Report Lab2 for the Machine Learning course

Nicholas Attolino

October 2023

Contents

Introduction

This work is a report for the second laboratory of the Machine Learning course. The subject of this laboratory is the **Linear Regression**, the possibility of fitting a linear model to two or more variables relationship. The programs and the tests are implemented in MATLAB, a versatile platform that serves as a numerical computation and statistical analysis environment, as well as a programming language[2]. The chapters are divided in:

1. Goal;
2. Methods used;
3. Results.

1 Goal

The objective of this project is to create multiple Linear Regression models within the MATLAB environment.
What is a *Linear Regression*? The Linear Regression is a method that linearly models the connection between a singular output and one or more predictor variables, which are often referred to as the dependent and independent variables[1].
In this work, thehe models object focus on two datasets:

- turkish-se-SP500vsMSCI, a 536 by 2 matrix dataset;
- mtcarsdata-4features, a 32 by 5 matrix dataset.

After creating the models, multiple tests shall be executed.

1.1 Data Files overview

```
- Attolino-Lab2.zip
    - Attolino-Lab2
        - Linear_Regres_OneDim.m
        - Mean_Square_Error_OneDim.m
        - MotorTrends_Dataset.m
        - mtcarsdata.csv
        - turkish.csv
        - Turkish_Dataset.m
```

In the zip archive there are the files shown above, where we find:

- Linear regression function for one-dimensional problem;
- Mean Square Error function;
- Script of the tasks relatives to the Motor Trends Car dataset;
- Motor Trends Car dataset;
- Turkish dataset;
- Script of the tasks relatives to the Turkish dataset.

2 Methods used

2.1 Dataset elaboration

To load and make the datasets usable, the *readmatrix* function was used.
In particular, the Motor Trends Car dataset is split in 4 different column vectors to make it easier to work on and in this case the *Motor_Trends_Car_load* function has been made since it's often needed to repeat the division of the whole dataset.

2.2 One-dimensional case

Considering linear regression as an optimization problem, the goal is to find the parameters that minimize the mean value of loss over the entire dataset.
The mean value is the objective function and, choosing the mean square error as loss function, it looks like this:

$$J_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2 \tag{1}$$

where N is the number of observations, t the target and y the estimated value.

It's possible to find the minimum of the objective when its derivative is equal to zero.
When this happens, it's proved that:

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

Thanks to this formula, the developed function called *Linear_Regres_OneDim* is able to fit a regression model: it receives in input a matrix of two columns, one for the observations and one for the targets, and returns the slope parameter.
To approximate the target, it is necessary to multiply the slope parameter with the observation x.
In particular, when this function receives the string *'withOffset'* as additional input, it will also return the offset parameter w_0:

$$w_0 = \bar{t} - w_1 \bar{x} \tag{3}$$

where $\bar{x}$ and $\bar{t}$ are respectively the mean of x and t:

$$\bar{x} = \frac{1}{N} \sum_{l=1}^{N} x_l \qquad\qquad \bar{t} = \frac{1}{N} \sum_{l=1}^{N} t_l \tag{4}$$

In this case the slope have a different formula since it's computed by centering around the mean of the observations and the targets:

$$w_1 = \frac{\sum_{l=1}^{N} (x_l - \bar{x})(t_l - \bar{t})}{\sum_{l=1}^{N} (x_l - \bar{x})^2} \tag{5}$$

2.3 Multi-dimensional case

As for the multidimensional linear regression problem, it works differently: now there are d parameters and the data are composed of d-dimensional vectors so X is a matrix of N by d.
The goal becomes to make the model $\boldsymbol{y} = \boldsymbol{Xw}$ as similar as possible to the N-dimensional vector of the targets $\boldsymbol{t}$.
Accordingly, the minimum of the objective

$$J_{MSE} = \frac{1}{2} \|\mathbf{y} - \mathbf{t}\|^2 \tag{6}$$

is obtained by setting its gradient $\nabla J_{MSE} = 0$ and because of this, we will have:

$$w = (X^T X)^{-1} X^T \mathbf{t} \tag{7}$$

where $(X^T X)^{-1} X^T$ is the Moore-Penrose pseudoinverse of X, a formula that allow us to compute $\boldsymbol{w}$.

