FYS-STK4155 Mandatory Assignment 1: Regression analysis and resampling methods

Minh Chien Nguyen (davidngcz@gmail.com)

September 27, 2018

Abstract

The aim of this project is to compare three traditional approaches to regression analysis: Ordinary Least Squares method, Ridge regression and Lasso regression, using both simulated data and real data. In order to perform a proper assessment of the presented techniques, we employ a statistical method called k-fold cross-validation.

Introduction

Regression is the oldest, simple and widely used supervised machine learning algorithm for data analysis. This method is mostly used for forecasting and finding out relationship between variables. The different between regression techniques is based on the number of independent variables and the type of relationship between the independent and dependent variables. Here, we will restrict ourselves to the case of polynomial regression.

Three traditional methods of regression are recognized: Ordinary Least Squares method, Ridge regression and Lasso regression. All of them have received much attention among practitioners, as well as in academic literature.

This text is organized as follows. Main tools and results which allow for application to fitting functions are presented in Section 1. We further test performance of the presented methods on a Franke's function. Finally, the techniques are applied to real data in Section 2.2. In the last part of the text, we resume the principal results and suggest potential improvements in the future.

1 Mathematical tools

This section provides an overview of basic regression methods. Preliminary knowledge of statistic is expected. There are many excellent textbooks (see e. g. [1],[3] and [2]) which cover these topics. Here, however, the subtle mathematical issues are not addressed and we will present the main tools and results without details, but it will suffice for immediate application to fitting functions.

1.1 Linear regression

Linear regression is the most widely used statistical technique for predictive modeling, since it is very simple and often provides an good description of how the inputs affect the output.

Let $(x_1, y_1), \dots, (x_N, y_N)$ to be observed data, where each $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ is a vector obtained from the ith measurement. The set of variables $(x_1, \dots, x_N)$ is called the independent variable or the predictor variable while the set of variable $y = (y_1, \dots, y_N)$is called the dependent or the response variable.

The goal of the regression analysis is to extract/exploit relationship between y_i and x_i. The linear regression model assumes that there exists $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$ satisfying

$$y_i = f(x_i) = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j. \tag{1}$$

The method of least squares suggest that the coefficients β should minimize the residual sum of squares

$$\hat{\beta} = \text{argmin}_\beta (\sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2). \tag{2}$$

The unique solution to equation (2) is given by (see [2])

$$\hat{\beta} = \left(\hat{X}^T \hat{X}\right)^{-1} \hat{X}^T \hat{y}. \tag{3}$$

1.2 Regularization

One of the most common method to overcome overfitting is regularization. The main concept behind this approach is simplifying the model as much as possible. In other words, we reduce the magnitude of the coefficients of inputs in our model. In the following part, we present two different types of regression techniques using regularization.

Ridge Regression

The first techniques is called ridge regression and its coefficients are defined by

$$\hat{\beta}^{\text{ridge}} = \text{argmin}_{\beta}(\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2). \tag{4}$$

In equation (4) we added an extra term, which is known as the penalty term. $\lambda \geq 0$ is called tuning parameter. It easy to see that higher the value of α, higher is the term

$$\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

therefore the magnitude of coefficients β are reduced. Roughly speaking, we punish the cost function for high values of β. It is important to note that the parameter λ can be learned as well, using a method called cross validation that will be discussed in section 1.3. The ridge regression solution is given by

$$\hat{\beta} = \left(\hat{X}^T\hat{X} + \lambda I\right)^{-1} \hat{X}^T\hat{y}, \tag{5}$$

where I is the $p \times p$ identity matrix.

Lasso Regression

Lasso is similar to ridge, but with a small twist. The lasso estimate is in the form

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta}(\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|). \tag{6}$$

An important note to notice in equation (6) is that the only difference from ridge regression is term $\lambda \sum_{j=1}^{p} |\beta_j|$. Here, the method might include fewer predictors in the model, since lasso sets the irrelevant coefficient β to zero.

1.3 K-folds cross-variation

K-folds cross-variation is a very useful technique for calculating the performance of prediction models. The general procedure can be described as follows:

1. Shuffle the dataset randomly.
2. Divide the dataset into k groups.
3. For each group:
   (a) Take the group as a test data set.
   (b) Take the remaining groups as a training data set.
   (c) Fit a model on the training set and evaluate it on the test set.
   (d) Calculate the evaluation score.
4. Estimate the accuracy of the model by averaging the evaluation score derived in all the k cases.

