Technical University of Denmark

Case 1

Course:
02582 Computational Data Analysis

Author(s):
Alessandro Montemurro (s171964)
Christoffer Hjort (s144224)
Daniel Thoren (s144222)

March 20, 2018

1 Data cleaning and pre-processing

Upon reading from the CSV file, the data is divided into input, X and output, y. The output consists of just 100 observations, which together with the first 100 input observations is used for selecting and training an appropriate model. Later the model predicts outputs for the remaining 1000 input observations.

One of the features within the input is categorical, i.e $X_{100} \in \{A, B, C\}$. To quantify the data, the categorical feature, X_{100} is replaced by a three-feature matrix $[X_{101}, X_{102}, X_{103}]$ - generated by one-hot encoding / one-out-of-k encoding, such that:

$$\begin{cases} \text{if } X_{100} \text{ is } A, & X_{101} = 1, \text{else } 0 \\ \text{if } X_{100} \text{ is } B, & X_{102} = 1, \text{else } 0 \\ \text{if } X_{100} \text{ is } C, & X_{103} = 1, \text{else } 0 \end{cases}$$

Due to the relatively small size of the data, any missing input features within an observation, are replaced with the column average. Before applying any regularization technique, the data is standardized - subtracting the mean of the columns and dividing all the variables by their standard deviation. This moves the variables to a zero-mean and makes them independent of their scale.

2 Model selection

Only linear models are considered for solving this problem, including, OLS, LARS, Ridge and ElasticNet. These are all linear models that each utilize different ways to regularize the weight parameters. Cross-validation is used to properly estimate the generalization error of these models. In this case 10-fold cross-validation is used, which results in a training set of size 90 and a test set of size 10.

For each model, cross-validation is used to determine the optimal regularization parameter by running 10-fold cross validation every time a parameter value is tested.

Figure 1 Shows the performance of OLS, which clearly fits the training data really well, since it has a training error of 0. However, OLS has a high test error of 1.5710 which indicates that the model is over-fitting and could use some regularization to increase its training error and lower its test error.

Table 1: OLS performance

Model	Training error	Test error
OLS	0.0000	1.5710

Figure 1 shows the training error and test error as a function of the number of non-zero coefficients. It is immediately clear by the test error, that the LARS model performs much better than OLS.

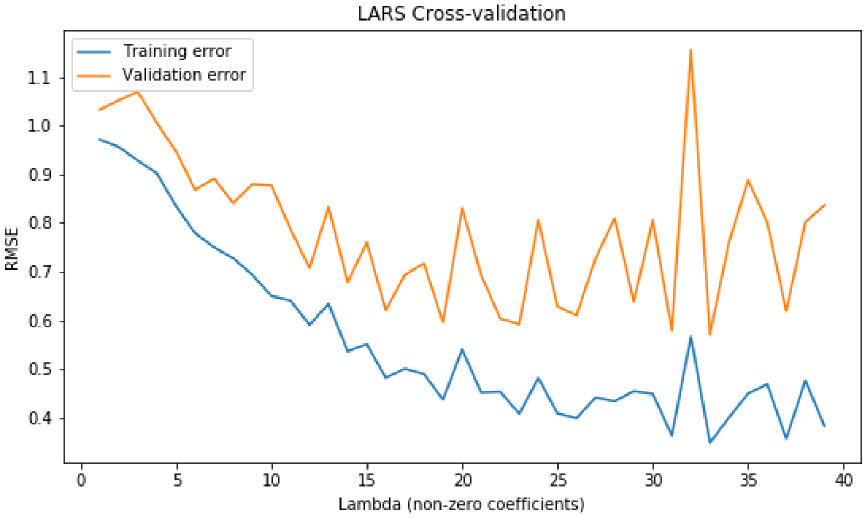

Figure 1: Cross-validation of the LARS model

Figure 2 shows the performance of the Ridge model as a function of the L1-norm. It shows that Ridge performs worse than LARS.

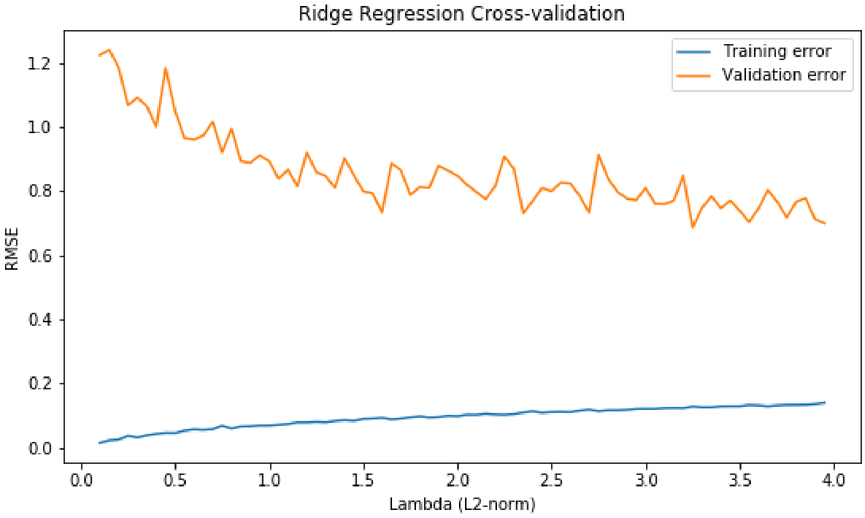

Figure 2: Cross-validation of the Ridge model

Finally figure 3 shows the performance of the ElasticNet model. Figure 3 is a scatter of models with different values of the L2-norm and L1-norm. The points are colored as a function of their RMSE, as indicated by the color bar.

