University of Oslo

FYS-STK3155

Project 1

Eirik eirik@ifi.uio.no
Felix felix@ifi.uio.no
Brage Wiseth bragewi@ifi.uio.no

September 29, 2023

https://github.com/bragewiseth/MachineLearningProjects

Contents

Abstract

In this project, we explored the use of different methods for solving linear regres- sion on topological data. We emplyed OLS, ridge and lasso methods for linear regression on simulated height data using *Franke's Function*, Where the goal was to fit polynomi als to minimize the mean square error. As well as real topological data with the same goal.

Keywords: Linear Regression, Fitting Polynomials, Scaling, Bias-Variance

1. Introduction

To get a smooth start on this project we begin to construct some fake topological data. This can be constructed with *Franke's Function*. This gives us a nice dataset to start linear regression. A natrual start would be to use ordinary least squares to try to fit a polynomial to some degree. This is done by constructing a design matrix X with the polynomial terms and then solving the linear regression equation

$$\beta = (X^T X)^{-1} X^T \mathbf{y} \tag{1}$$

where β is the coefficients of the polynomial and $\mathbf{y}$ is the data. This is a nice start, but we can do better. We can add a penalty term to the cost function, this is called ridge regression.

$$\beta = (X^T X + \lambda I)^{-1} X^T \mathbf{y} \tag{2}$$

where λ is a hyperparameter that we can tune to get a better fit. This is a nice start, but we can do better. We can add a penalty term to the cost function, this is called ridge regression.

$$\beta = (X^T X + \lambda I)^{-1} X^T \mathbf{y} \tag{3}$$

where λ is a hyperparameter that we can tune to get a better fit. This is a nice start, but we can do better. We can add a penalty term to the cost function, this is called ridge regression.

$$\beta = (X^T X + \lambda I)^{-1} X^T \mathbf{y} \tag{4}$$

where λ is a hyperparameter that we can tune to get a better fit. This is a nice start, but we can do better. We can add a penalty term to the cost function, this is called ridge regression.

1.1 Structure of the report

2. Discussion on Scaling

Table 1: Unscaled sample design matrix fitting one-dimensional polynomial of degree 5

1.	0.	0.	0.	0.	0.
1.	0.25	0.0625	0.01562	0.00391	0.00098
1.	0.5	0.25	0.125	0.0625	0.03125
1.	0.75	0.5625	0.42188	0.31641	0.2373
1.	1.	1.	1.	1.	1.

Table 2: Scaled sample design matrix fitting one-dimensional polynomial of degree 5

```
0.  -1.41421  -1.0171   -0.83189  -0.728    -0.66226
0.  -0.70711  -0.84758  -0.7903   -0.71772  -0.65971
0.  0.        -0.33903  -0.49913  -0.56348  -0.58075
0.  0.70711   0.50855   0.29116   0.10488   -0.0433
0.  1.41421   1.69516   1.83016   1.90431   1.94603
```

the code for generating this output [1]

```
MSE for OLS on unscaled data:   0.010349396022903145
MSE for OLS on scaled data:     0.010349396024145656
MSE for Ridge on unscaled data: 0.02106077418650843
MSE for Ridge on scaled data:   0.01782525371566323
```

the code for generating this output [2]

First x_{unscaled} and x_{scaled} is not that different, the original data was close to zero-centered and not that spread out, which means that initially by just looking at the data scaling is not that necessary. When we add the polynomial terms we can now see that some of the entries of X_{unscaled} get really small as an example $0.1^5 = 0.00001$ this makes the columns of X_{unscaled} live in their own order of magnitude and scaling should be considered to bring them back to the same order of magnitude. The act of not scaling results in β spanning from -50 to 48 while scaling gives a smaller span from -10 to 13. Now this alone may not justify why we should scale this data set, as scaled and unscaled OLS yields the same MSE. However when doing ridge regression the cost function is directly dependent on the magnitude of β_i. Now with each β_i varying alot, some are getting more penalized than others. As we can see the MSE for unscaled data is much higher than for scaled data in the ridge case. This leads us to conclude that we should scale the data, making it easier to tweak λ and giving us nicer numbers to work with.

2.1 Analysis

3. Ordinary Least Squares

As we increase the order of the polynomial fit we see a corresponding decrease in MSE and increase in R2.