Finally we have:

$$\mathbf{y}^* = \mathbf{w}^* \cdot \mathbf{x}^* \tag{8}$$

where the value $y*$ estimates the most probable value of the output when a point $\boldsymbol{x*}$ is received.

3 Results

3.1 One-dimensional problem without intercept for Turkish dataset

After invoking the linear regression function on the Turkish dataset, it yields the slope parameter as the output.
To visualize the way the function models the data, it's beneficial to create a plot comparing the first column of the dataset (observations) to the first column multiplied by the parameter w.

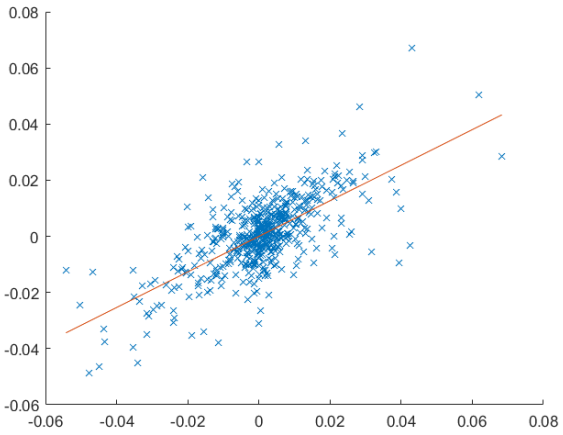

Figure 1: Scatter plot of the dataset with its least squares solution

The same test has been also carried out for ten times with random subsets of about 5% of the whole dataset showing this result:

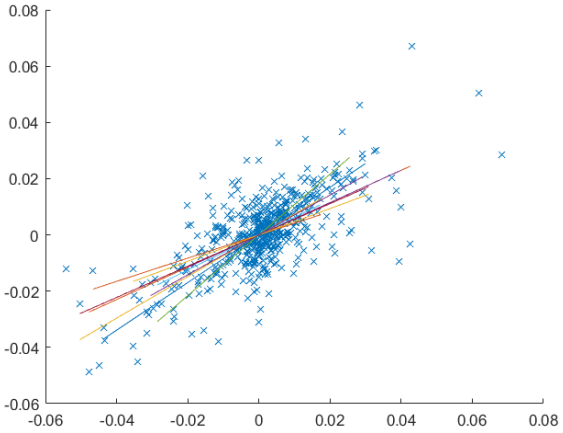

Figure 2: Scatter plot of the dataset with its least squares solutions

For each test, the software calculates the objective function for both the 5% data subset and the remaining 95% data subset.
As depicted in Figure 3, it's evident that the objective function value for the 5% subset is definitely lower than that for the 95% subset, demonstrating improved performance on the "training" data. This discrepancy arises from the model being specifically tailored to that dataset.

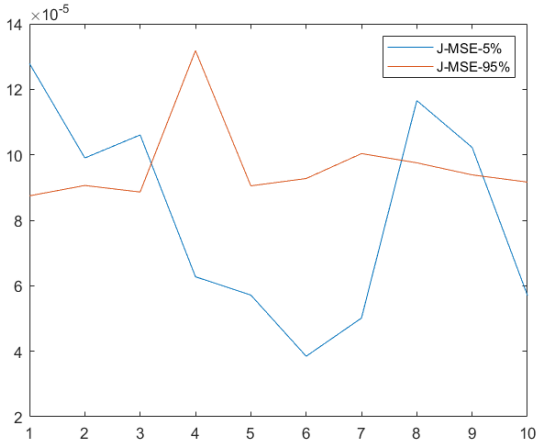

Figure 3: J_{MSE} over the 10 tests

3.2 Graphical comparison of solutions from two different Turkish subsets

The program provides users with the option to select between obtaining random data subsets from different ends of the Turkish dataset or from any point of it. While the preprocessing steps share similarities in both scenarios, they are not identical, necessitating the use of two distinct sets of commands.
In the case of obtaining random subsets from the different ends, the program first extracts the initial and final 20% of the complete dataset and subsequently selects half of the observations randomly from each of these subsets. This results in two subsets, each containing 10% of the total observations, drawn from the two extremes of the entire dataset.
Acquiring observations from any point of the dataset is a more straightforward process: the program generates n random indices and employs them to store the observations, where n corresponds to 10% of the total number of observations.

