Machine Learning lab 2

Linear regression

Alice Maria Catalano

5157341

Abstract- Linear regression, in statistic, is a model that describes the relationship between two variables and the influence they have on each other. The goal of this process is to find a linear relationship between a target (dependent variable) and one or more predictors based on the measured data (independent variable).
It's one of the most important supervised algorithms, used in many fields to analyze behaviors and efficiency of something.

INTRODUCTION

Here will follow a brief description of the laboratory and the mathematical functions used.

1. Laboratory tasks

The dataset on which the linear regression algorithm needed to be applied were:

- The Turkish stock exchange data can be downloaded from the U.C.I. Machine Learning Repository
- The MT cars data are available as the command "mtcars" in the (open source) R statistical/data analysis language and environment,

The first task was to get them with a function that would make it readable by MATLAB.

On the data obtained, the fit to the different linear regression models seen in class which are:

1. One-dimensional problem without intercept on the Turkish stock exchange data
2. Compare graphically the solution obtained on different random subsets (10%) of the whole data set
3. One-dimensional problem with intercept on the Motor Trends car data, using columns mpg and weight
4. Multi-dimensional problem on the complete MTcars data, using all four columns (predict mpg with the other three columns)

The last task is a test on the regression model, re-running points 1,3 and 4 from the second task using the square error loss (MSE) as objective function on different sizes of the dataset.

2. Linear regression problem
   The linear regression is seen as an optimization problem in which the functional dependency between measured data is approximated. The approximation is applied by the parameter *w* on the model that predicts the target *t* with a given observation *x*, good for every point. It forecasts one continuous variable using observations (= one or more other variables) related to it. The generic goal is to minimize the mean value of the loss over the whole data set:

$$J = \frac{1}{N}\sum_{l=1}^{N}\lambda(y_l, t_l)\,.$$

J is the objective function or cost function,
N is the number of observations,

λ is the loss function,
t_l measured target value,
y_l is the inferred target value.

In this laboratory the particular square error loss function was used.

$$J_{\text{MSE}} = \frac{1}{N}\sum_{l=1}^{N}(y_l - t_l)^2$$

The goal is to minimize the mean square error objective with respect to fixed data.

1. One-dimensional problem without intercept
   This is the value of *w* obtained

$$w = \frac{\sum_{l=1}^{N} x_l t_l}{\sum_{l=1}^{N} x_l^2}$$

2. One-dimensional problem with intercept
   The solution in this case can be found by centering around the mean $\bar{x}$ of x and $\bar{t}$ of t.

$$\bar{x} = \frac{1}{N}\sum_{l=1}^{N} x_l \qquad \bar{t} = \frac{1}{N}\sum_{l=1}^{N} t_l$$

   Here we switch from a linear to an affine model
   $y = w_0 + xw_1$

$$w_1 = \frac{\sum_{l=1}^{N}(x_l - \bar{x})(t_l - \bar{t})}{\sum_{l=1}^{N}(x_l - \bar{x})^2}$$

$$w_0 = \bar{t} - w_1\bar{x}$$

   With the following gain and intercept
   In which
   w_1= slope; gain
   w_0= intercept, offset; bias.

3. Multi-dimensional problem $\mathbf{w} = (X^T X)^{-1} X^T \mathbf{t}$
   The data is now composed of d-dimensional vectors, stored into a N x *d* matrix, in which: N is the number of observations, *d* is the number of features observed. Since the data are now d-dimensional, we have d parameters in a d-dimensional vector. The linear model now results like $y = Xw$ where:

IMPLEMENTATION

1. Task 1: get the data
   As already said here, the data upload was asked suggesting 2 functions: "load" or "*csvread*", but because of the mixed structure of the data with numbers and strings the "*readmatrix*" function was used instead.
2. Task 2: fit a linear regression model
   Here the results of each regression model will be shown.
   a- One-dimensional problem without intercept
   The function, explained in the previous paragraph at the subsection dedicated to this problem, was implemented on the "*linearRegression.m*". This function will calculate the value of *w* and plot it with the following result.

