FYS-STK3155 Project 1

Lasse Pladsen, Parham Qanbari, & Sander V. Vattøy

(Dated: September 29, 2023)

We have studied three different regression methods, ordinary least squares, ridge, and lasso, and what affects their learned prediction accuracy. We found ... . We have done a bias-variance analysis on OLS using a bootstrap resampling technique and found that ... . We then study the cross-validation resampling method and analyze all three regression methods ... . Finally we applied our models to real world topographic data and we saw that ... .

I. INTRODUCTION

In today's modern world machine learning has emerged as a revolutionary technology. Even if just superficially, machine learning has become well known around the world. By allowing computers to learn from supplied data we create powerful tools with incredible real world applications for analyzation and prediction.

In this project we will explore and study low-level machine learning methods. Our focus will be on understanding and optimizing the potential of data regression methods such as ordinary least squares (OLS), ridge regression, and lasso regression. We will look into the factors that influence their accuracy and predictive capabilities, employing a variety of strategies to minimize errors and improve performance. Finally, we will apply our finely-tuned algorithms to analyze real-world topographic data.

II. THEORY

A. Design matrix for linear regression

We create a so-called design matrix $\mathbf{X}$ for the regression methods from two input variables $\mathbf{x}$ and $\mathbf{y}$. We use $\mathbf{X}$ to create our linear regression predictions. Each row in $\mathbf{X}$ represents a polynomial from one data sample. We choose a max polynomial degree p with n data samples such that $\mathbf{X} \in \mathbb{R}^{n\times l}$, where $l = floor[\frac{1}{2}(p+1)(p+2)]$, and is given by

$$\mathbf{X} = \begin{bmatrix} 1 & x_0 & y_0 & x_0^2 & x_0 y_0 & y_0^2 & \cdots & x_0^{p-1} & \cdots & y_0^{p-1} \\ 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & \cdots & x_1^{p-1} & \cdots & y_1^{p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & y_{n-1} & x_{n-1}^2 & x_{n-1}y_{n-1} & y_{n-1}^2 & \cdots & x_{n-1}^{p-1} & \cdots & y_{n-1}^{p-1} \end{bmatrix} \tag{1}$$

We then make our linear regression predictions by creating the polynomials from $\mathbf{X}$ and the polynomial coefficients $\boldsymbol{\beta} \in \mathbb{R}^l$ as follows

$$\tilde{\mathbf{z}} = \boldsymbol{X\beta} \tag{2}$$

B. Regression methods

The different methods have different ways of calculating the optimal $\boldsymbol{\beta}$-coefficients which we call $\hat{\boldsymbol{\beta}}$.

1. Ordinary least squares

$$\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{z} \tag{3}$$

2. Ridge

$$\hat{\boldsymbol{\beta}}_{Ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{z} \tag{4}$$

where $\mathbf{I}$ is the identity matrix, and λ is a non-negative regularization parameter.

3. Lasso

For lasso regression we do not have an analytical expression, but we will use the functionalities of the machine learning python module `SciKit-Learn`.

4. Error and coefficient of determination

To calculate our regression models' prediction error from the actual data we will be using the mean square error (MSE) defined by

$$MSE(\mathbf{z}, \tilde{\mathbf{z}}_i) = \frac{1}{n}\sum_{i=0}^{n-1}(z_i - \tilde{z}_i)^2, \tag{5}$$

We calcilate the coefficient of determination for our model, called the R^2, given as

$$R^2(\mathbf{z}, \tilde{\mathbf{z}}_i) = 1 - \frac{\sum_{i=0}^{n-1}(z_i - \tilde{z}_i)^2}{\sum_{i=0}^{n-1}(z_i - \bar{z}_i)^2} \tag{6}$$

C. The Franke function

To study our regression methods we will be using the Franke function, which is a two dimentional weighted sum of four exponentials:

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

D. Bias-variance trade-off

We will study bias-variance trade-off by using **bootstrap** resampling technique.