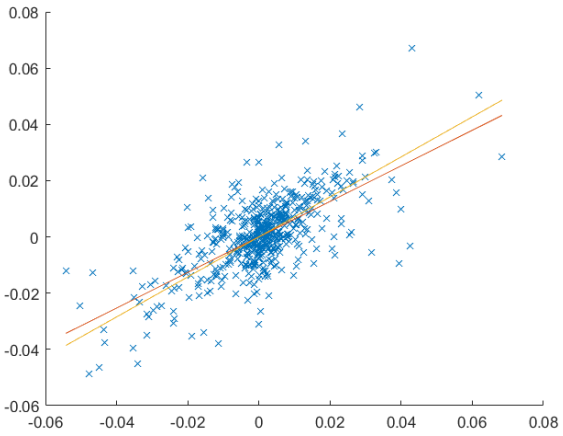

Figure 4: Comparison between solutions of different subsets

The point of taking data from different ends of the data set is that, since the data are collected across time, data collected in similar periods may be more similar than data collected from the beginning and the end of the whole period. In fact, in this case, the subsets' distribution are much different and so are their approximations.

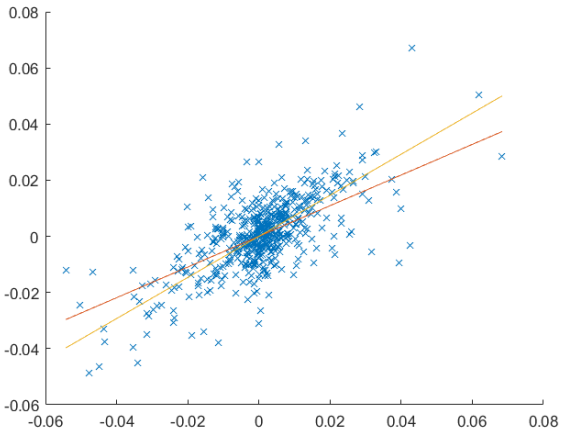

Figure 5: Comparison between solutions of different ends of the dataset

It's quite evident that the behavior remains fairly consistent when working with a random subset, resembling the behavior when the entire dataset is considered. This is primarily because the sampling algorithm selects instances that likely encompass the entire time span. These instances suffice to grasp the dataset's average behavior.

3.3 One-dimensional problem with intercept for Motor Trends Car

The program uses the weigh of a car to approximate its MPG (Miles Per Gallon). Figure 6 displays a scatter plot of these two attributes along with their linear approximation. In this case, the linear regression function also accepts the argument for requesting the intercept, which corresponds to the second output of the same function.

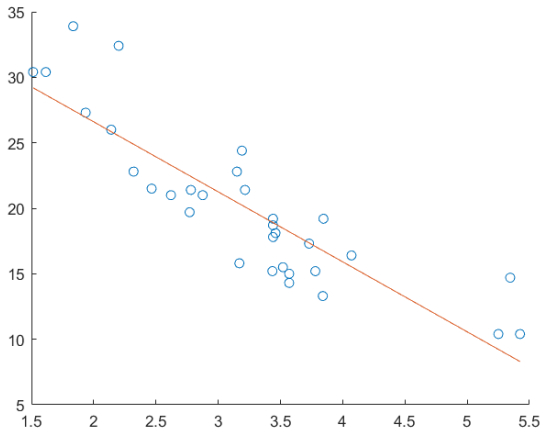

Figure 6: Weight vs MPG, linear approximation with intercept

An identical test was conducted ten times using random subsets, each comprising approximately 5% of the entire dataset. When fitting a model to only 5% of the 32 observations, it involves selecting just two data points and connecting them with a straight line:

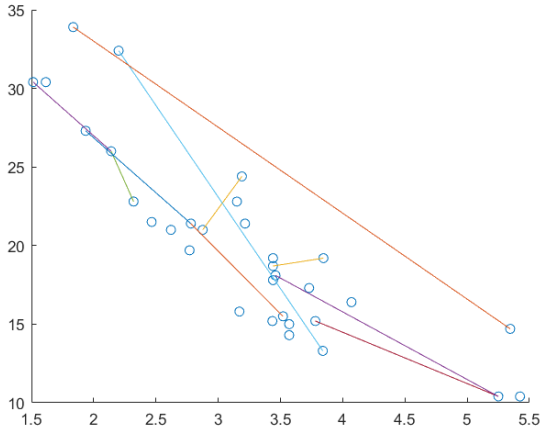

Figure 7: Scatter plot of the subsets with its least squares solutions

As there's only a single line connecting two points, the resulting slope parameter ensures that the algorithm will not miss, causing the objective to converge towards zero (on average, $J_{MSE} = 7.2575 \cdot 10^{-30}$).