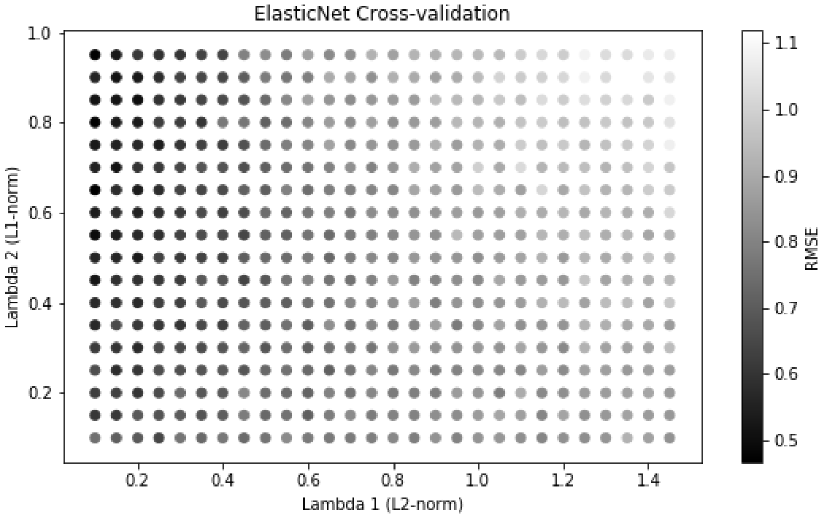

Figure 3: Cross-validation of the ElasticNet model

The top 5 best performing models are shown in table 2. These are the top 5 models of LARS, Ridge and ElasticNet, and it is clear that ElasticNet dominates with best performance. Because of this, we choose ElasticNet as our model of choice.

Table 2: Top 5 models by RMSE (sorted by test error)

Model	L1-norm	L2-norm	Training error	Test error
ElasticNet	0.1	0.65	0.2636	0.4665
ElasticNet	0.1	0.95	0.2822	0.4702
ElasticNet	0.1	0.8	0.2745	0.4726
ElasticNet	0.1	0.55	0.2552	0.4974
ElasticNet	0.2	0.85	0.3755	0.5040

The final model is chosen using the one-standard-deviation rule. The standard deviation of the test error was 0.1332 and therefore we choose the model that is closest to a test error of 0.5998 but slightly worse. The final model is shown in table 3.

Table 3: Final model with $+1\sigma$ from the best model

Model	L1-norm	L2-norm	Training error	Test error
ElasticNet	0.3	0.95	0.4671	0.6004

3 Results

The ElasticNet from the previous section forms a robust sparse estimate using the combination of L_2-norm shrinkage from Ridge and L_1-norm parameter selection from Lasso. By opening up the model in table 3, we can examine the features which the model deems most important - along with their relative weights. Figure 4 shows the model's predictions of the 1000 test samples. We expect this prediction to have an RMSE of around 0.6 based on the model selection process.

Table 4: Weight parameters of the final model

Variable	Weight
x_9	9.1615
x_{13}	-1.7919
x_{16}	1.4128
x_{22}	1.4906
x_{28}	4.1041
x_{38}	6.4429
x_{48}	1.4107
x_{67}	6.1567
x_{68}	3.9231
x_{70}	1.0401
x_{72}	2.4915
x_{75}	4.8922
x_{79}	-4.8195
x_{98}	5.9070
x_{101}	-5.2437
x_{102}	1.5149

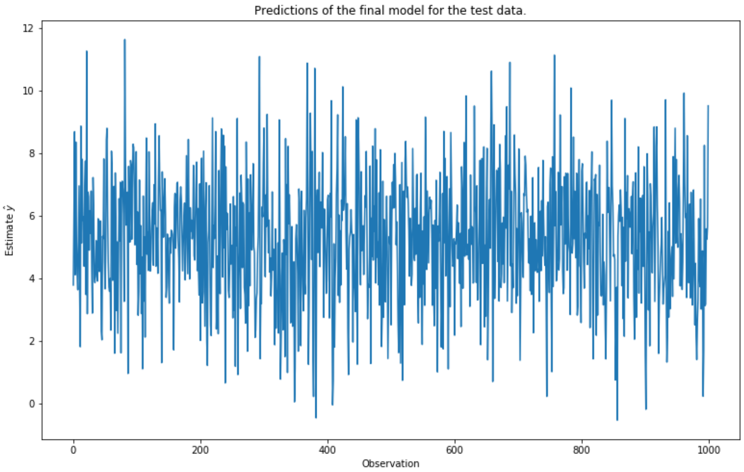

Figure 4: Predictions of the final model with 1 standard deviation from the best model.