A basic supervised problem

Introduction to Machine Learning

University of Barcelona

November 27, 2015

Authors:
KAZANCLI, Erol
LEYVA, María

Contents

1 Data set analysis

1. **Plot the training samples and their corresponding label.**

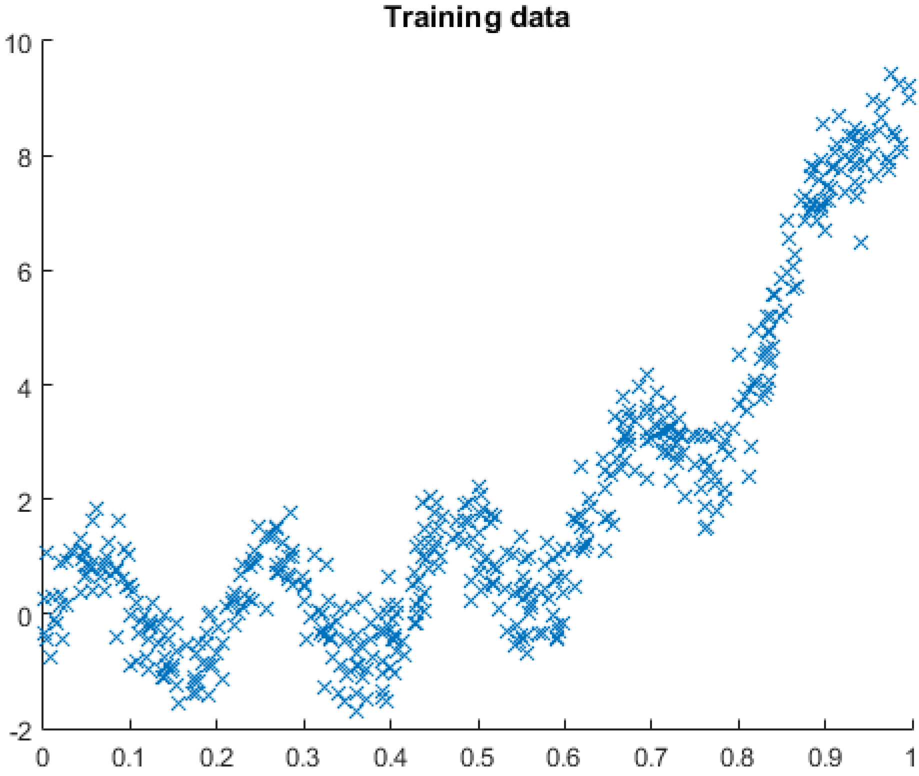

Figure 1: Training data

2 Analytical solution for the linear regression method

1. **Which is the optimal value of the linear regression weights?**

```
w =

  -2.047410273675812
   7.994792918349567
```

Which means that the function is:

$$f(x) = -2.047410273675812 + 7.994792918349567x$$

2. **Plot the data set and the line learned by the model. Does it looks like a good linear approximation?**

As we can see in the figure, it seems to approximate quite well for being a linear regression.

To have more information, we calculate the cost function:

$$J(w0, w1) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i; w_0, w_1) - y_i))^2$$

```
cost =

   2.587669494785314
```

We get that the average distance between the real y and our regressed y is quite big, though it is probably the best we can do using a linear approximation. Maybe we should consider other alternatives.