For more information about regularization, we refer to [3].

1.4 Performance criteria

To test the performance of our results, we list the relevant criteria that we use throughout this section. In the next part, we will denote $\tilde{\hat{y}}_i$ the predicted value of the $i-th$ sample and y_i is the corresponding true value.

1. Mean squared error (MSE) - measures the average of the squares of the errors, which is defined as

$$MSE(\hat{y}, \hat{\tilde{y}}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2, \tag{7}$$

   it is clear that the smaller the value of MSE, the better the fit.

2. Coefficient of determination R^2 - is a number that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable. The R^2 is defined as

$$R^2(\hat{y}, \tilde{\hat{y}}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y})^2}, \tag{8}$$

   where the mean value of $\hat{y}$ is defined as $\bar{y} = \frac{1}{n}\sum_{i=0}^{n-1} y_i$.

3. Bias - is the difference between the expected prediction of our model and the correct value which we want to predict. Bias is defined as

$$Bias^2 = \sum_i (f(\boldsymbol{x}_i) - E_{\mathcal{L}}[\hat{g}_{\mathcal{L}}(\boldsymbol{x}_i)])^2, \tag{9}$$

   where $E_{\mathcal{L}}$ denotes the expected value of the functional over the dataset $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots\}$.

4. Variance - is an error from sensitivity to small fluctuation in the training set.

$$Var = \sum_i E[(\hat{g}(\boldsymbol{x}_i) - E[\hat{g}(\boldsymbol{x}_i)])^2], \tag{10}$$

   Models with high variance usually perform very well on training data but have high error on test data.

2 Testing on simulated and real data

After the theoretical considerations of the previous sections, we now want to test the performance of the presented techniques. First, we study how to fit polynomials to a Franke's function. Then we apply the methods on the polynomial approximation of digital terrain data.

2.1 Franke's function

In this section we perform a regression analysis of a specific two-dimensional function called Franke's function which is defined for $x, y \in [0, 1]$ as

$$f(x,y) = \frac{3}{4}\exp\left(-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right) + \frac{3}{4}\exp\left(-\frac{(9x+1)^2}{49} - \frac{(9y+1)}{10}\right) \tag{11}$$

$$+ \frac{1}{2}\exp\left(-\frac{(9x-7)^2}{4} - \frac{(9y-3)^2}{4}\right) - \frac{1}{5}\exp\left(-(9x-4)^2 - (9y-7)^2\right). \tag{12}$$

Figure 1 shows a three-dimensional plot of the Franke's function with an added stochastic noise to it using the normal distribution $N(0, 1)$. From equation (12) we know that the Franke's function is a weighted sum of four exponentials and it is clearly an infinitely differentiable function on

$[0,1] \times [0,1]$. Therefore, to perform an regression analysis of this function, we will test polynomial fits in x and y up to fifth order.

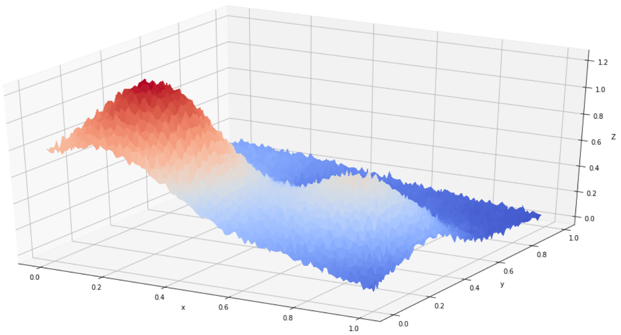

Figure 1: Average fitness of the best fit individual in each generation for 24 cities using Baldwinian GA

2.1.1 OLS

In this part we perform an OLS regression analysis of the Franke's function. First, we generate the arrays of values for x and y with a step size $h = 0.01$. Then, we evaluate the Franke's function adding stochastic noise $N(0,1)$. Next, we split our data set into a training set and a test set, where the training set contains 70% of the original data set. Finally, we fit a OLS polynomial regression model order 3 on the training set and evaluate it on the test set.
The estimation of coefficients β and their confident intervals are shown in Table 1. The value of MSE is 0.0080, meaning that the estimators of our model predicts observations with relatively high accuracy. The value of the coefficient of determination R^2 is 0.9015 saying that the regression predictions approximate the real data points very well.

