Linear Regression and Resampling Methods

Project 1 FYS-STK4155

Grzegorz Dariusz Kajda

October 11, 2022

Abstract

In the following project we examine the methods of Ordinary Least Squares regression, Ridge regression and Lasso regression, and their application to terrain data. The study begins with the employment of OLS to polynomials of x and y upto the fifth order, and fitting of said polynomials to Franke function. We proceed by applying resampling techniques of bootstrap and kFold cross-validation to evaluate our model of the OLS, and repeat the process for Ridge and Lasso regression afterwards. By applying bootstrap to all three models, we are able to visualize the bias-variance tradeoff, and thus study the mean squared error as a function of bias and variance. By compairing the results of each model, we find that Ridge regression gives rise to the best fit, with OLS coming in second, while Lasso struggles to converge, making it difficult to gauge its performance.

Testing the models using topograhic data yeilded better results for Lasso regression, however the computational expenses associated its coordinate gradient are significant. Thus Ridge regression once again proves to produce the best fit, with the OLS falling slightly behind.

Contents

1 Introduction

The idea of predicting the future or the unknown may seem like a fool's errand to most, escpecially in a world ever-changning at a rate never seen before. While it is true that predicting the unseen is a nontrivial task, it is certainly possible. Many of the observations we make each day display a linear relationship, or allow us to make assumptions about such relationships. With this in mind, we can prove that for task which are not too complex, we can use the technique of regression analysis.

Hence, we are going study three different variants of regression in this article, namely the Ordinary Least Squares regression, Ridge regression and Lasso regression. We will start by genereting a small dataset with an addition of stochastic noise to it, and use it with our models to fit polynomials upto n-th order to the Franke Function. While doing this, the models will be evalueated on the basis of the mean squared error and R2-score. Resampling techniques known as bootstrap and kFold cross-validation will also be applied in order to assess the performance of our models. When we have become familiar with the possible

optimization techniques for our algorithms, we will apply our models to realterrain data. For the sake of context, this article has been divided into four major parts, the first being the introduction. The section that follows will present relevant mathematical theory, while section 3 will present the results obtained during the practical part along with a discussion of the results. The article will end with a conclusion.

2 Theory

2.1 Linear models

As mentioned in the introduction, linear regression is based on the assumption of linear relationships between observations or variables. From the statistical point of view, this means that if we're given a dependent variable y, we can explain the variable through a set of k feauteres $x=(x_0,x_1,x_2,...x_{k-1})$. Hence, this can be mathematically stated as follows:

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

This linear relationship can be rewritten in terms of a linear model, by introducing a set of coefficients $\beta=(\beta_0,\beta_1,\beta_2,...,\beta_{p-1})$. This allows to transform the equation from above to the form:

$$\tilde{y}=\mathbf{x}^T\beta$$

Expanding this logic to a dataset of m response variables $y=(y_0,y_1,y_2,...,y_{m-1})$, we can rewrite this equation again, now by stacking all feature vectors on top each other to form a design matrix $\mathbf{X}$:

$$\boldsymbol{X}=\begin{bmatrix}1 & x_0^1 & x_0^2 & ... & ... & x_0^{k-1}\\ 1 & x_1^1 & x_1^2 & ... & ... & x_1^{k-1}\\ 1 & x_2^1 & x_2^2 & ... & ... & x_2^{k-1}\\ ... & ... & ... & ... & ... & ...\\ 1 & x_{m-1}^1 & x_{m-1}^2 & ... & ... & x_{m-1}^{k-1}\end{bmatrix}$$

We can now write the problem of linear regression in terms of the product of the design matrix and regression parameters β:

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

This equation can now be analytically solved for optimal values of the parameters β

Now, in a general linear model, we assume that we can parametrize our function terms of a polynomial of degree *n-1*:

$$y=y(x)\rightarrow y(x_i)=\tilde{y_i}+\epsilon_i=\Sigma_{j=0}^{n-1}\beta_j x_i^j+\epsilon_i$$

With a little linear algebra, this equation can be rewritten to a simplier form:

$$y=X\beta+\epsilon$$

2.2 Ordinary Least Squares

2.3 Ridge Regression

2.4 Lasso Regression

3 Appendix A - Analytical Solution of OLS

Now let us assume that there exists a continous function *(*f(x)) with a normally distributed error ϵ *N(0,σ^2)* which describes our data:

$$y = f(x) + \epsilon$$

Function f(x) has been approximated through our model $\tilde{y}$, where we minimized the *Residuals sum of squares* $(y - \tilde{y})^2$, where:

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

As we know, $\mathbf{X}$ is our design matrix containing all of the independent variables $\mathbf{x}$ used to approximate $\mathbf{y}$. We are now going to show that the expectation value of $\mathbf{y}$ for any given element i can be written in the following way:

$$\mathbb{E}[y] = \Sigma_j x_i \beta_j = X_{i,*}\beta$$

Let us start the proof with the element by rewriting the expactation value of $\mathbf{y}$:

$$\mathbb{E}[y] = (1/n) * \Sigma_j y_i = (1/n) * \Sigma_{i=0}(f(x_i) + \epsilon_i)$$

Now we see that in order to prove out that $\mathbb{E}[y]$ is equal to the product $X_{i,*}\beta$, we need to prove that the value of $\epsilon_i = 0$. We can easily do it by finding the first derivative of the cost functions MSE:

$$\frac{\partial C(\beta)}{\partial \beta} = 0$$

As you can see, we set the derivative equal to zero in order to find the optimal parameters that will minimize our error.

$$X^T(y - X\beta) = X^T y - X^T X = 0$$

Now if this matrix $X^T X$ is invertible, which it is only if X is orthonormal, then with little algebra, we have the following solution for the optimal parameters:

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

Now in the situation where $X^T X$ is invertible, the error which we try to minimize will be equal to zero:

$$\epsilon = y - \tilde{y} = y - X(X^T X)^{-1} X^T y = y - y = 0$$

If you pay attention however, we could've from the start assumed that the value of $\epsilon = 0$, and written the proof in the following way:

$$\mathbb{E}[y_i] = \mathbb{E}[X_{i,*} + beta] + \mathbb{E}[\epsilon_i]$$
$$= X_{i,*}beta + 0 = \mathbb{E}[y] = X_{i,*}beta$$

This is simply caused by $\mathbb{E}[\epsilon_i]$ being by definition equal to zero, as it can be interpreted as the mean value of the error. Since the the mean value of the distribution of ϵ is equal to zero, we can write $\epsilon_i = 0$. Now the next thing we are going to prove, is that the variance of y_i is equal to σ^2. From the lecture notes and *Pattern Recognition and Machine Learning by Christopher M. Bishop*, we know that the equation giving us variance, can be written in terms of a expectation value:

$$Var(y_i) = \mathbb{E}[y_i - \mathbb{E}[y_i]] = \mathbb{E}[y_i^2] - (\mathbb{E}[y_i])^2$$
$$= \mathbb{E}[(X_{i,*}\beta + \epsilon_i)^2] - (X_{i,*}\beta)^2$$
$$= \mathbb{E}[(X_{i,*}\beta)^2 + 2\epsilon_i X_{i,*}\beta + \epsilon_i^2] - (X_{i,*}\beta)^2 = (X_{i,*}\beta)^2 + 2\mathbb{E}$$