Regression

Giuseppe L'Erario

Introduction

The regression models are used for prediction problems applied on continuous data.

The basic concept is the "construction" of a function able to approximate the target values given the train data (in other words, a function describing the phenomenon), in order to make a prediction upon new data.

First, a *Linear regression model* is analysed, then *Polynomial regression models*.

1 Linear regression

The equation of a linear model is:

$$y = w_0x_0 + w_1x_1 + ... + w_nx_n = \sum w_ix_i = w^Tx \tag{1}$$

where w_0 is the intercept of the target value axis, and $x_0 = 1$.

When the phenomenon is described only by one variable, the model is a simple straight line:

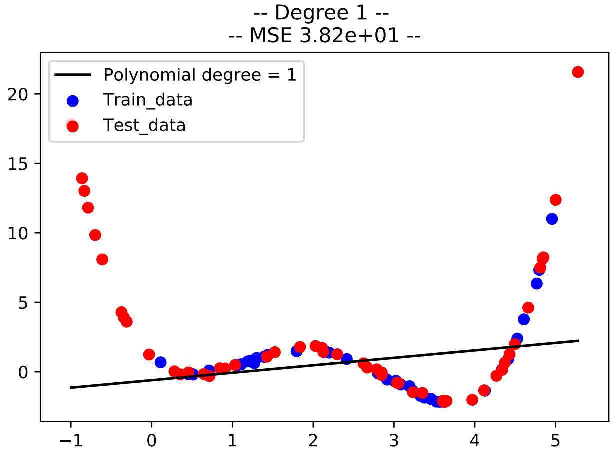
$$y = w_0 + w_1x \tag{2}$$

After the loading of the train/test data set, I create the object *PolinomialFeatures* from library *sklearn* and than fit the model:

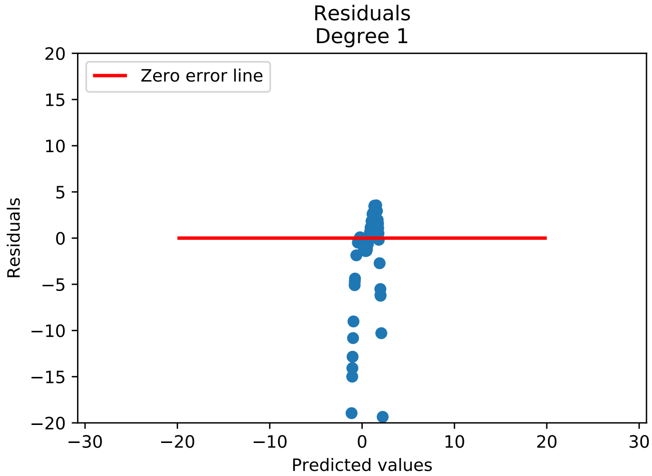
```
poly=PolynomialFeatures(degree=i, include_bias=False)
xPoly=poly.fit_transform(X_train.reshape(-1,1))
lr=linear_model.LinearRegression()
lr.fit(xPoly, Y_train)
```

The results are shown in fig.1(a). It is clear that the phenomenon is not linear and its point cannot be interpolated by a linear function. The residuals are not concentrated around zero error line (fig.1(b)) and the values of the predictione do not cover the *Image* of the function[1]. The *mean square error* is equal to $3.83 * 10$.

[1]The image of the function has a range between -2 and 20, while the values of the prediction are around zero (approximately flat line).



