## 1 Linear Regression

### 1.1 Derivation of regularized Linear Regression

#### 1.1.1 Preliminary questions

1. The design matrix X is derived from the Error function of polynomial functions for writing it as a product of a vector and a matrix. In this error function we need to find a partial derivate for , for getting equations. The fact that we introduce an x0 (for convention = 1) involves that the matrix (and tor will have a size of m x (n +1) and in the case of the vector of 1 x (n +1).
2. If we consider the cost function , the gradient represents a vector containing the partial derivative of the cost function with respect to every component of the vector and it can be calculated as follows.

Let

Then

The result is an columns vector.

If we want to give a meaning, the gradient denotes the direction of greatest change in the cost function.[[1]](#footnote-1) To minimize the cost function (e.g. find the best parameters ), the gradient must be set equal to zero.

1. The Jacobian matrix is a generalization of the gradient. In general we talk about gradient when we are working on functions that maps , so the Jacobian matrix in this case would just be a vector. We talk about Jacobian matrix when working on functions mapping .[[2]](#footnote-2)

If we have a vector function that maps , where

The Jacobian matrix is defined as

1. The Jacobian matrix is a rows, columns matrix of the form

The first row of the matrix contains the partial derivative of the first element of the hypothesis vector w.r.t. each of the parameters contained in the vector .

#### 1.1.2 Regularized linear regression optimal parameters

The cost function for the regularized linear regression is defined as follows.

We want to show that

Where represents the best fitting parameters to minimize the cost function.

We start by taking the partial derivative of with respect to .

Here we used the fact that .

Now we want to set and isolate .

If we use the fact that , we get

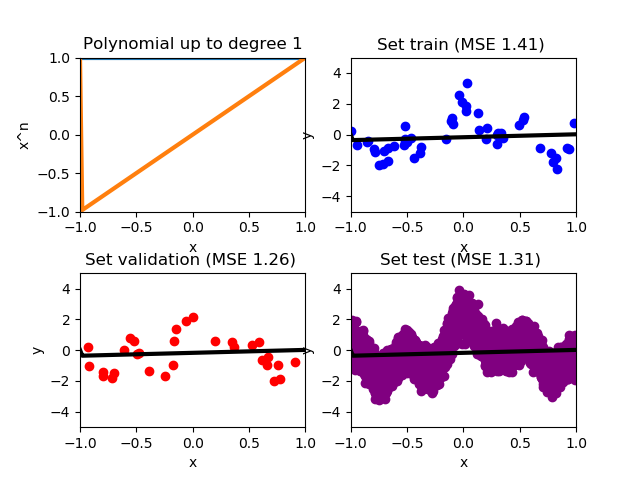
## 1.2 Linear Regression with polynomial features

### 1) Plot Poly with different degrees

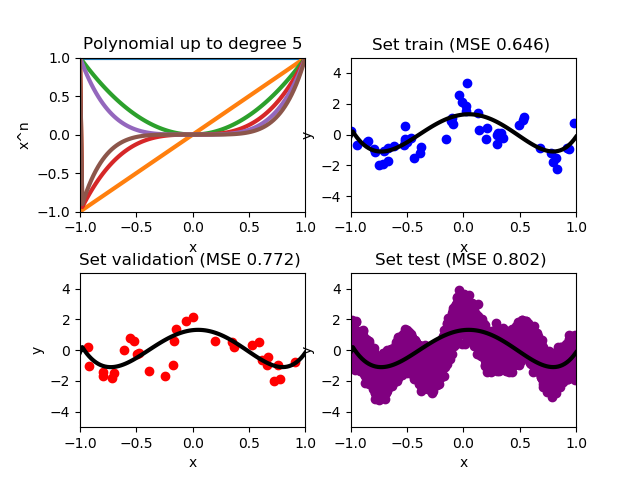
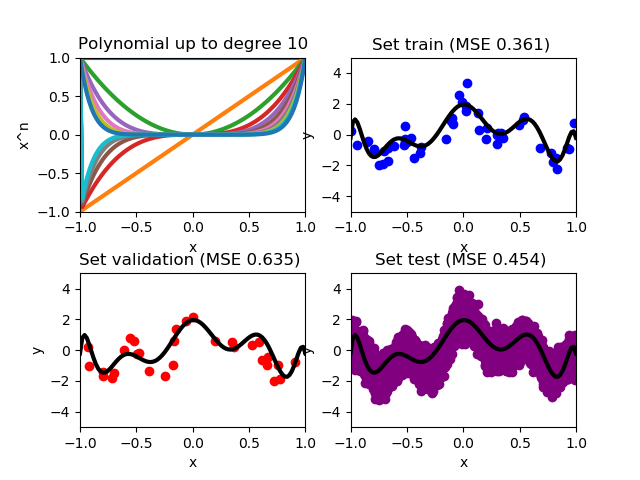
This table shows the mean square error values for each data set and for a function degree of 1, 5, 10 and 22. As you can see except for the error for testing set with degree = 22 the biggest errors are found in the graphs with function degree 1. This is because it is difficult to approximate a data set to a line (function of degree = 1).