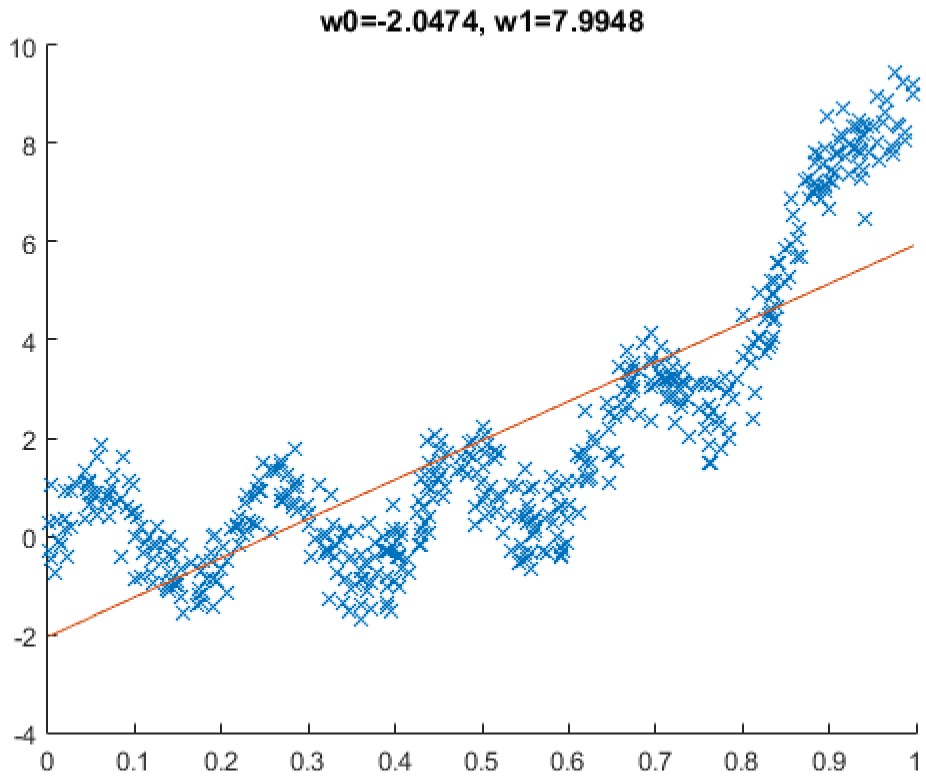

Figure 2: Linear regression

3 Linear regression and the descent method

1. **Which is the optimal value of the linear regression weights using the descent method?**

```
w_001 =

  -0.004963993893897
   4.211390018549198
```

Which means that the function is:

$$f(x) = -0.004963993893897 + 4.211390018549198x$$

2. **Which are the parameters of the descent method used to obtain the optimal value?**

Learning rate: 0.01

Iterations: 1000

Initial weights: $w_0 = 0, w_1 = 0$

3. **Plot the convergence curve of the method.**

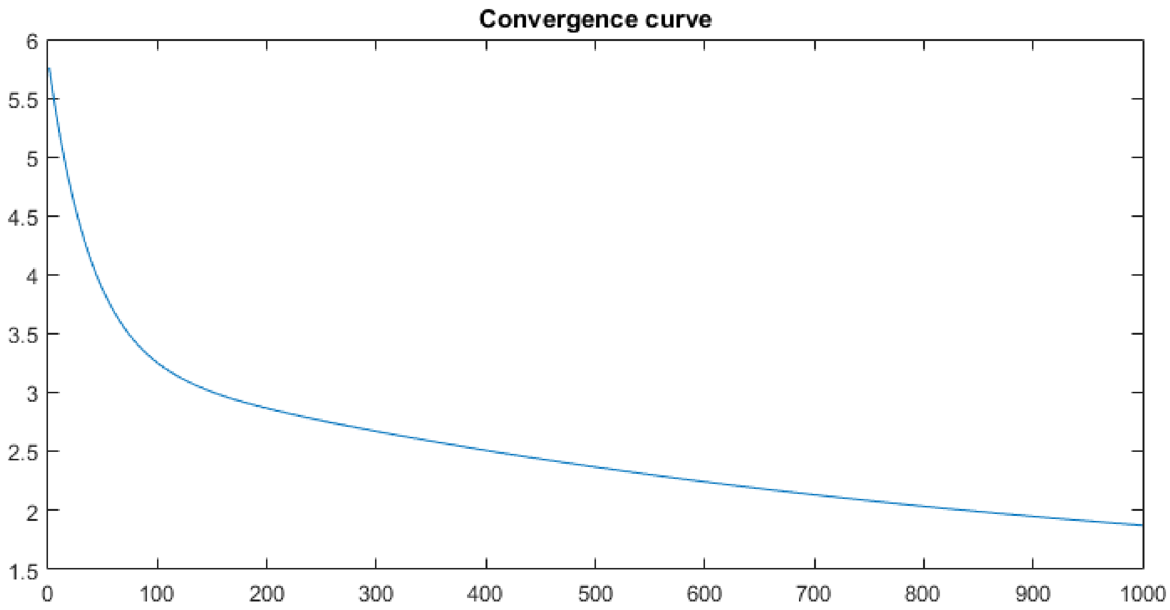

Figure 3: Gradient descent method with learning rate=0.01

4. **Zoom in the flat convergence part. Does it oscillate? Why?**

As we can see, it does not oscillate at all. The reason for this is that the steps always go in the same direction, preventing it from oscillation.

