

Compulsory exercise 2: Group XX

TMA4268 Statistical Learning V2018

NN1, NN2 and NN3

12 mars, 2018

1a)

- Q1: We have d possible predictors. First of all, we can choose how many of these predictors we want to use. We have the choice to use $k = 0, 1, 2, \dots, d$ predictors in our model. If we use $k = 0$ predictors, we don't really have a linear regression model, instead we will get a model that predicts new samples as the average of all previous samples. Even though this is not a linear regression model, we want to investigate whether this gives a better result than using a linear regression model would. That would imply that there is no linear connection between the data from our sample. Let us now focus on the choices of k that results in linear regression models. If we choose to use $k = 1$ predictor, we again have a choice in which of the d possible predictors we want to use. Obviously there are d possible choices for $k = 1$, but if we want to be more systematic we can say that we choose k different predictors from d possible ones, and we can not choose the same twice. Hence, the number of possible combinations using k predictors is given by the standard nCr-formula $\binom{d}{k}$. Now we have our d different choices of k , which all gives different linear regression models. In total, the number of models is

$$\binom{d}{1} + \binom{d}{2} + \dots + \binom{d}{d}.$$

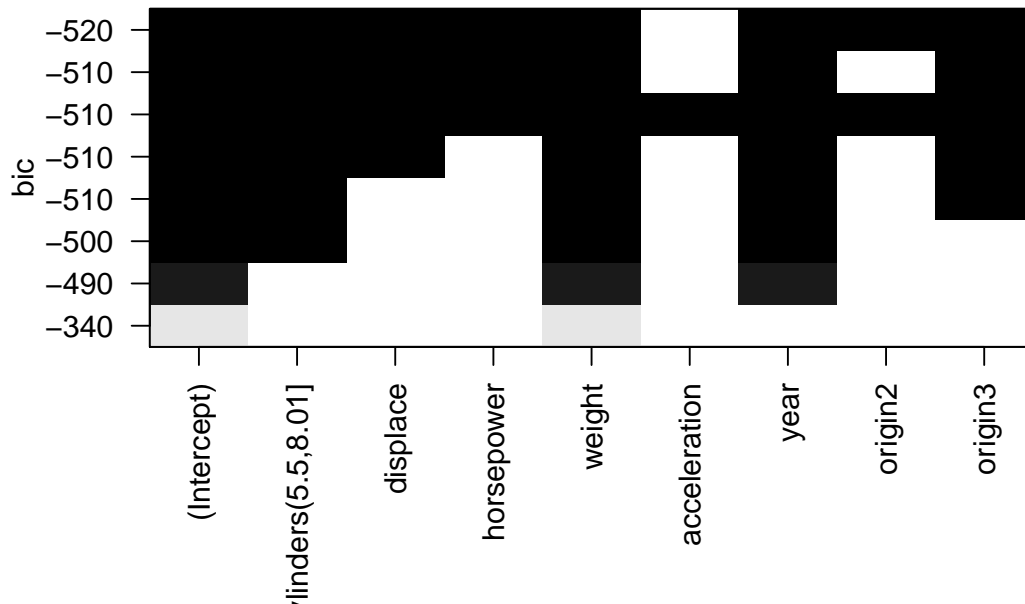
. The binomial theorem states that

$$(1+x)^n = \sum_{k=0}^n \binom{n}{k} x^k.$$

. By setting $x = 1$, we see that the number of different linear regression models is given by

$$\sum_{k=1}^d \binom{d}{k} = 2^d - 1.$$

- Q2: The following algorithm illustrates how the best subset method finds the best model amongst the $2^d - 1$ possible ones: Denote first by \mathcal{M}_0 the model using no predictors. This model simply predicts the sample mean for new observations. Next, for $k = 1, 2, \dots, d$, fit all $\binom{d}{k}$ models, by solving the equation $\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$, where \mathbf{X} and \mathbf{Y} only contains the predictors for one specific model. Calculate the RSS, and choose the model with the lowest RSS amongst the $\binom{d}{k}$ models, to be considered later. Now after the for-loop, we have $d + 1$ possible models to consider. Calculate $BIC = \frac{1}{n}(RSS + \log(d)k\hat{\sigma}^2)$ for each, and choose the model with the lowest BIC . This is the best model by the best subset method, using BIC -criterion. R^2 is given by $1 - RSS/TSS$. In this expression, we do not care how many predictors are being used. This is a problem, because training error is a poor estimate for test error, since it may cause overfitting. A possible solution to this is to add penalties, depending on how many predictors we use. If we were to use R^2 instead of BIC to choose the best model, our best model would surely contain all d predictors since over-fitting would make our model fit the training data better. This is unwanted as we would rather have a model that best fit test data, which BIC better simulates.