What is interesting to note is that with higher degree functions the error is reduced but up to a certain point above which a higher degree negatively influences the value of the error (the smaller error for each data set is highlighted in green).

|  |  |  |  |
| --- | --- | --- | --- |
| **Degrees / Data error** | **Training set mse** | **Validation set mse** | **Testing set mse** |
| **1** | 1.4098850380043075 | 1.2591711733047637 | 1.3097844676162802 |
| **5** | 0.6456691914460443 | 0.7717206940485468 | 0.8020970590101822 |
| **10** | 0.36118151579299157 | 0.6347917160195701 | 0.453719176366401 |
| **22** | 0.33903781756998747 | 0.9190236574545647 | 5.20235874300072 |

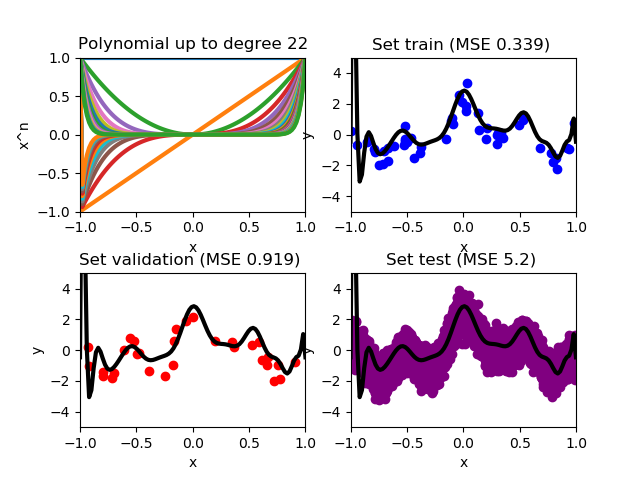
Here below are the graphs representing the data in the table.

