University of Oslo

FYS-STK3155

Project 1

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

October 4, 2023

https://github.com/bragewiseth/MachineLearningProjects

Contents

Abstract

In this paper, we delve into the realm of machine learning model optimization and evaluation. Our study encompasses various regression techniques, including Ordinary Least Squares (OLS), Ridge, and Lasso regression, to analyze their effectiveness in handling simple and more complex datasets. Additionally, we employ bootstrap resampling and cross-validation methodologies to rigorously assess model performance and enhance generalization. A significant portion of our investigation is dedicated to understanding the delicate balance between bias and variance. We explore how regularization methods like Ridge and Lasso impact bias-variance trade-offs, offering insights into the stability and predictive power of these models. Furthermore, we provide empirical evidence of the benefits of cross-validation and bootstrap techniques in mitigating overfitting and improving model robustness. We found that { ..results.. }. Additionally we verify and compare our findings with well established theory and libraris such as SKLearn.

Keywords: Linear Regression, Scaling, Bias & Variance

1. Introduction

Machine learning has emerged as a powerful tool in data analysis, providing the ability to uncover complex patterns and relationships in diverse datasets. But, at its core, machine learning is all about finding functions that capture the underlying structure of the data. The use of machine learning algorithms to approximate functions is the essence of this paper.

Our motivation for this research lies in the exploration of machine learning techniques to approximate the terrain on our planet, which can perhaps be described by such a function. Earth's terrain exhibits peaks and troughs, hills and valleys, much like some polynomial functions. Fortunately, we can employ standard linear regression techniques to approximate polynomials, but the terrain presents its own set of challenges. Firstly, the terrain's true underlying function may not be a polynomial at all, and its complexity may vary significantly from one location to another. Secondly, our landscape is teeming with small, intricate details. Some regions are characterized by flat and smooth surfaces, while others are marked by rough and uneven terrain. Focusing too much on these minute details can lead to model overfitting, making it crucial to strike a careful balance between model complexity and generalization. In this context, regularization and resampling techniques, including Ridge and Lasso regression with bootstrap and cross validation, have proven indispensable. By introducing regularization and resampling, we aim to find the sweet spot between bias and Variance. And getting the best predictions we can with our assumptions.
To embark on this exploration, we will begin with a simpler case: "Franke's function." which mimics our real terrain data. This function serves as a foundational starting point, allowing us to assess our model's performance in a controlled environment before venturing into the complexity of real-world terrain data. Through this gradual progression, we provide ourselves wih a framework that can be applied to more complex and varied real-world terrain datasets.

Data: We begin by introducing the dataset used for our analysis, highlighting data collection and preprocessing procedures. Understanding the characteristics of the terrain data is fundamental to our modeling endeavor.

Methods and Scaling: Next, we delve into the methodology, encompassing the implementation of polynomial regression models and the application of regularization techniques such as Ridge and Lasso. Additionally, we will discuss the importance of proper scaling for model stability and convergence.
Bias-Variance Trade-off: A significant portion of our study will revolve around the critical concept of bias and variance. We'll explore how regularization methods influence this trade-off and delve into the fine balance between model complexity and generalization.
Results: In this section, we will present the outcomes of our experiments, showcasing the performance of different models and regularization techniques. Through empirical evidence, we aim to provide insights into the effectiveness of our approach.
Conclusion: Finally, we will summarize the key findings and their implications for terrain modeling with machine learning. Our conclusion will underscore the importance of regularization in achieving accurate representations of complex terrains and provide a perspective on future research directions.

code for generating all figures and data can be found at /MACHINELEARNINGPROJECTS/PROJECT1/SRC

2. Data

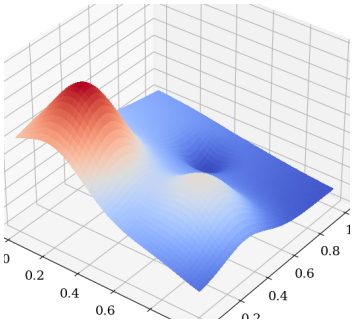
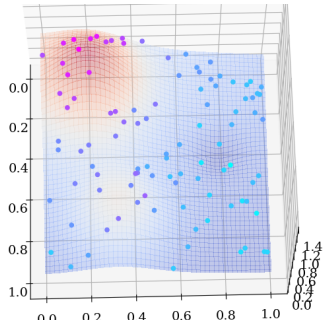