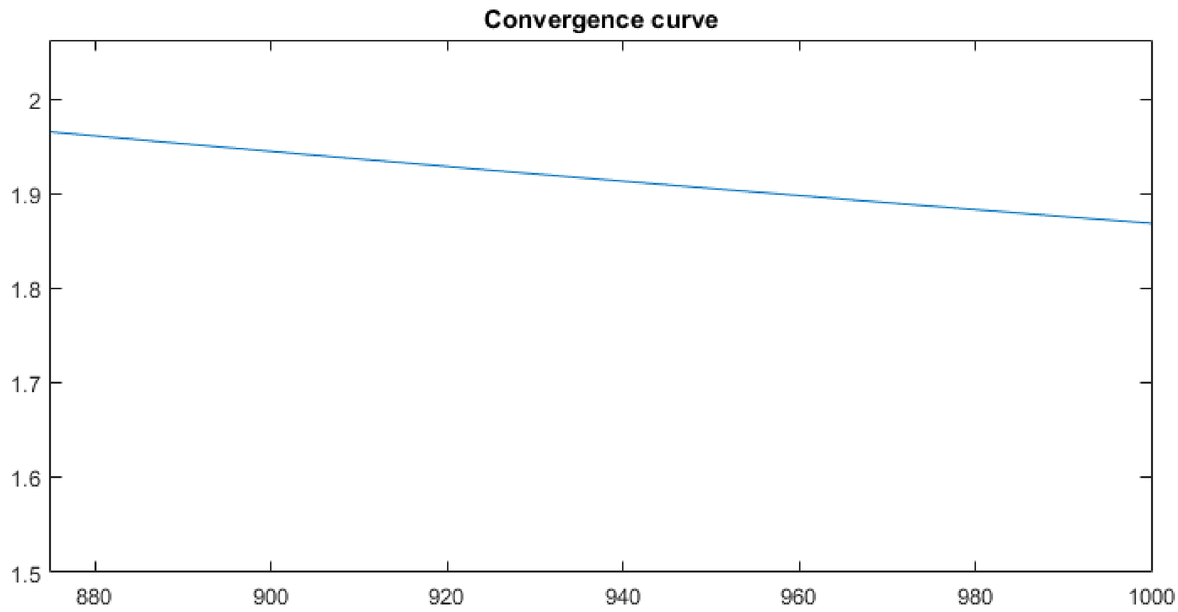

Figure 4: Zoom convergence part

5. **Change the learning rate to 0.1. Plot the convergence curve of the method.**

