

MACHINE LEARNING
FYS-STK 4155

PROJECT 1
KARI ERIKSEN

Abstract

Here is my abstract going to be

Introduction

One of the simplest models within the field of machine learning and statistical learning is linear regression. Because of its simplicity and the fact that it forms the base of many other more advanced methods it is highly recommended to understand it.

In this project we are studying several linear regression methods, the ordinary least square, Ridge regression and last the Lasso regression. Our first approach to understand these methods is to test them on the Franke function 1, a two-dimensional function which is widely used on testing interpolation and fitting algorithms. We evaluate our models ability to predict the outcome(solution) using a resampling method called bootstrap and explore statistical errors such as bias, variance and Mean Square Error (MSE). All methods are explained in the sections below.

At last we will be using real data from a website that provides terrain data over the earth, <https://earthexplorer.usgs.gov/>, and test out method on these.

Theory

The Franke function is a two-dimensional function consisting of a sum of four exponentials, eq. 1. This will be the function which we will generate data from to feed our regression models. We define it for $x, y \in [0, 1]$.

It can be seen represented as a surface plot in figure 1.

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

Regression methods

Linear Regression

Linear regression is a method in statistics that predicts the response of one or several explanatory variables. It assumes a linear relationship between the dependent and independent variables. At its simplest form we could try to find the straight line between two points. This equation is fairly simple.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\epsilon} \quad (2)$$

Here \hat{y} is a dependent variable, the outcome, x is an independent variable, or the predictor, and $\hat{\beta}_0$ and $\hat{\beta}_1$ the intercept and slope respectively. ϵ is the error in our prediction. The solution for $\hat{\beta}_0$ and $\hat{\beta}_1$ in this problem is best found with least square and is also fairly

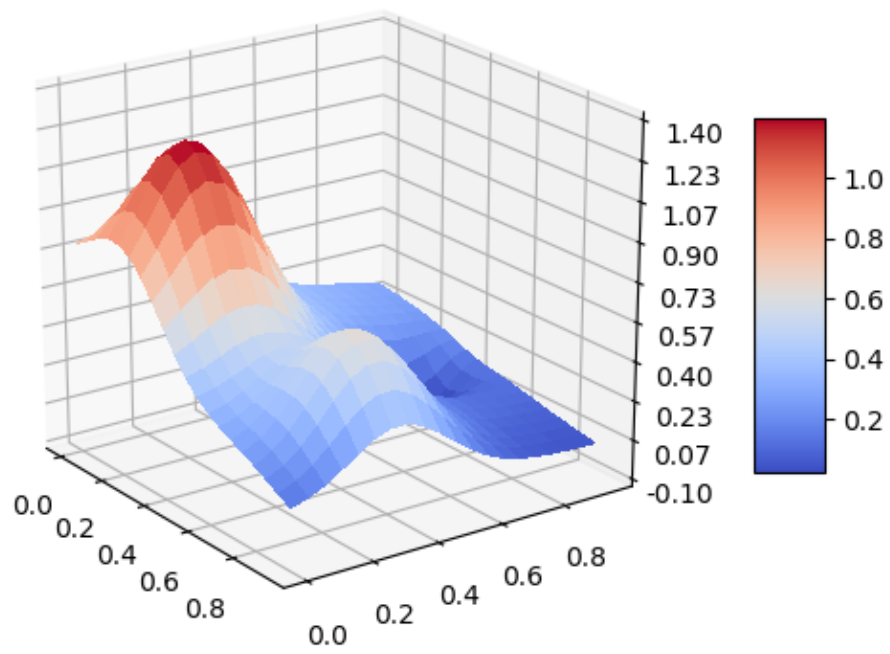


Figure 1: Surface plot of Franke function for $x, y \in [0, 1]$ generated with python code given in the assignment paper.

easy. Calculating the mean over both variables (\bar{x} and \bar{y}) we can find the parameters that give the prediction that differs the least from the exact solution.

$$\beta_1 = \frac{\sum^n (x_i - \bar{x})(y_i - \bar{y})}{\sum^n (x_i - \bar{x})^2} \quad (3)$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad (4)$$

If we have several predictors we can extend our problem to a more general case.

$$\hat{y} = f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j \quad (5)$$

Now X is a vector containing all predictors, $X^\top = \{X_0, X_1, X_2, X_3, \dots, X_p\}$, β_0 is the intercept and β_j is a vector keeping all coefficients for each predictor, the parameters we are searching for. \hat{y} is the predicted values of $y = f(X)$. Moving β_0 to the β -vector and adding an extra column with 1's to the design matrix X^\top we can reduce the problem to vector form and get the following. We will make use of this notation when finding solutions using least square etc.

$$\hat{y} = \hat{X}\hat{\beta} + \epsilon \quad (6)$$

In our case we want to search for a solution that is non-linear, as the Franke function is not linear. Instead we must look for a polynomial of higher degree. For the i 'th y -value the equation is given below.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \dots + \beta_d x_i^d \quad (7)$$

For this problem the design matrix will look like the following.

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_i & x_i^2 & \dots & x_i^d \end{bmatrix} \quad (8)$$

In addition our function depends on both x and y , so we have two variable that have polynomial features. These may also have interacting terms and therefore we must add interaction between the two variables. Thus our design matrix up to degree d looks like eq 9.

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & \dots & y_1^d \\ 1 & x_2 & y_2 & x_2^2 & x_2 y_2 & y_2^2 & \dots & y_2^d \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_i & y_i & x_i^2 & x_i y_i & y_i^2 & \dots & y_i^d \end{bmatrix} \quad (9)$$

This is the matrix we use to solve our problem at hand, and depending on the matrix having full column rank and the property $\mathbf{X}^\top \mathbf{X}$ being positive definite we will look at separate methods for finding solutions.

Ordinary Least Square

The least square method selects the parameters β so that residual sum of squares (RSS) is minimized.

$$\text{RSS}(\beta) = \sum_{i=1}^N (y_i - x_i^\top \beta)^2 \quad (10)$$

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) \quad (11)$$

$$\frac{\partial \text{RSS}(\beta)}{\partial \beta} = 0 = -2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\beta) \quad (12)$$

$$\mathbf{X}^\top \mathbf{y} - \mathbf{X}^\top \mathbf{X}\beta = 0 \quad (13)$$

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \quad (14)$$

Ridge Regression

$$\hat{y} = \hat{X}\hat{\beta} + \epsilon \quad (15)$$

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y} \quad (16)$$

Lasso Regression

Resampling method

Bootstrap

Evaluate some statistical estimates

Results

Hope I get some results...

Discussion

..to discuss.