III. METHODS

The first thing we are going to do is to creat our own code for the regression method OLS which is given by equation 2 and 3. To analyse the data we use equation 5 and 6 to find the mean square error and R2-score.

Next up we make our own Ridge regression method using equation 2 and 4, and do the same MSE and R2 analysis as above, but now for different values of the λ parameter.

Last regression method we want to analyse is the Lasso method. We do a similar analysis as for the previous regressions, but now use `Scikit-Learn`.

We'll now do a Bias-variance trade off, but only for the OLS method.

IV. RESULTS

test [1]

V. DISCUSSION

VI. CONCLUSION

Appendix A: Github repository

`https://github.com/LassePladsen/FYS-STK3155-projects/tree/main/project1`

Appendix B: List of source code

Here is a list of the code we have developed in this project (NB: står i prosjektoppgaven at vi må ha med dette: ”The report file should include all of your discussions and a list of the codes you have developed. Do not include library files which are available at the course homepage, unless you have made specific changes to them.”:

- ...
- ...
- ...

Appendix C: Analytical derivations

a. *Expectation value of* $\mathbf{y}$

We will show that $\mathbb{E}(y_i) = \mathbf{X}_{i,*}\boldsymbol{\beta}$ by using $\mathbf{y} = \mathbf{f} + \epsilon \simeq \mathbf{X}\beta + \epsilon$ and separating the expectation value of a sum. Here we approximated $\mathbf{f}$ with $\boldsymbol{X\beta}$ using OLS. Then taking a value of $\mathbf{y}$ with index i we get $y_i = \sum_j X_{ij}\beta_j + \epsilon_i$:

$$\begin{aligned} \mathbb{E}(y_i) &= \mathbb{E}(\Sigma_j X_{ij}\beta_j + \epsilon_i) \\ &= \mathbb{E}(\Sigma_j X_{ij}\beta_j) + \mathbb{E}(\epsilon_i) \\ &= \mathbb{E}(\Sigma_j X_{ij}\beta_j) \\ &= \Sigma_j X_{ij}\beta_j \\ &= \mathbf{X}_{i,*}\boldsymbol{\beta} \end{aligned}$$

b. *Variance of* $\mathbf{y}$

Using the same method as above we will now show $Var(y_i) = \sigma^2$ where σ^2 is the variance of our data's stochastic noise ϵ. Here we use the definition of variance being $Var(x) = \mathbb{E}(x^2) - \mathbb{E}(x)^2$:

$$\begin{aligned} Var(y_i) &= \mathbb{E}(y_i^2) - \mathbb{E}(y_i)^2 \\ &= \mathbb{E}[(\mathbf{X}_{\mathbf{i},*}\beta) + \epsilon_i)^2] - (\mathbf{X}_{\mathbf{i},*}\beta)^2 \\ &= \mathbb{E}[(\mathbf{X}_{\mathbf{i},*}\beta)^2 + \epsilon_i^2 + 2\mathbf{X}_{\mathbf{i},*}\beta\epsilon_i] - (\mathbf{X}_{\mathbf{i},*}\beta)^2 \\ &= \mathbb{E}[(\mathbf{X}_{\mathbf{i},*}\beta)^2] + \mathbb{E}[\epsilon_i^2] + \mathbb{E}[2\mathbf{X}_{\mathbf{i},*}\beta\epsilon_i] - (\mathbf{X}_{\mathbf{i},*}\beta)^2 \\ &= (\mathbf{X}_{\mathbf{i},*}\beta)^2 + \mathbb{E}[\epsilon_i^2] + 2\mathbf{X}_{\mathbf{i},*}\beta\mathbb{E}[\epsilon_i] - (\mathbf{X}_{\mathbf{i},*}\beta)^2 \\ &= \mathbb{E}[\epsilon_i^2] \\ &= Var(\epsilon_i) + \mathbb{E}(\epsilon)^2 \\ &= Var(\epsilon_i) \\ &= \sigma^2 \end{aligned}$$