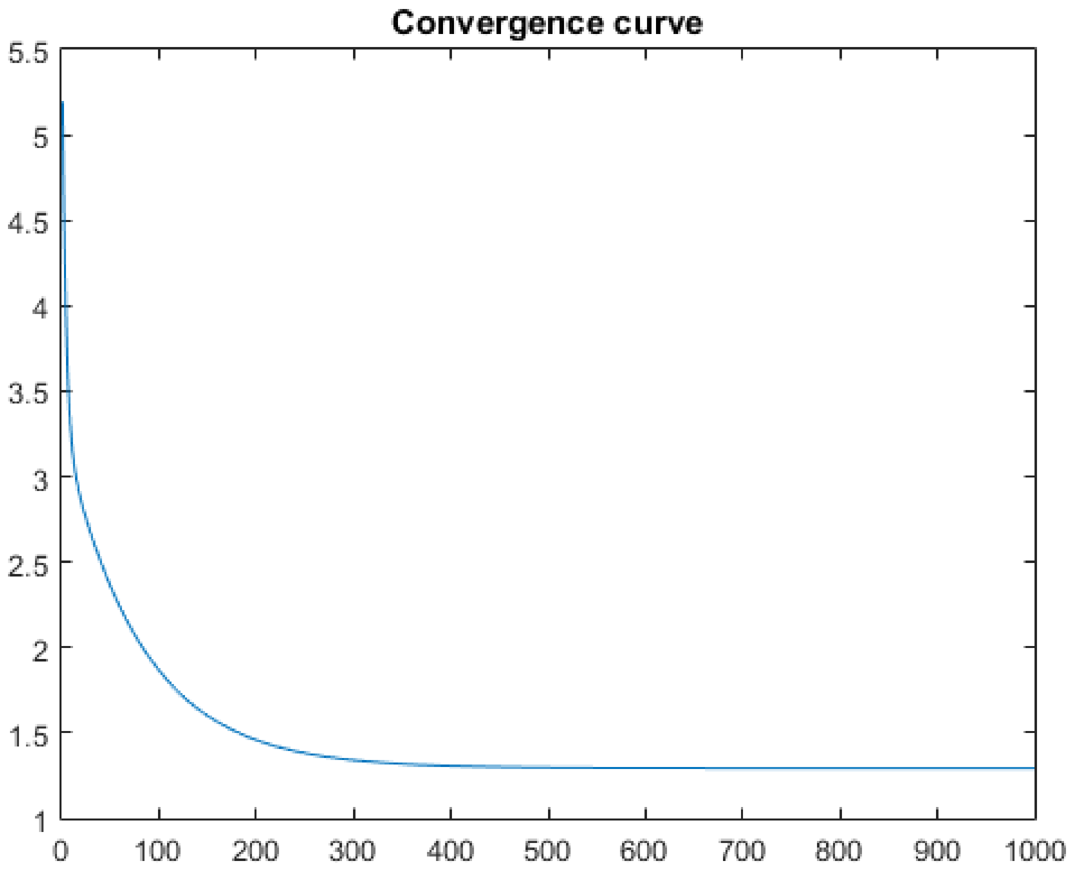

Figure 5: Gradient descent method with learning rate=0.1

We change the descent method to:

$$\Delta x = -\frac{\bigtriangledown f(x))}{\|\bigtriangledown f(x))\|}$$

1. **Which is the optimal value of the linear regression weights using the modified descent method?**

```
modified_w =

  -1.600000000000000
   7.399999999999990
```

   Which means that the function is:

$$f(x) = -1.600000000000000 + 7.399999999999990x$$

2. **Which are the parameters of the descent method used to obtain the optimal value? i.e. learning rate value, number of iterations.**

   Learning rate: 0.1

   Iterations: 100

   Initial weights: $w_0 = 0, w_1 = 0$

3. **Plot the convergence curve of the method.**

   As we can see in this figure, the method converges faster than the one used before. Before we used 1000 iterations, but here we can see that it converges to the minimum a bit before the 100th iteration.

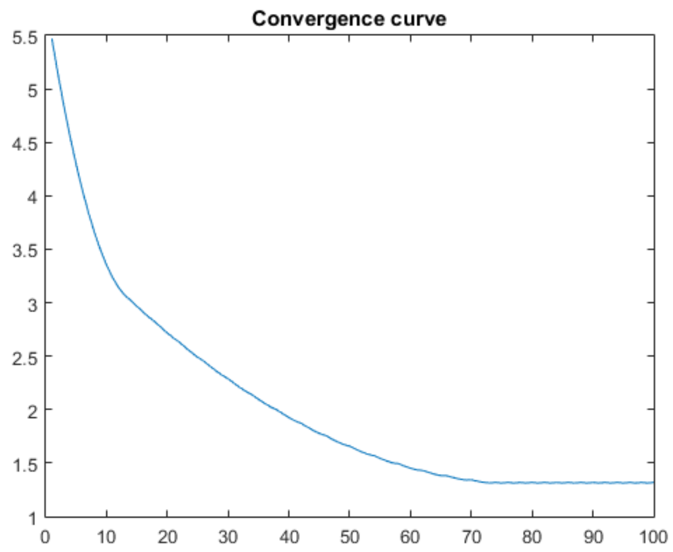

Figure 6: Modified gradient descent method with learning rate=0.1

4. **Zoom in the flat convergence part. Does it oscillate? Why?**

   Here we can appreciate an small oscillation. Here the gradient function is different and now the step is not directly proportional to the gradient, so we can see it oscillate. Anyway, it doesn't deviate much from the minimum, so it is not alarming.