β	Confident interval
0.959	(0.958, 0.961)
-0.380	(-0.387 ,-0.373)
1.441	(1.434, 1.448)
-1.807	(-1.820, -1.794)
-6.742	(-6.755, -6.729)
1.865	(1.855, 1.876)
1.121	(1.113, 1.129)
4.737	(4.729, 4.745)
0.461	(0.453, 0.468)
-1.445	(-1.452, -1.437)

Table 1: My caption

Now, we perform a resampling of the data for the OLS regression analysis least square regression analysis using polynomials in x and y up to fifth order. The process is set up as follows

1. Generate the arrays of values for x and y with a step size $h = 0.01$.
2. Evaluate the Franke's function with an added stochastic noise to it using the normal distribution $N(0, 1)$.
3. Perform resampling of the data using k-fold Cross-Validation
4. Fit the OLS regression model for each fold and each order of polynomials.

Table 2 shows the results obtained using classical OLS method with resampling. We see that as the order of the polynomial fit increases the value of the mean squared error (MSE) decreases, this means that using higher order polynomials is much more accurate.

	MSE	R2 score
3	0.0082	0.9
4	0.0044	0.94
5	0.0025	0.97

Table 2: My caption

Similarly, there is an increase in the value R-square as the order of polynomial increases. In the case of polynomial fit order 5, R^2 is 0.97, meaning, 97% of variance in z is explained by the predictors.

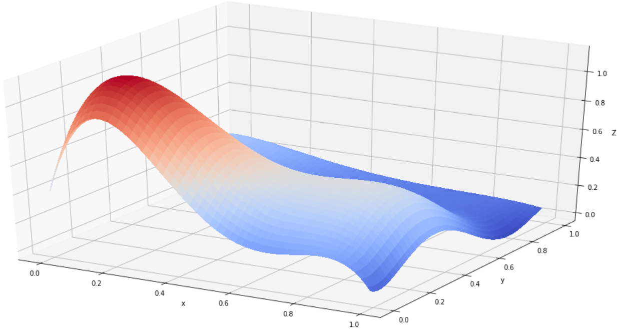

Figure 2: Average fitness of the best fit individual in each generation for 24 cities using Baldwinian GA

Whenever we discuss model prediction, it is important to understand prediction errors (bias and variance). Bias for the final model is: 0.00245 and Variance is: $5.48e-07$. We conclude that our model is underfit neither overfit, since the error of our model is relatively low.

2.1.2 Rigde regression on the Franke function with resampling

In this part, we perform the same analysis as in the previous section for the Ridge regression with different values of λ but now only with resampling. As mentioned above our model is not overfit, therefore it is not surprising that the best result is obtained using low value of the tuning parameter λ. In Tables 3, 4 and 5 we see that as the value of λ decreases the value of MSE for the Ridge regression model with resampling decreases, meanwhile the coefficient of determination R^2 is increasing.

λ	MSE	R^2 score
0.001	0.0082	0.90
0.01	0.0082	0.90
0.1	0.0082	0.90
1	0.0102	0.87
10	0.0161	0.80
100	0.0235	0.71

Table 3: order3

λ	MSE	R^2 score
0.001	0.0043	0.94
0.01	0.0047	0.94
0.1	0.0071	0.91
1	0.0091	0.88
10	0.0137	0.83
100	0.0223	0.73

Table 4: order 4

λ	MSE	R^2 score
0.001	0.0027	0.96
0.01	0.0037	0.95
0.1	0.0056	0.93
1	0.0089	0.89
10	0.0124	0.85
100	0.0213	0.74

Table 5: order 5

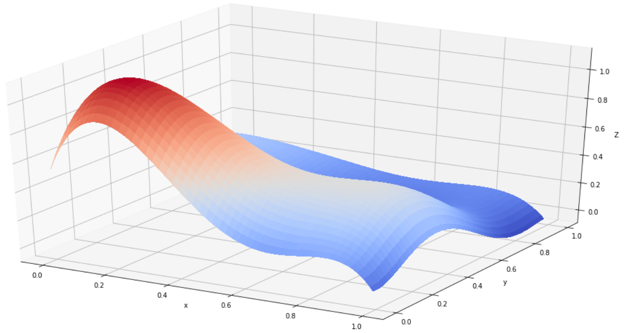

Figure 3: Average fitness of the best fit individual in each generation for 24 cities using Baldwinian GA