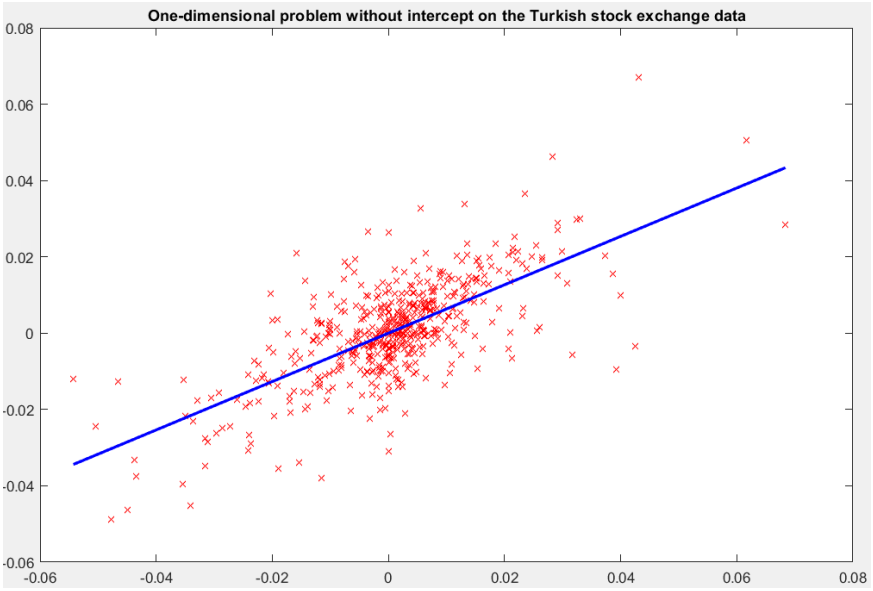

Figure 1- 1D linear regression without intercept

b- Compare graphically the solution

The comparison should be executed on the 10% of the whole dataset of the stock market. With the function "*compare.m*" the dataset is split in 9 subsets as shown in Figure 2:

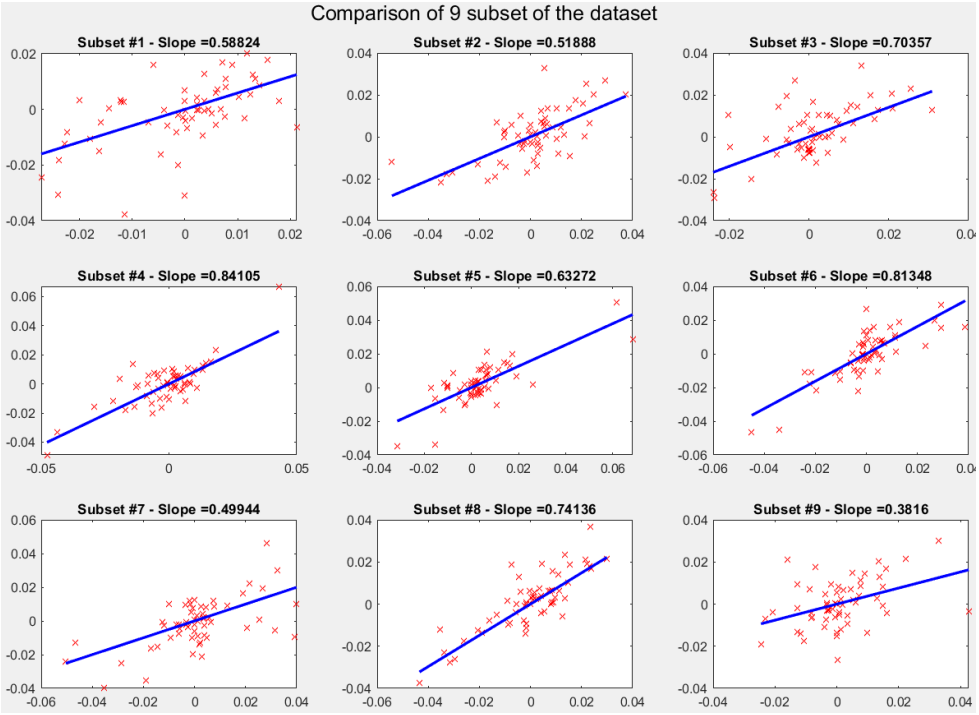

Figure 2- Comparison of different subset of stock market data

c- One-dimensional problem with intercept

To predict the mpg and weights features from the motor car trends data. The implementation of the formulas described before in the subsection dedicated to this task is in the "*linearReg_offset.m*". The result is in figure 3.

d- Multi-dimensional problem

The prediction of the mpg with all the other feature is done. It was needed to implement the equation of the regression problem in the function "*linearReg_mD.m*" which will print a table with the dataset values inferred and not (Figure 4)