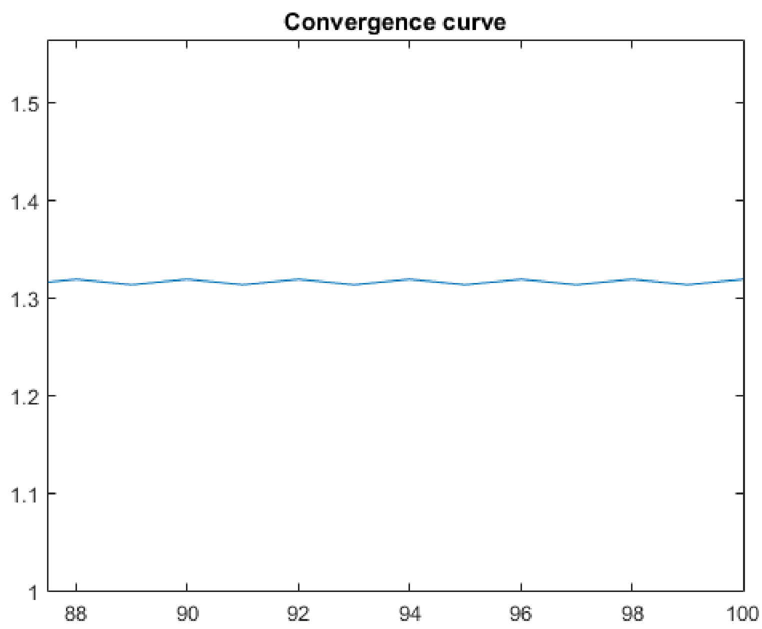

Figure 7: Zoom convergence part

4 A second model

1. **Transform the training set into the set with examples described by z considering p = 3. Apply, the analytic solution code (if properly coded it should work without modifications). Which is the optimal value of the weights?**

```
% polynomial conversion to augmented matrix
polyX = [ones(length(x),1)];
for i = 1 : power
    polyX = [polyX polyX(:, i).*x];
end

weights = inv(polyX'*polyX) * (polyX')*y;
```

```
% weights with polynomial degree 3
weights =

 0.3901
-2.1077
-2.1699
13.6737
```

2. **Plot the data set and the curve just found. Does it fit better the data? Why?**

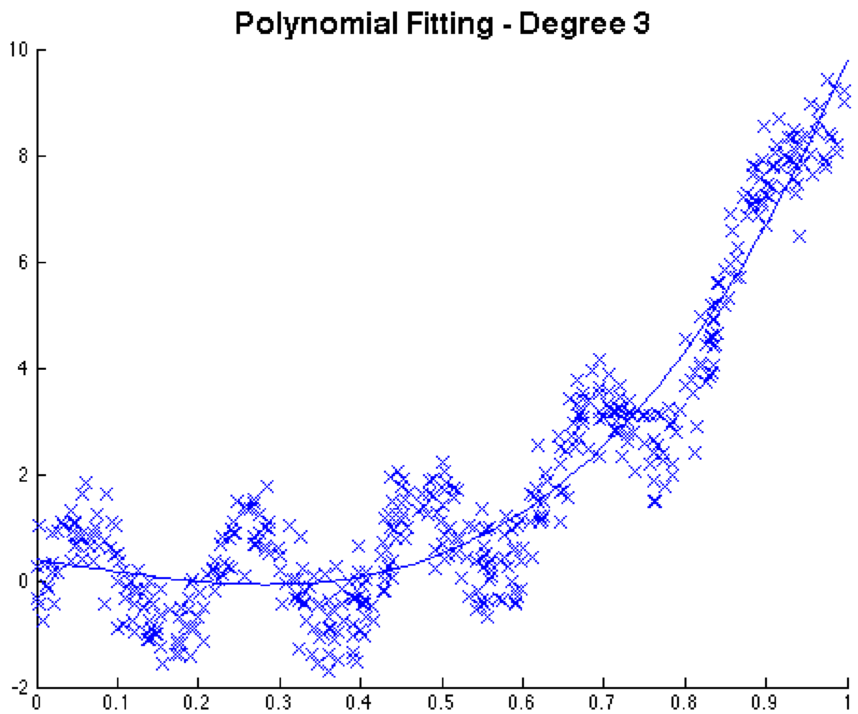

Figure 8: Polynomial regression - Degree 3

The polynomial model with degree 3 (p = 3) fits the data better. This is because, as can be seen from the plot, the data does not display linear characteristics, it rather appears to own some curvatures and inflections. This implies a high order polynomial may fit the data better because a higher order polynomial is more flexible than a linear model and better able to capture the oscillations in the data. The third order polynomial is more flexible and expressive than a linear model in this case.