Figure 1: franke's function with noise

3. linear regression models

Figure 2: franke's function

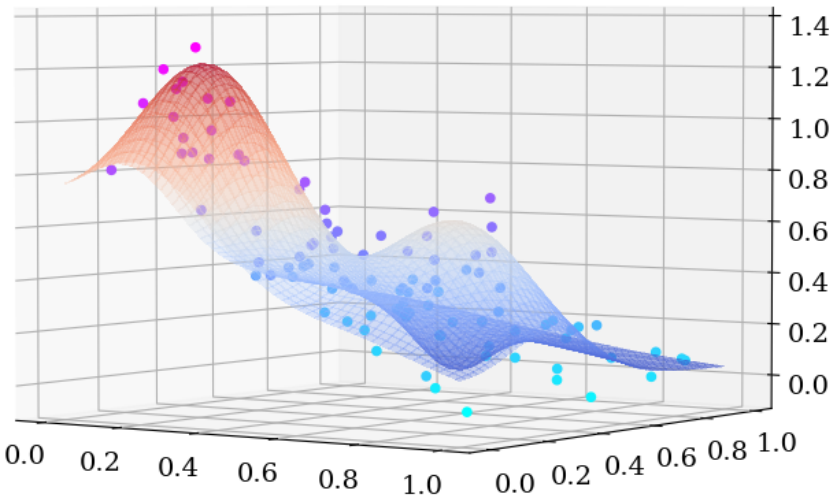


$$mse(\mathbf{y},\tilde{\mathbf{y}}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i-\tilde{y}_i)^2,$$

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

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

Data



3.1 Ordinary Least Squares (OLS)

3.2 Ridge Regression

3.3 lasso Regression

3.4 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.

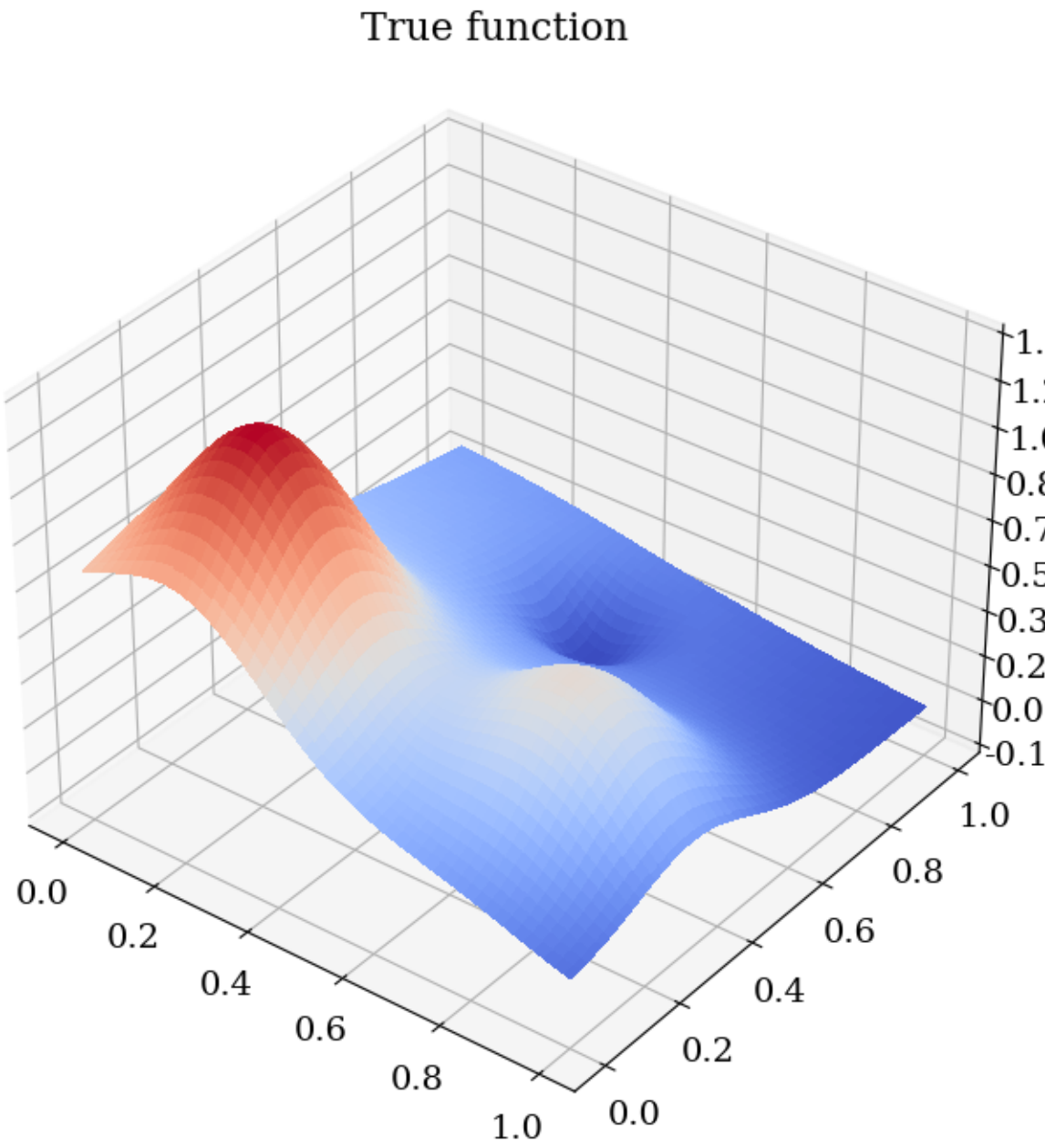
Table 2: scaled sample design matrix fitting one-dimensional polynomial of degree 5