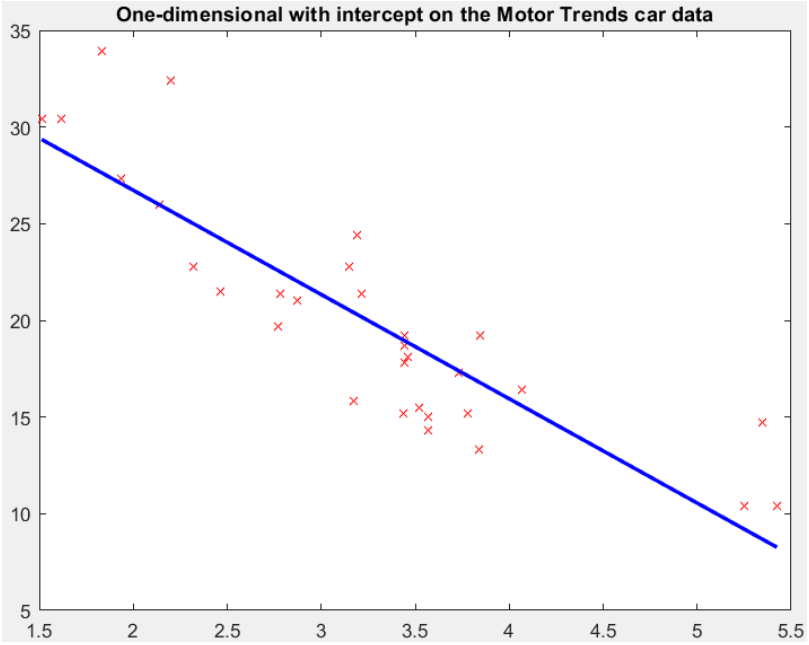

Figure 3- 1D linear regression with intercept

	disp	hp	weight	dataset mpg	Predicted mpg
1	160	110	2.8750	21	20.6889
2	108	93	2.3200	22.8000	19.3273
3	258	110	3.2150	21.4000	13.4010
4	360	175	3.4400	18.7000	7.1052
5	225	105	3.4600	18.1000	19.9438
6	360	245	3.5700	14.3000	11.7141
7	146.7000	62	3.1900	24.4000	23.9120
8	140.8000	95	3.1500	22.8000	25.5554
9	167.6000	123	3.4400	19.2000	27.1433
10	167.6000	123	3.4400	17.8000	27.1433
11	275.8000	180	4.0700	16.4000	24.6319
12	275.8000	180	3.7300	17.3000	20.5587
13	275.8000	180	3.7800	15.2000	21.1577
14	472	205	5.2500	10.4000	17.1127
15	460	215	5.4240	10.4000	21.0243
16	440	230	5.3450	14.7000	23.0503
17	78.7000	66	2.2000	32.4000	20.1094
18	75.7000	52	1.6150	30.4000	12.8386
19	71.1000	65	1.8350	33.9000	16.5742
20	120.1000	97	2.4650	21.5000	19.8360
21	318	150	3.5200	15.5000	11.8429

Figure 4- multi-dimensional regression

3. Task 3: Test a regression model

   As already described in the introduction paragraph, in this task the MSE was computed on the training set (5% of the whole data) and apply the model obtained on the remaining 95%.

	Dataset	Percentage	MSE
1	Train set	4.8507	8.2284e-05
2	Test set	94.9627	9.6601e-05

Figure 5- 1D case without offset

	Dataset	Percentage	MSE
1	Train set	3.2258	NaN
2	Test set	93.5484	NaN

Figure 6- 1D case with offset

	Dataset	Percentage	MSE
1	Train set	3.2258	2.3042e-29
2	Test set	93.5484	1.3351e+03

Figure 7- multi-D case

CONCLUSIONS

Some of the results obtained are pretty clear. In Figure 1 we have just a representation of simple linear regression algorithm working on the dataset.

In Figure 2 we can see that the slope obtained for each subset is different from the others, because of the reduced dimensions of the subsets.

Figure 3, as figure 1, is just the representation of the model used.

Figure 4 is the representation of the data calculated by the equation defined in the introduction paragraph.

Figure 5,6,7 display how the MSE on the 95% of the dataset is higher than the MSE on the train set. That’s because the model overfits the data, so the model can’