4. Ridge

5. Lasso

6. Paper & Pencil

The assumption we have made is that there exists a continuous function f (x) and a normal distributed error N (0, 2) which describes our data y = f (x) + We then approximate this function f (x) with our model ˜y from the solution of the linear regression equations (ordinary least squares OLS), that is our function f is approximated by ˜y where we minimized (y − ˜y)2, with ˜y = X . The matrix X is the so-called design or feature matrix. yi N (Xi,∗ , 2), that is y follows a normal distribution with mean value X and variance

2. We can use this when we define a so-called confidence interval for the parameters . A given parameter j is given by the diagonal matrix element of the above matrix. **??**

7. Bias & Variance

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

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

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

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

$$\begin{aligned}\mathrm{Bias}[\tilde{y}] &= \mathbb{E}\left[(\mathbf{y} - \mathbb{E}\left[\tilde{\mathbf{y}}\right])^2\right] = \mathbb{E}[\mathbf{y}^2] - 2\mathbb{E}[\mathbf{y}]\mathbb{E}[\tilde{\mathbf{y}}] + \mathbb{E}[\tilde{\mathbf{y}}^2] \\ &= \mathbf{f}^2 + \sigma^2 - 2\mathbf{f}\mathbb{E}[\tilde{\mathbf{y}}] + \mathbb{E}[\tilde{\mathbf{y}}]^2 \\ &= (\mathbf{f} - \mathbb{E}[\tilde{\mathbf{y}}])^2 + \sigma^2\end{aligned}$$

To find $\mathbb{E}[\tilde{\mathbf{y}}]$ we can use bootstrap to get different samples for $\tilde{\mathbf{y}}$ and then take the mean of that distribution. The term *variance* refers to how spread out our observations are. We can for our case think of the observations $\tilde{y}_i$ as either individual predictions, or the mean of predictions from one test set.
In a sense the bias resembles the mathematical defninition for variance, We can see that the

bias term is a distance measure of $\mathbf{y}$ from the expected value of $\tilde{\mathbf{y}}$. If we take the individual observations approach this can be tough of as A high bias will then result in a large band around $\tilde{\mathbf{y}}$ in which y_i will exist. The variance term is a measure of how much the $\tilde{\mathbf{y}}$ varies around its own mean. In other terms a high variance model will experience large variance in the predicted values $\tilde{\mathbf{y}}$ from one test set to another. Or as a result of this a large variance in score if you will. Now these two terms are in a sense opposites, it is likely that a model with high bias will have low variance and vice versa. This is something we as designers of the model get to tune, if we value a lower bias more than we value a low variance or vice versa, we can choose a model accordingly

Appendix

SVD

svd

$$MSE(\mathbf{y},\tilde{\mathbf{y}}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i-\tilde{y}_i)^2,$$

$$R^2(\mathbf{y},\tilde{\mathbf{y}}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i-\tilde{y}_i)^2}{\sum_{i=0}^{n-1}(y_i-\bar{y})^2},$$

$$\bar{y} = \frac{1}{n}\sum_{i=0}^{n-1} y_i.$$

adding these two we get

$$\frac{\partial\left[\frac{1}{n}||\mathbf{y}-\mathbf{X}\beta||_2^2+\lambda||\beta||_2^2\right]}{\partial\beta} = 0 = \frac{2}{n}\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)\beta - \frac{2}{n}\mathbf{X}^T\mathbf{y} \implies \beta = \left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

The minimization of ridge addition alone is

$$\min_{\beta\in\mathbb{R}^p}\frac{1}{n}\lambda||\beta||_2^2 = \frac{2}{n}\lambda\beta$$

We know that

$$\min_{\beta\in\mathbb{R}^p}\frac{1}{n}||\mathbf{y}-\mathbf{X}\beta||_2^2 = \frac{2}{n}\mathbf{X}^T\mathbf{X}\beta - \frac{2}{n}\mathbf{X}^T\mathbf{y}$$

$$\min_{\beta\in\mathbb{R}^p}\frac{1}{n}\sum_{i=0}^{n-1}(y_i-\tilde{y}_i)^2 = \frac{1}{n}||\mathbf{y}-\mathbf{X}\beta||_2^2,$$

$$\min_{\beta\in\mathbb{R}^p}\frac{1}{n}||\mathbf{y}-\mathbf{X}\beta||_2^2+\lambda||\beta||_2^2$$

$$\hat{\beta}_{\text{OLS}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y},$$

$$\hat{\beta}_{\text{Ridge}} = \left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y},$$