5 Evaluating a model

1. **Use the first half of the data set for training and the second half for validation.**
2. **Optimize the models (you can use any of the methods implemented before) and plot the validation set and the 6 models plots.**

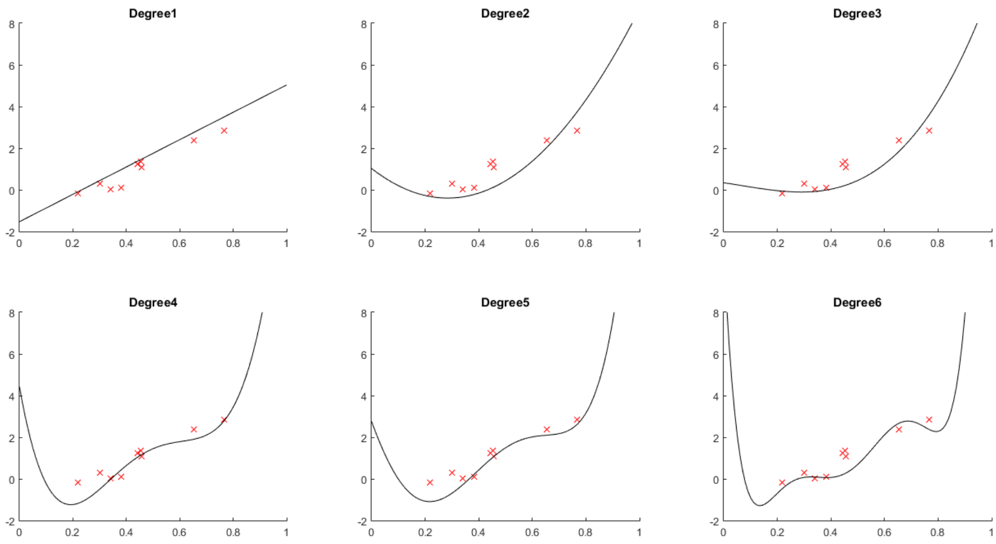

Figure 9: Validation Data with Polynomial Regression

3. **Plot the training set and the 6 models plots.**

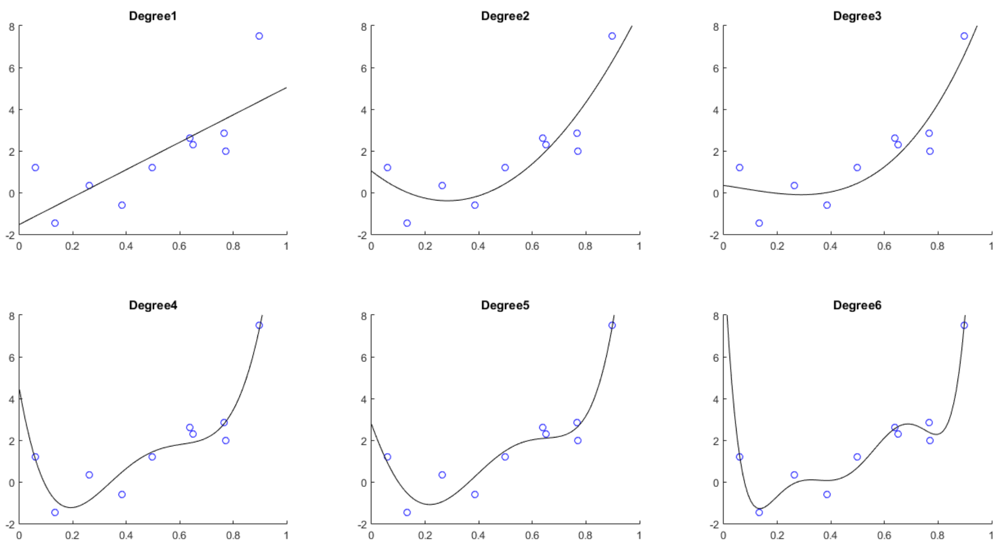

Figure 10: Training Data with Polynomial Regression

4. **Compute the RMS error on the training set and on the validation set. Plot both error curves and describe their behavior.**

   For the training data, we can see that the error strictly decreases as the polynomial degree increases, which can be explained by better fitting at first and overfitting in the end. Overfit-

