Applying Regression and Resampling Techniques on Franke Function and Real Terrain Data

Andreas Isene*, Christian D. Nguyen*, and Daniel Fremming*

The field of Machine Learning (ML) stands central when it comes to gathering, processing, evaluating and drawing conclusions from data. Scientists from this field often work with large data sets, which are then filtered to draw out meaningful information which can have various forms of applications, such as predicting the results of a political election or weather forecast. These prediction models, or more befitting, linear regressions, come with implications when generalizing to a data set. They can be difficult to use and tuning their parameters can be troublesome to tune. The model can either be too simple to describe the complexity or too specialized - either way, it fails as a means of predicting unseen data.
In this paper we introduce 3 regression methods for the reader, them being the ordinary least squares(OLS), Ridge- and Lasso-regression methods. Before applying the regression models on real world terrain data, we fine-tune them by trying to fit them to the 2D Franke's function, and once more by applying 2 resampling techniques, i.e. kFold cross-validation and bootstrapping. In the latter, we will take a look into the bias-variance trade-off which is key in fine-tuning our model. **Fifth: your results - what was your best model and with what MSE/R2. Sixth: implications: is the best model a good model, i.e. can it solve your problem? In other wordst your abstract is missing the start and the end...**

CONTENTS

1. INTRODUCTION

Never before have the world generated as much volume of data as of today. In the current stage of the Information Age data dictates the way we live as it can help us foresee and predict patterns, nonetheless, also enable us to see the bigger picture.

Regression analysis is a central tool in science as a whole. It can be utilized to make predictions based on multiple independent variables. As an example one could predict house prices based on location, size, number of rooms, etc. There are various different methods that has been developed, each with its unique implications. As such we have been motivated to explore specifically *ordinary least squares*, *ridge regression*, and *lasso regression*. These methods have been applied to fit models on real topographic data. They were then be evaluated and improved through resampling techniques.

What is the purpose of what we have done. Is it really just to "explore regression"? What about "investigate

* These authors contributed equally to this work
[1] For full excerpt of conversion with chatGPT, see github material

wether regression can be used to accuratly predict terrain data"? The first paragraph of should be a motivation, not too general. Second paragraph: what you have done (in more detail than the abstract). Third: Structure of the repoprt. References I liked your method of referencing chatGPT! The references are correct, but consider *where* you reference. For instance you might want to find a source that supports that regression analysis is a central tool (i.e. first sentence of the introduction) Looks like a good start! Keep on with the good work :)

This is some simple reference (1). This is another elaborate reference (2; 3) In conversation with GPT we established that it liked ice cream [1].

2. THEORY

2.1. Regression analysis

2.2. OLS

2.3. Ridge

2.4. Lasso

2.5. Bias-Variance trade-off

We define the cost function:

$$C(X,\beta) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y_i})^2 = \mathbb{E}[(y-\tilde{y})^2] \qquad (1)$$

The last equation describes the expected residual error on the data dataset:

$$\mathbb{E}[(y-\tilde{y})^2] = \mathbb{E}[(y-\tilde{X\beta})^2]$$

Which can be rewritten as a term consisting of the bias and variance term:

$$\mathbb{E}[(y-\tilde{y})^2] = Bias[\tilde{y}] + Var[\tilde{y}] + \sigma^2 \qquad (2)$$

The derivation of the term can be accessed in appendix A 7.7.1

The bias term represents the error caused by in-built assumptions in our used method. It consists of the average prediction over all data sets subtracted with the desired regression function. The variance term describes the difference between solutions for the data sets and their average(variance of the model arounds its mean) (4). ¡- fix this

As for the sigma term, it represents the variance of the error, i.e. the noise of our method, where this quantity is expected to be irreducible. Since the sigma term is irreducible we will omit its interpretation in this section. The equation of minimum squared error(MSE) can be regarded as the expected loss when using our regression model, where we desire to minimize the error. This minimization can be dubbed the bias-variance trade-off, since it is about striking a balance between the bias and variance term. As for interpreting the terms(bias and variance), we will look into 2 scenarios as to how model complexity affects their behaviour. Let $\tilde{y}$ denote our model. The more complex it is, the better it is to go through the data points, hence we will acquire a lower bias. Respectively, the variance will be higher because the a higher model complexity makes the model "move" through more data points and it learns the noise of the training data. This flexibility in the model will lead run into the problem of overfitting our model, it works well on the training data we input, but it will run into problems once given new data it has never "seen", i.e. overfitting makes the model less able to generalize.