Use the singular value decomposition of an $n\times p$ matrix $\mathbf{X}$ (our design matrix)

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices of dimensions $n\times n$ and $p\times p$, respectively, and $\mathbf{\Sigma}$ is an $n\times p$ matrix which contains the ingular values only. This material was discussed during the lectures of week 35.

Show that you can write the OLS solutions in terms of the eigenvectors (the columns) of the orthogonal matrix $\mathbf{U}$ as

$$\tilde{\mathbf{y}}_{\text{OLS}} = \mathbf{X}\beta = \sum_{j=0}^{p-1} \mathbf{u}_j \mathbf{u}_j^T \mathbf{y}.$$

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$$

$$\tilde{\mathbf{y}}_{\text{OLS}} = \mathbf{X}\beta = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\left(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\right)^{-1}\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}^{T^{-1}}\mathbf{\Sigma}^{2^{-1}}\mathbf{V}^{-1}\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{y}$$

using the orthogonality of $\mathbf{V}$ and $\mathbf{U}$ we get. Multiplying with $\mathbf{\Sigma}$ removes columns from $\mathbf{U}$ with eigenvalues equal to zero.

$$\tilde{\mathbf{y}}_{\text{OLS}} = \mathbf{U}\mathbf{U}^T\mathbf{y} = \sum_{j=0}^{p-1} \mathbf{u}_j \mathbf{u}_j^T \mathbf{y}$$

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon$$

$$\tilde{\mathbf{y}} = \mathbf{X}\beta.$$

$$\mathbb{E}(y_i) = \sum_j x_{ij}\beta_j = \mathbf{X}_{i,*}\,\beta$$

$$Var(y_i) = \sigma^2$$
$$Var(\hat{\beta}) = \sigma^2\,(\mathbf{X}^T\mathbf{X})^{-1}.$$

Math Behind the Confidence Interval section 6

We can assume that y follows some function f with some noise ϵ

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

$$\mathbb{E}[\mathbf{y}] = \mathbb{E}[f(\mathbf{x}) + \epsilon] = \mathbb{E}[f(\mathbf{x})] + \mathbb{E}[\epsilon_i]$$

The expected value of ϵ_i is 0, $f(x)$ is a non-stochastic variable and is approimated by $\mathbf{X}\beta$

$$\mathbb{E}[y_i] = \mathbf{X}_{\mathbf{i},*}\beta$$

The variance is defined as

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

for the expected value of β we can insert the definition of β from earlier

$$\mathbb{E}[\hat{\beta}] = \mathbb{E}[(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbb{E}[\mathbf{Y}] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\beta = \beta$$

$$\begin{aligned}
Var(\hat{\beta}) &= \mathbb{E}\{[\beta - \mathbb{E}(\beta)][\beta - \mathbb{E}(\beta)]^T\} \\
&= \mathbb{E}\{[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} - \beta]\,[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} - \beta]^T\} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\,\mathbb{E}\{\mathbf{y}\,\mathbf{y}^T\}\,\mathbf{X}\,(\mathbf{X}^T\mathbf{X})^{-1} - \beta\,\beta^T \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\,\{\mathbf{X}\,\beta\,\beta^T\,\mathbf{X}^T + \sigma^2\}\,\mathbf{X}\,(\mathbf{X}^T\mathbf{X})^{-1} - \beta\,\beta^T \\
&= \beta\,\beta^T + \sigma^2\,(\mathbf{X}^T\mathbf{X})^{-1} - \beta\,\beta^T \quad = \quad \sigma^2\,(\mathbf{X}^T\mathbf{X})^{-1},
\end{aligned}$$

Bibliography

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Book in preparation for MIT Press.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[4] Wessel N. van Wieringen. Lecture notes on ridge regression, 2023.

[5] Brage Wiseth, Eirik ?, and Felix ? MachineLearningProjects. `https://github.com/bragewiseth/MachineLearningProjects`, September 2023.