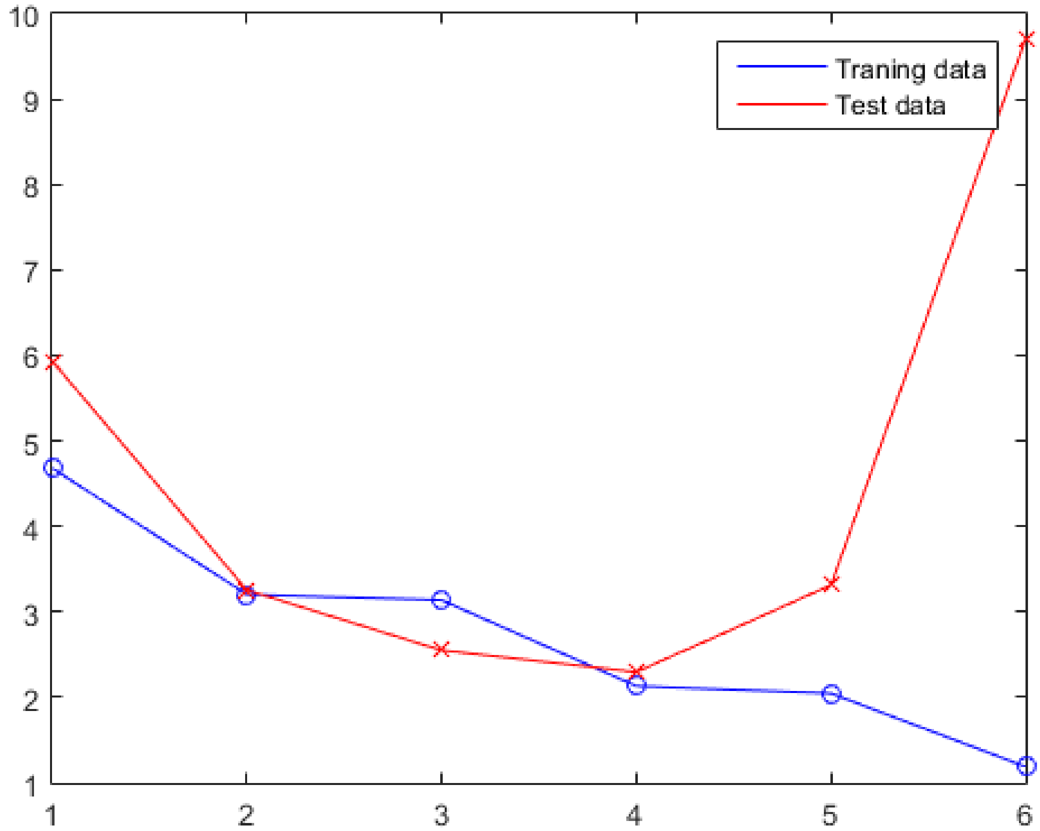

Figure 11: Polynomial degree vs. Error

ting points to a model where the fitted model passes close to or through all of the points in the training set, leading to an error close to zero in the training set.

For the validation data, the error decreases up to the polynomial order of 4, which is due to the increasing flexibility of the model capturing more properties of the data. However, the error starts to increase after this point, which can be explained by the loss of generalizing power of the higher order polynomials caused by the overfitting. i.e. these higher order models are overly tuned to the training data, therefore unable to represent correctly the new data.

5. **Does the selected model agrees with the model that performs the best on the training set? Which one do you think is the optimal choice? Why?**

   The selected model should be the polynomial with degree 4 because it has the lowest error with the validation data, which implies better generalizing power for the unknown data. Even though higher order polynomials result in lower errors with the training data, they cause the validation error to increase, which means worse representative power for the new data. Therefore the selected model does not agree with the model that performs best on the training set. The optimal choice in this case should be the polynomial with degree 4 because it performs best with the validation data and performs good enough with the training set. Still, this generalization may be inaccurate because we have very few data, therefore any new data sample has the power to change the decision in favour of any of the other models.

6 Running the code

In order to run all our code, you can execute the W3 script included in the “Code” folder. This will interactively run all the scripts corresponding to all the exercises, plotting the figures and showing in console the data this assignment asked for.

Please note that after each exercise block, the program will close all the windows and clear the variables and the command line.