Function of degree = 1



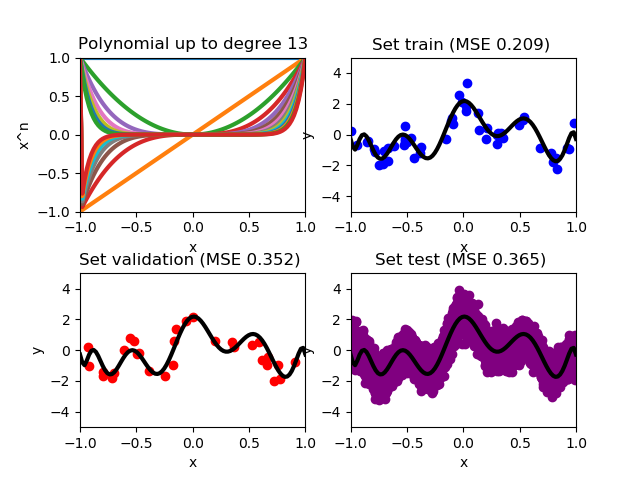
Function with degree = 10

Function with degree = 5

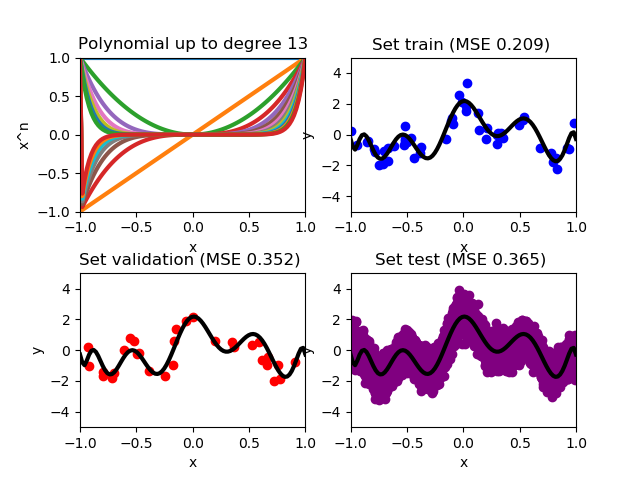


Function with degree = 22

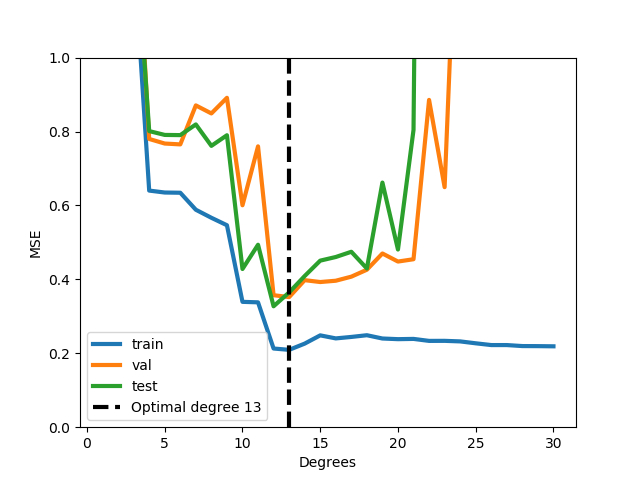
### 2) Best degree for training set

For this step we have chosen to take lamda = 0.55 to have a strong regulation, as we will see later the bigger lamda the more the training set and validation set will look like each other. Here below the graph of the function in every set of data with the optimal degree founded for the training set (13). With a lamda even slightly lower and therefore a weaker regularization we would have completely different results.

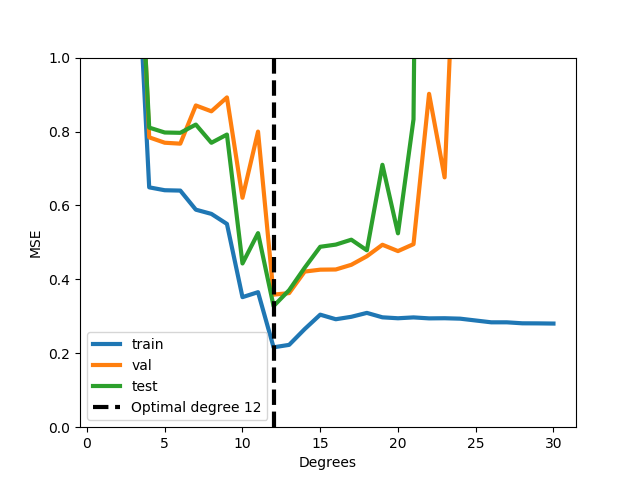
### 3) Best degree for validation set

Here below the graph of the function in every set of data with the optimal degree founded for the validation set (13). Lambda is always = 0.55.

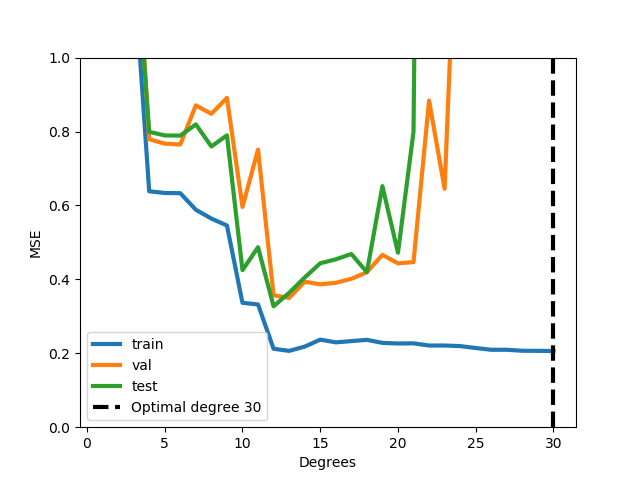
### 4) Error functions

Here below the graph of the function of errors for the different sets of data is shown. As we can see the functions have the optimal cost that coincide with the optimal degree.

The test set has a MSE that is smaller than the MSE of function with degree = 13 for a function with degree = 12. This is probably due a not so high regularization. In fact if we increase Lambda = 0.8 we get an optimal degree for validation and training set = 12 (as shown below).

If instead of increasing lambda we reduce it (e.g. lambda= 0.5) we obtain different optimal degrees for each data set :

END GRAPH



Cost functions and optimal degree for the training set with lambda = 0.5

### 5) Discussion

Cost functions and optimal degree for the validation set with lambda = 0.5

What we can deduce from the derived data and graphs is that thanks to the regularization we can improve the values of the training set in order to make them more truthful and reliable and removing the problem of overfitting. The validation set in this case serves just to this, in the moment in which the results obtained from the training set approach those obtained from the validation set we could consider them valid.

In more we have discovered the differences between a weak regularization and a strong regularization, in fact with the variation of lambda also changes the degree of optimal function. In our case we took for the final result lambda = 0.8 so that the optimal functions for the 3 data sets coincide in a function of degree 12.

### 6) Final answers to question 2 & 3

Considering the fact that a strong regularization is better than a weak one we have chosen as final value of lambda 0.8 obtaining therefore, as already said before, a function of grade 12 as function that reduces to the minimum the costs for all data sets. (graph showing the error functions is shown above – END GRAPH)

1. <https://en.wikipedia.org/wiki/Gradient> [↑](#footnote-ref-1)
2. <https://math.stackexchange.com/questions/1519367/difference-between-gradient-and-jacobian> [↑](#footnote-ref-2)