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

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

At first glance, there isn't a big difference between the unscaled data (xunscaled) and the scaled data (xscaled). This is because the original data was already close to having a mean close to zero and not being spread out too much. So, you might think that scaling the data isn't necessary. However, when we introduce polynomial terms, we notice that some values in xunscaled become extremely small, for example, 0.15 turns into 0.00001. This means that the columns of xunscaled have vastly different orders of magnitude, and this calls for scaling to bring them back to a similar scale. If we don't scale, the coefficients ( ) can range widely, from -50 to 48. But when we scale the data, this range narrows down to -10 to 13. Now, this alone might not seem like a strong reason to scale the data, especially when we use plain linear regression (OLS), as it doesn't change the Mean Squared Error (MSE) much.

However, things change when we use Ridge regression. In Ridge, the regularization cost depends on the magnitude of i. With unscaled data, where i varies a lot, some coefficients get penalized more than others. This results in a significantly higher MSE for unscaled data compared to scaled data in Ridge regression. So, to make it easier to tune the regularization parameter ( ) and to ensure fair treatment of coefficients in Ridge regression, it's a good


True function
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

idea to scale the data. This not only simplifies the regularization process but also leads to better predictive performance.

4. Results and Discussion

Note to self på real data tester vi ikke alle hyperparametere, det tar alt for lang tid vi bruker den beste k fra synthetic data of gjør binærsøk på lambda og degrees

we see that ols remains constant since it is not a function of . for the mse from the training set we se that the mse wil increase for larger values of for rigde regression. this is expected since the ols wil give the best approximation for the data it has been trained on (compared to rigde), while rigde punishes the weights from becomming too large and therefore gives a not so tight fit. this can explain why rigde does better than ols for some values of . as stated before the ols wil try to fit the model to the best of its ability to the training data, this includes the noise which might not generalize to the test set, hence giving rigde a greater score. the term for this is called ***overfitting***. and balancing between overfitting and underfitting is called the ***bias variance tradeoff***.

4.1 confidence interval

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. **??**

5. bias & variance

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

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

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

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

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

to find $[\tilde{\mathbf{y}}]$ 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

appendix

Derivation of The Optimal Parameters β

OLS

The expression for the standard Mean Squared Error (MSE) which we used to define our cost function and the equations for the ordinary least squares (OLS) method, was given by the optimization problem

$$\min_{\beta\in\mathbb{R}^p}\frac{1}{n}\left\{(\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\beta)\right\}.$$

can also be written as

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

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

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

Ridge

We know that

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

The minimization of ridge addition alone is

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

adding these two we get

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

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

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

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

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

SVD

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

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

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

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

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.

