Mariia Okuneva, M.Sc.

Data Mining
Home Assignment 2

This home assignment will help you understand why you should never use the same dataset to fit and evaluate your model. Furthermore, you will compare performance of validation set and cross-validation approaches. To that end, you will conduct a simulation study.

Consider following random variables

$$u \sim N(0,1) \qquad x \sim N(0,1)$$

and a model

$$y = 1 + x^3 + u.$$

Your task is to write an R script that contains the following parts. But first, download the script template *HA2_yourname.R* from OLAT.

1. Write a function for generating samples of a particular size of random variables x and y. The output of this function should be a data frame with x and y.

   Use this function to generate a dataset of size $n = 200$, split it into a train set and validation set (randomly choose 20% of your observations as a validation set).

   Plot the training data and the true regression line. If you use set.seed(90) before simulating the data, you might get the following result:

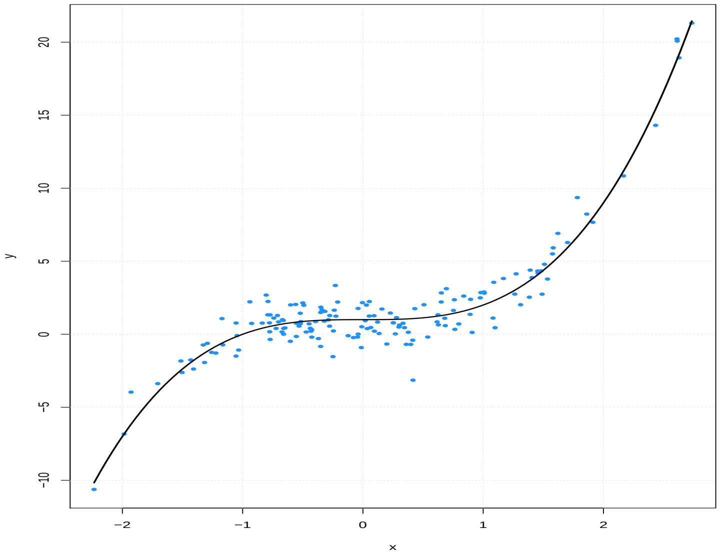

Figure 1: Training data and true regression line

2. Write two functions which you will use in the exercise 3.

   a) The first one (*calc_mse*) helps you to compute MSE and depends on two parameters: actual, predicted. Easy!

   b) The second one (*my.kfold*) is designed to compute test error by using cross-validation approach. Input: model (the model you fit), data, K (you divide data into K equal-sized parts). Output: CV error. If your data can not be split evenly into K parts, the function should produce an error message and set CV error to NA.

3. Now you are ready to set up a simulation study. Number of replications = 100.

   Compare polynomial fits of orders one to ten by using validation set and cross-validation methods (K=10). Also show why you should not use training data set to evaluate the model.

   Your pseudo-code would look like this:

```
For each replication
        begin
        Simulate data (n=200)
        Split the data into train and validation set as in 1 (80/20).
        For polynomial degree from 1 to 10
                begin
                Fit the model
                Calculate MSE for train data set (only to show that it is a bad idea)
                Use validation set approach to compute MSE
                Calculate CV error by using function from 2.
                end
        Choose polynomial degree based on training MSEs,
        validation set approach MSEs, and CV errors.
        end
```

   Now produce the table with the mean (over all simulations) and SD of training error, validation set error and CV error corresponding to the polynomial fits of orders one to ten. If you use set.seed(90) before the loop, your results might look like this:

Table 1: Simulation Results

Polynomidal degree	Mean, Train	SD, Train	Mean, Val	SD, Val	Mean, CV	SD, CV
1	6.67	3.88	7.07	8.36	7.18	3.85
2	5.84	2.99	7.44	8.95	7.60	4.36
3	0.99	0.10	1.05	0.26	1.04	0.09
4	0.98	0.10	1.11	0.39	1.07	0.12
5	0.97	0.10	1.22	1.36	1.13	0.19
6	0.96	0.10	2.29	8.94	1.63	1.87
7	0.96	0.10	28.32	248.16	4.18	13.09
8	0.95	0.10	110.76	1026.51	72.11	588.93
9	0.94	0.10	381.73	2266.67	59.35	188.24
10	0.94	0.10	2688.71	22745.16	2162.45	14335.52

4. How many times (out of 100) you choose polynomial degrees from 1 to 10 based on training

MSE, validation set error, CV error? Plot your results! If you used set.seed(90), your results might look like this:

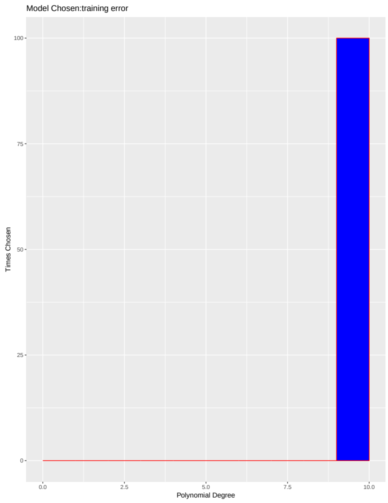

Figure 2: Training error

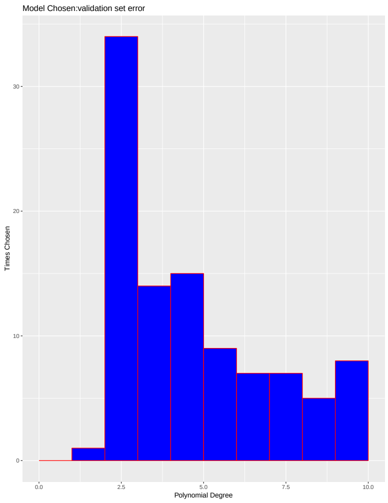

Figure 3: Validation Set

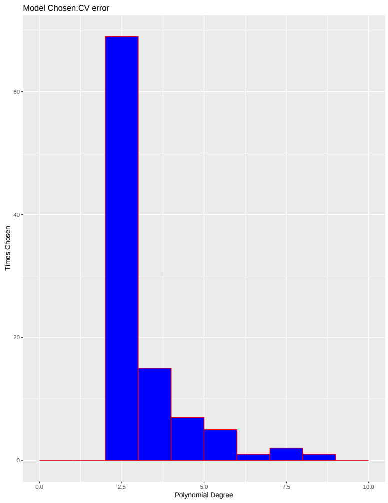

Figure 4: 10-Fold CV

5. Comment on your results. Remember that you know the polynomial degree of your true model.

Remarks: Write comments for everything you do. Codes that are not written using the template and/or that return error messages will not be evaluated.
Submission: Submit your scripts via email to *mokuneva[at]stat-econ.uni-kiel.de* until the end of June 13th (until 00:00:00, June 14th)