If the model is too rigid, it will output a high bias and low variance, and the model is not sufficient to fit the data points - we say it will underfit. Similarly to the case with a complex model, it will result in a poor generalization.

1. SVD

2. Resampling techniques

2.6. Bootstrap

2.7. Kfold - cross validation

2.8. feature scaling

1. splitting data

3. METHODS

3.1. Franke's function

3.2. Implementation-workflow with code

- Describe the methods and algorithms
- You need to explain how you implemented the methods and also say something about the structure of your algorithm and present some parts of your code
- You should plug in some calculations to demonstrate your code, such as selected runs used to validate and verify your results. The latter is extremely important!!

4. RESULTS

4.1. Franke Fucntion

Declare no. samples and no. of polymer degree

1. OLS

   - MSE - plot
   - R^2 - plot
   - *beta*-analysis (function of the degree of polynomials - table?

2. Ridge

   - MSE - plot
   - R^2 - plot
   - *beta*-analysis (function of the degree of polynomials - table?
   - *lambda*-analysis

3. Lasso

   - MSE - plot
   - R^2 - plot
   - *beta*-analysis (function of the degree of polynomials - table?
   - *lambda*-analysis

4.2. Bias-Variance trade-off

Perform then a bias-variance analysis of the Franke function by studying the MSE value as function of the complexity of your model.

4.3. Adding complexity with resampling techniques - implementing the kfold cross-validation

evaluate again the MSE function resulting from the test folds.

4.4. Terrain Data

1. Ridge

2. Lasso

5. DISCUSSION

5.1. Regression Analysis of the Franke's Function

1. Discussion of Model Performance differences of methods

2. Differences in the three models

3. Which model fits the data best?

4. Bias-Variance Trade-off

5. MSE in Kfold Cross-validation vs. MSE in Bootstrap

Compare the MSE you get from your cross-validation code with the one you got from your bootstrap code. Comment your results. Try 5 10 folds. In addition to using the ordinary least squares method, you should include both Ridge and Lasso regression

Discuss the bias and variance trade-off as function of your model complexity (the degree of the polynomial) and the number of data points, and possibly also your training and test data using the bootstrap resampling method

5.2. Terrain Data and General Remarks

6. CONCLUSION AND FURHER WORK

7. APPENDIX

7.1. Analytical expression for the MSE

Assumptions:

1. ε is independent from x which gives: $\mathbb{E}[\varepsilon] = 0$)
2. $y = f(x) + \varepsilon$. Here we will simplify f=f(x).
3. $f(x)$ is a fixed, deterministic function of x, hence $\mathbb{E}[f(x)] = f(x)$

7.2. The mean-square error of our model can be written:

$$\begin{aligned}\text{MSE} &= \mathbb{E}[(y - \tilde{y})^2] \\ &= \mathbb{E}[y^2 - 2y\tilde{y} + \tilde{y}^2] \\ &= \mathbb{E}[y^2] - 2\mathbb{E}[y\tilde{y}] + \mathbb{E}[\tilde{y}^2]\end{aligned}$$