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

Math Behind the Confidence Interval section

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

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

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

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

$$Var(y_i) = \sigma^2$$

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

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

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

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

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

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

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

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

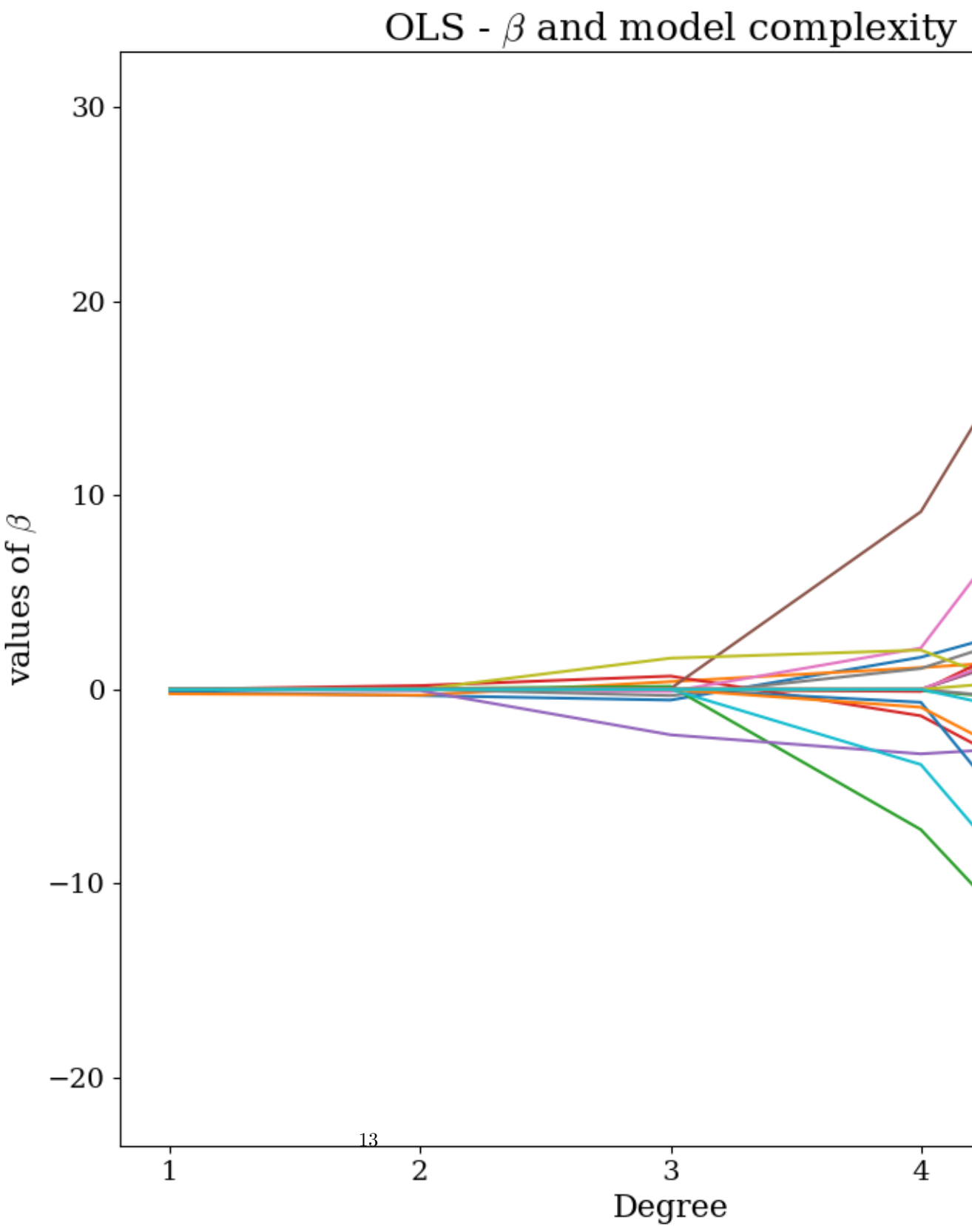
ridge

$$\mathbb{E}\big[\hat{\beta}^{\text{Ridge}}\big] = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top\mathbf{X})\beta.$$

$$Var[\hat{\beta}^{\text{Ridge}}] = \sigma^2[\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}]^{-1}\mathbf{X}^T\mathbf{X}\{[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}]^{-1}\}^T,$$

$$\mathbb{E}[\hat{\beta}^{\text{Ridge}}] = \mathbb{E}\left[\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\mathbf{X}^T\mathbf{y}\right] = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\mathbf{X}^T\mathbb{E}[\mathbf{Y}] = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\beta$$

$$\begin{aligned}
Var(\hat{\beta}^{\text{Ridge}}) &= \mathbb{E}\{[\beta - \mathbb{E}(\beta)][\beta - \mathbb{E}(\beta)]^T\} \\
&= \mathbb{E}\left\{\left[\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\mathbf{X}^T\mathbf{y} - \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\beta\right]\left[\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\mathbf{X}^T\mathbf{y} - \left(\mathbf{X}^T\mathbf{X} + \right.\right.\right. \\
&= \mathbb{E}\left\{\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\mathbf{X}^T\,\mathbf{y}\,\mathbf{y}^T\,\mathbf{X}\,\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1} - \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\beta\,\beta^T\left(\mathbf{X}^T\mathbf{X}\right)\left(\mathbf{X}\right.\right. \\
&= \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\mathbf{X}^T\,\mathbb{E}\{\mathbf{y}\,\mathbf{y}^T\}\,\mathbf{X}\,\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1} - \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\beta\,\beta^T\left(\mathbf{X}^T\mathbf{X}\right)\left(\mathbf{X}\right. \\
&= \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\mathbf{X}^T\,\left(\mathbf{X}\,\beta\,\beta^T\,\mathbf{X}^T + \sigma^2\mathbf{I}\right)\,\mathbf{X}\,\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1} - \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{pp}\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\beta\,\beta^T
\end{aligned}$$


OLS - β and model complexity
values of β
30
20
10
0
−10
−20
1
2
3
4
Degree

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.