(a) Interpolation



(b) Residuals

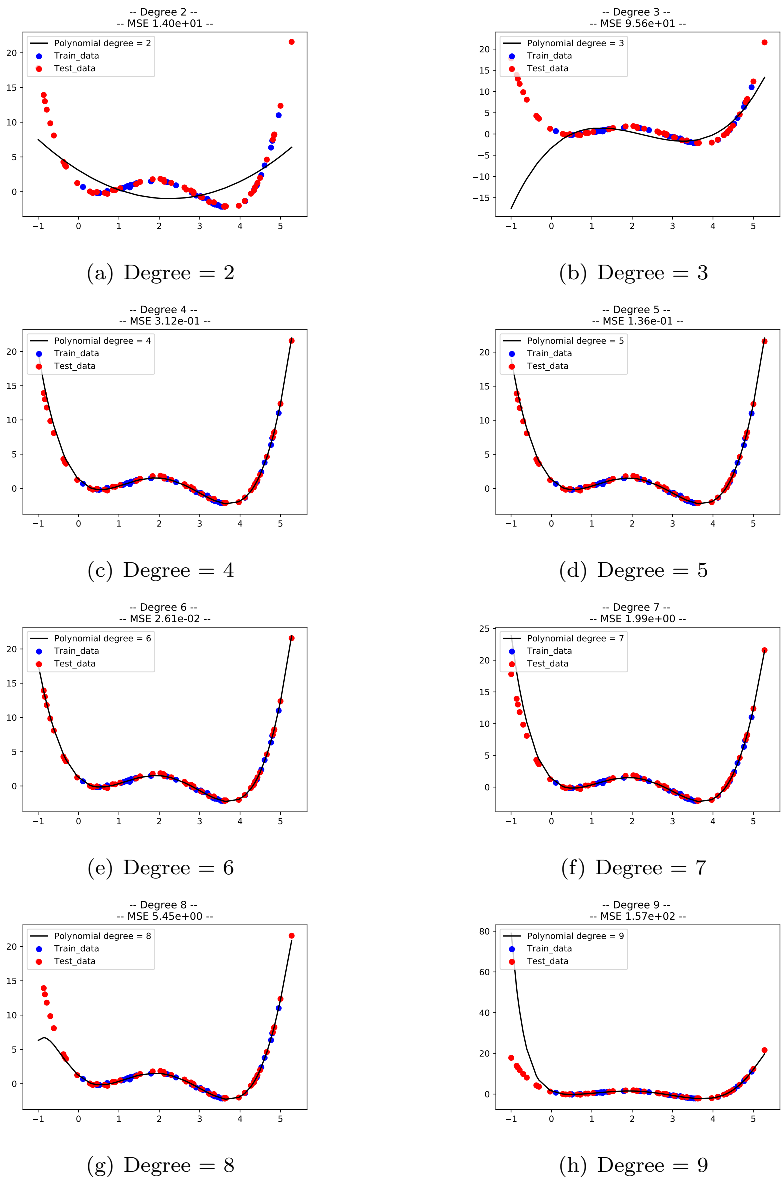
Figure 1: Linear Regression

2 Polynomial regression

With a polynomial regression the function that maps the independent variable and the value function is modelled as a polynomial of n grade:

$$y = w_0 x^0 + w_1 x^1 + w_2 x^2 + ... + w_n x^n \tag{3}$$

If the phenomenon is not linear a better prediction can be achieved with a polynomial function, taking care to find the good balance between wrong prediction and overfitting.



(a) Degree = 2 (b) Degree = 3

(c) Degree = 4 (d) Degree = 5

(e) Degree = 6 (f) Degree = 7

(g) Degree = 8 (h) Degree = 9

Figure 2: Polynomial Regression

The *Mean Square Error* is plotted in fig.3.

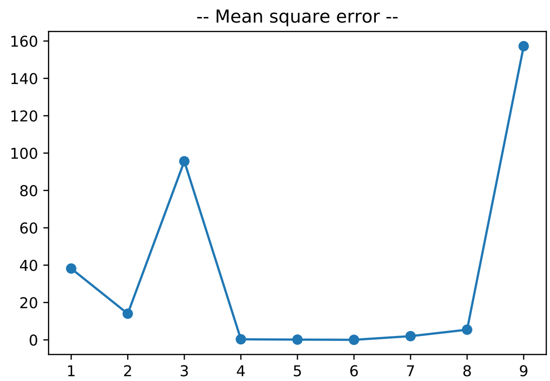
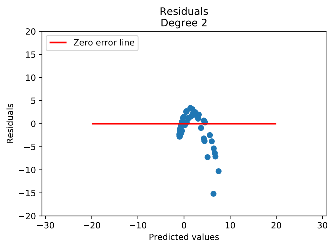


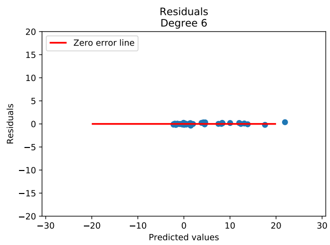
Figure 3: Mean square error

The lowest error is reached with 6^{th} degree polynomial, in fig.2(e), with a $MSE = 2.608 * 10^{-2}$. We can observe a similar behaviour in polynomials from degree 4 to 8.

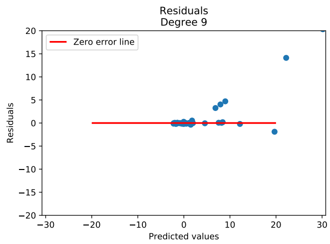
The 9^{th} degree polynomial (fig.2(h)) presents a sensible overfitting, while the 3^{rd} degree polynomial (fig.2(b)) has a great error because of its shape.



(a) Degree = 3



(b) Degree = 6



(c) Degree = 9

Figure 4: Residuals of $3^{rd}, 6^{th}, 9^{th}$ degree polynomials

The gap between the test values and the prediction values is presented in fig.4: the values in fig.4(b) are all around zero line error, confirming that 6^{th} degree polynomial regression leads to the best performances.