c. *OLS expectation value of optimal* β

Here we will show that the expectation value for the optimal $\boldsymbol{\beta}$ for OLS, $\hat{\boldsymbol{\beta}}_{OLS}$, equals $\boldsymbol{\beta}_{OLS}$:

$$\begin{aligned}
\mathbb{E}(\hat{\boldsymbol{\beta}}_{OLS}) &= \mathbb{E}[(\mathbf{X^T X})^{-1}\mathbf{X^T y}] \\
&= (\mathbf{X^T X})^{-1}\mathbf{X^T}\mathbb{E}[\mathbf{y}] \\
&= (\mathbf{X^T X})^{-1}\mathbf{X^T}\boldsymbol{X}\boldsymbol{\beta}_{OLS} \\
&= \boldsymbol{\beta}_{OLS}
\end{aligned}$$

Here we used that the expectation value of the non-stochastic matrix is just the matrix itself ($\mathbb{E}(\mathbf{X}) = X$) since it has zero variance (non-stochastic).

d. Variance of optimal $\boldsymbol{\beta}_{OLS}$

Here we will show that the variance of the optimal $\boldsymbol{\beta}$ for OLS, $Var(\hat{\boldsymbol{\beta}}_{OLS}) = \sigma^2(X^T X)^{-1}$:

$$\begin{aligned}
Var(\hat{\boldsymbol{\beta}}_{OLS}) &= \mathbb{E}[(\hat{\boldsymbol{\beta}}_{OLS})^2] - \mathbb{E}[\hat{\boldsymbol{\beta}}_{OLS}]^2 \\
&= \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y})^2] - \boldsymbol{\beta}^2 \\
&= \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y})(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y})^T] - \boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= \mathbb{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}] - \boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbb{E}[\mathbf{y}\mathbf{y}^T]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} - \boldsymbol{\beta}^T\boldsymbol{\beta}
\end{aligned}$$

Here we will calculate $\mathbb{E}[\mathbf{y}\mathbf{y}^T]$ separately for ease:

$$\begin{aligned}
\mathbb{E}[\mathbf{y}\mathbf{y}^T] &= \mathbb{E}[(\mathbf{X}\boldsymbol{\beta} \boldsymbol{+} \boldsymbol{\epsilon})(\mathbf{X}\beta + \boldsymbol{\epsilon})^T] \\
&= \mathbb{E}[(\mathbf{X}\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}^T + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + \mathbf{X}\boldsymbol{\beta}\boldsymbol{\epsilon}^T + \boldsymbol{\epsilon}\boldsymbol{\beta}^T\mathbf{X}^T] \\
&= \mathbf{X}\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}^T + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] + \mathbf{X}\boldsymbol{\beta}\mathbb{E}[\boldsymbol{\epsilon}^T] + \mathbb{E}[\boldsymbol{\epsilon}]\boldsymbol{\beta}^T\mathbf{X}^T \\
&= \mathbf{X}\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}^T + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] \\
&= \mathbf{X}\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}^T + \sigma^2\mathbf{I}
\end{aligned}$$

Now I put this back into the Variance expression:

$$\begin{aligned}
Var(\hat{\boldsymbol{\beta}}_{OLS}) &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}^T + \sigma^2\mathbf{I})\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} - \boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= [\boldsymbol{\beta}\boldsymbol{\beta}^T\mathbf{X}^T + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\sigma^2]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} - \boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= [\boldsymbol{\beta}\boldsymbol{\beta}^T + \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}] - \boldsymbol{\beta}^T\boldsymbol{\beta} \\
&= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}
\end{aligned}$$

[1] Hjorth-Jensen, M. (2023). Project 1 on machine learning. `https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/project1.html`. [Online; accessed 26-September-2023].