1b)

- Q3: As explained in Q2, RSS is calculated for all models, and for each model complexity, we choose the model with the lowest RSS as our best model. The `regsubsets`-function in R performs the actual calculations, and in the summary we can see which predictors corresponds to the lowest RSS for each model complexity. We see from this summary, which is printed in the task, that the best model with two predictors use the “weight” and “year” columns.
- Q4: Now assuming we have the best model for each model complexity, we still want to choose which model complexity to use. As discussed in Q2, we would have the smallest RSS for the highest possible model complexity in the training set, but since we want to avoid overfitting, we add penalties to more complex models. Therefore, for each model complexity, we add the penalties according to the BIC -criterion, and then choose the model with the lowest value as our best method. The `regsubsets`-function in R also does the calculations for the penalties, using BIC -criterion in addition to other penalty-adding methods. Thus we can from the `regsubsets`-function also retrieve the BIC -values for each model complexity, and choose the lowest one as our model. F Choose model and evaluate.

R code to fit your final model.

- Q5: Use this model fit to predict new values for `ourAutoTest` and report the MSE.

1c)

- Q6. Explain how k -fold cross-validation is performed.
- Q7. Why may k -fold cross-validation be preferred to leave-one-out cross-validation?

1d@9

- Q8. R-code for 10-fold CV.

```
library(caret)
library(leaps)
# and so on
```

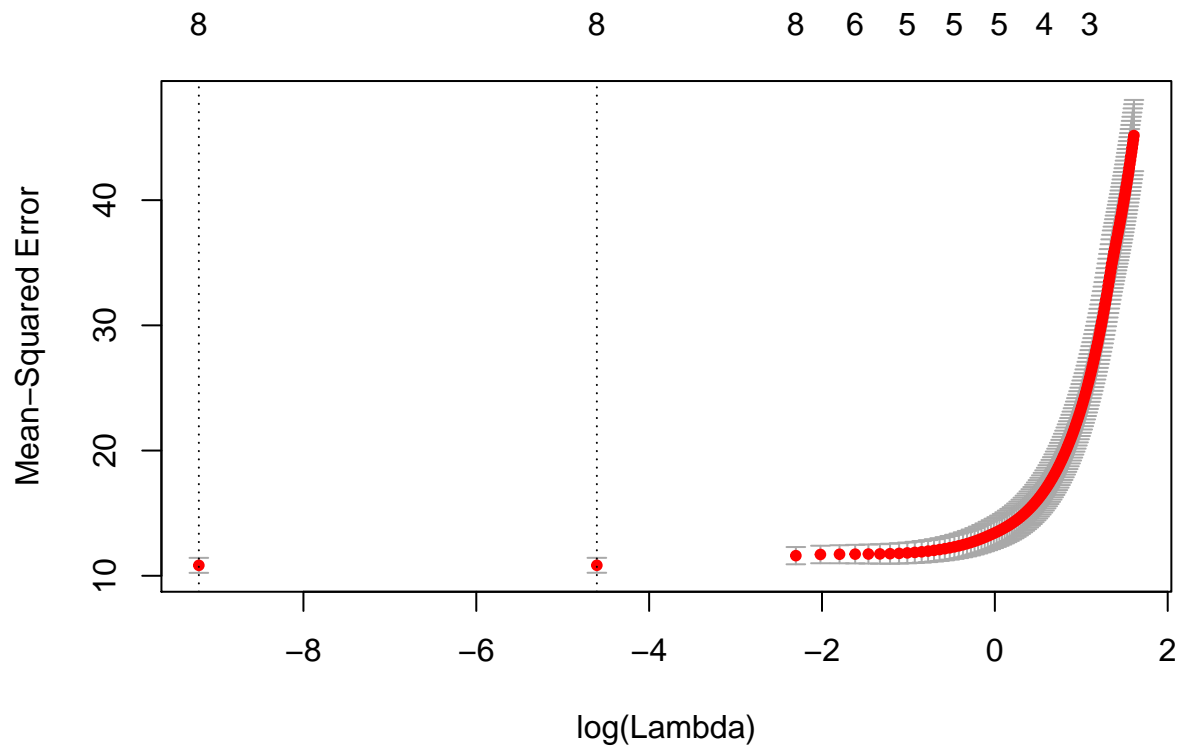
- Q9. What is the optimal model complexity (number of parameters) in your regression?

```
# MSE on test set
```

- Q10. Evaluate best model (or refer to Q4 and Q5).

2a) Explain figures

- Q11: Which figure (1 or 2) corresponds to ridge and which figure corresponds to lasso?
- Q12. Use the two figures and the above formulas to explain. . .
- Q13. Can you use lasso and/or ridge regression to perform model selection similar to what you did in Problem 1?



2b) Finding the optimal λ

- Q14: Explain what the function `cv.glmnet` does.
- Q15. Explain what we see in the above plot.
- Q16: Finding the optimal lambda:

```
# need some R code here
```

3c) Prediction

- Q17: Fit model, coefficients,...

```
# fit the lasso
```

```
# 0 for cylinder, displace, horsepower, weight, acceleration, year, 0 for origin2 and 0 for origin3  
newx=matrix(c(0,150,100,3000,10,82,0,0),nrow=1)  
# then do the prediction
```

- Q18: Predicted value:

3a)

- Q19: Fitting the specified gam

```
library(gam)  
# write R code
```

- Q20: The cubic spline basis.