Again, in this case the obtained results of the model using polynomial order 5 is similar to the results using ordinary OLS model. Bias for the final Rigde model is: 0.00436 and Variance: $8.61e-07$

2.1.3 Lasso regression on the Franke function with resampling

This part is essentially a repeat of the previous two ones, but now with Lasso regression with resampling. Tables 6, 7 and 8 illustrate the results for Ridge regression using different values of α and orders of polynomials. It is worth mentioning that for some values of α the coefficient of determination R^2 is negative. The reason is that by increasing λ we reduce the magnitude of the coefficients. The higher the values of alpha is , the bigger is the penalty and therefore the magnitude of coefficients are reduced more. However, we also know that our OLS model is not overfit, which means that all predictors in our model are important and by reducing the magnitude of their coefficients we get worse predictions.

λ	MSE	R^2 score
0.001	0.0142	0.82
0.01	0.0254	0.69
0.1	0.0830	-0.0011
1	0.0830	-0.0011
10	0.0830	-0.0011
100	0.0830	-0.0011

Table 7: 4

λ	MSE	R^2 score
0.001	0.0135	0.83
0.01	0.0254	0.69
0.1	0.0830	-0.0011
1	0.0830	-0.0011
10	0.0830	-0.0011
100	0.0830	-0.0011

Table 8: 5

λ	MSE	R^2 score
0.001	0.0180	0.78
0.01	0.0254	0.69
0.1	0.0830	-0.0011
1	0.0830	-0.0011
10	0.0830	-0.0011
100	0.0830	-0.0011

Table 6: 3

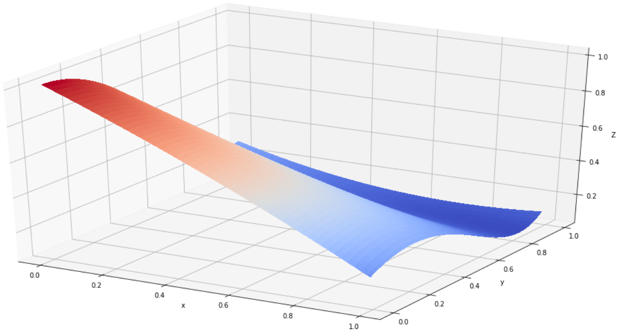

Figure 4: Average fitness of the best fit individual in each generation for 24 cities using Baldwinian GA

Bias for the best Lasso model is 0.01349 and Variance is $7.373e-07$. We can see that, the value for the bias of the Lasso model has been increased in comparison with the OLS and Ridge model. Therefore, in this case Lasso model is predicting worse than both OLS and Ridge.

2.2 Real data

To illustrate the techniques described in Section 1, we apply them to the digital terrain data from the website: https://earthexplorer.usgs.gov/.

2.2.1 OLS real data

	MSE	R2 score
3	305.03	0.71
4	219.98	0.78
5	170.80	0.82

Table 9: My caption

Avarage Bias for the final model is: 169.51 and avarage Variance is: 0.111.

2.2.2 Ridge real data

λ	MSE	R^2 score
0.001	304.89	0.71
0.01	304.89	0.71
0.1	304.89	0.71
1	305.09	0.71
10	321.40	0.66
100	760.59	-0.53

Table 10: order3

λ	MSE	R^2 score
0.001	219.86	0.78
0.01	219.86	0.78
0.1	219.86	0.78
1	220.27	0.78
10	248.02	0.69
100	599.39	-0.31

Table 11: order4

λ	MSE	R^2 score
0.001	170.65	0.82
0.01	170.65	0.82
0.1	170.65	0.82
1	171.38	0.81
10	209.01	0.70
100	443.66	0.01

Table 12: order5

Average bias for the best Rigde model is: 169.51 and average Variance: 0.102

2.2.3 Lasso real data

Conclusion

References

[1] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer.

[2] MARSLAND, Stephen. Machine learning: an algorithmic perspective. Chapman and Hall/CRC.

[3] Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning. Springer