We look into each term:

$$\mathbb{E}[y^2] \tag{3}$$

$$\mathbb{E}[y\tilde{y}] \tag{4}$$

$$\mathbb{E}[\tilde{y}^2] \tag{5}$$

From (1) we have:

$$\begin{aligned}\mathbb{E}[y^2] &= \mathbb{E}[(f + \varepsilon)^2] = \mathbb{E}[f^2 + 2f\varepsilon + \epsilon^2] \\ &= \mathbb{E}[f^2] + \mathbb{E}[\varepsilon^2] \\ &= \mathbb{E}[f^2] + \sigma^2 = f^2 + \sigma^2\end{aligned}$$

From (2):

$$\begin{aligned}\mathbb{E}[y\tilde{y}] &= \mathbb{E}[y]\mathbb{E}[\tilde{y}] = \mathbb{E}[f + \varepsilon]\mathbb{E}[\tilde{y}] \\ &= \mathbb{E}[f]\mathbb{E}[\tilde{y}] = f\mathbb{E}[\tilde{y}]\end{aligned}$$

We have:

$$\begin{aligned}Var[\tilde{y}] &= \mathbb{E}[(\tilde{y}] - \mathbb{E}[\tilde{y}])^2] \\ &= \mathbb{E}[\tilde{y}^2 - 2\tilde{y}\mathbb{E}[\tilde{y}] + (\mathbb{E}[\tilde{y}])^2] \\ &= \mathbb{E}[\tilde{y}^2] - 2\mathbb{E}[\tilde{y}]\mathbb{E}[\tilde{y}] + (\mathbb{E}[\tilde{y}])^2 \\ &= \mathbb{E}[\tilde{y}^2] - (\mathbb{E}[\tilde{y}])^2 \\ &\implies \mathbb{E}[\tilde{y}^2] = Var[\tilde{y}] + (\mathbb{E}[\tilde{y}])^2\end{aligned}$$

Put the terms together:

$$\begin{aligned}&\mathbb{E}[y^2] - 2\mathbb{E}[y\tilde{y}] + \mathbb{E}[\tilde{y}^2] \\ &= f^2 + \sigma^2 - 2f\mathbb{E}[\tilde{y}] + (Var(\tilde{y}) + (\mathbb{E}[\tilde{y}])^2) \\ &= f^2 - 2f\mathbb{E}[\tilde{y}] + (\mathbb{E}[\tilde{y}])^2 + Var(\tilde{y}) + \sigma^2 \\ &\implies \mathbb{E}[(f - \mathbb{E}[\tilde{y}])^2] + Var(\tilde{y}) + \sigma^2\end{aligned}$$

First term in above expression can be approximated:

$$\mathbb{E}[(f - \mathbb{E}[\tilde{y}])^2] \simeq \frac{1}{n}\sum_i (y_i - \mathbb{E}[\tilde{y}])^2 \text{ ,where } f_i \simeq y_i \tag{6}$$

From (4) we have that MSE can be written:

$$\mathbb{E}[(f - \mathbb{E}[\tilde{y}])^2] \simeq \frac{1}{n}\sum_i (y_i - \mathbb{E}[\tilde{y}])^2 = Bias[\tilde{y}] \tag{7}$$

Similarly, the variance can be expressed as:

$$\mathbb{E}[(\tilde{y} - \mathbb{E}[\tilde{y}])^2] \simeq \frac{1}{n}\sum_i (\tilde{y}_i - \mathbb{E}[\tilde{y}])^2 = Var[\tilde{y}]$$

Setting in both the bias and variance terms into the equation we then obtain:

$MSE = Bias(\tilde{y}) + Var(\tilde{y}) + \sigma^2$

REFERENCES

[1] Yoshua Bengio Ian Goodfellow and Aaron Courville. *Deep Learning*. MIT Press, Massachusetts, 2016.

[2] Yuxi (Hayden) Liu Sebastian Raschka and Vahid Mirjalili. *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing, Birmingham, UK, 2022.

[3] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Statistical Learning — Data Mining, Inference, and Prediction*. Springer Science+Business Media, New York, NY, 2009.

[4] Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Statistical Learning — Data Mining, Inference, and Prediction*. Springer Science+Business Media, New York, NY, 2009.