When it comes to applying that parameter to the remaining observations, the algorithm will struggle to accurately estimate the relationship between the points (on average, $J_{MSE} = 0.8796 \cdot 10^2$).

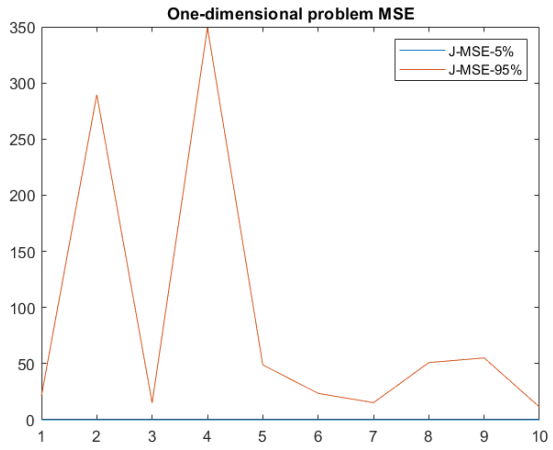

Figure 8: J_{MSE} over the 10 tests

This is the overfitting phenomenon: the creation of an analysis that matches a specific dataset too closely or perfectly, which can subsequently result in a failure to adapt to new data or make accurate predictions for future observations[3].

3.4 Multi-dimensional problem for Motor Trends Car

The program aims to discover a linear correlation between a car's Displacement, Horsepower, and Weight, and its Miles Per Gallon (MPG).
The provided solution managed to determine a slope parameter in this multidimensional scenario.
Just like in the one-dimensional problem, running the regression multiple times with an insufficient number of data points results in overfitting (on average, $J_{MSE} = 2.4526 \cdot 10^5$).
However, in this case as well, the objective for the 5% subset is higher than expected (on average, $J_{MSE} = 1.5907 \cdot 10^4$). This suggests that the relationship between the analyzed parameters is far from being linear.

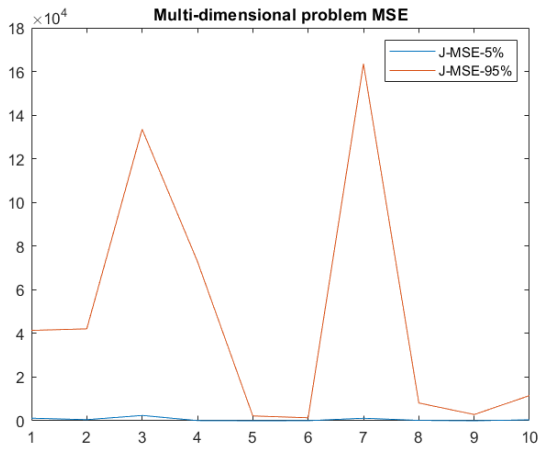

Figure 9: J_{MSE} over the 10 tests

4 Conclusions

As anticipated, a linear association between two variables can be quantified within the parameters of a model, typically yielding a minuscule margin of error.
However, challenges arise when dealing with sparsely populated datasets or when no linear correlation exists among certain attributes. In the former scenario, overfitting is likely to occur, while in the latter, the outcome is a considerably larger objective value and subpar model performance.
Furthermore, a visual illustration was presented, showcasing how randomly selected data points can effectively capture the overall behavior of the entire dataset.

References

[1] D. Freedman, *Statistical Models : Theory and Practice*. Cambridge University Press, 2005, ISBN: 0521854830.

[2] T. M. Inc., *MATLAB Version: 23.2.0.2365128 (R2023b)*. The MathWorks Inc., 2023.

[3] IBM, *What is overfitting?* [Online]. Available: `https://www.ibm.com/topics/overfitting`.