04)90110-1`.

[3] K. B. Hein, *Data Analysis and Machine Learning: Using Neural networks to solve ODEs and PDEs* (Department of Informatics, University of Oslo, Norway, 2018), URL `https://compphysics.github.io/MachineLearning/doc/pub/odenn/html/._odenn-bs000.html`.

[4] 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`.

[5] M. Hjorth-Jensen, *Applied data analysis and machine learning* (2025), URL `https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/intro.html`.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Journal of Machine Learning Research **12**, 2825 (2011), URL `http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html`.