CS9542B – Artificial Intelligence II – Assignment #1

Nicky Bayat – 251090689

**Refreshing Mathematics [20 points]**

**Question 1:**

Gradient is the derivation of a multi-variable function f with respect to x, therefore as a general rule the gradient of this function is:

In linear regression gradient can be calculate based on Mean Squared Error as described below and then this derivative is used to update w.

**Question 2:**

We know that:

If we consider C as identity matrix then (1):

Moreover, we know that:

Therefore, we can say (2):

Based on (1) and (2) we can conclude that:

reference: <https://math.stackexchange.com/questions/1808083/gradient-of-traceabatc-w-r-t-a-matrix-a>

**Question 3:**

Hessian matrix of f with respect to w is the first column of H matrix.

**Question 4:**

**Linear and Polynomial Regression [50 points]**

Linear regression line is calculated using **analytical solution** for gradient of loss.

1. The figures of this section will be saved in current directory.

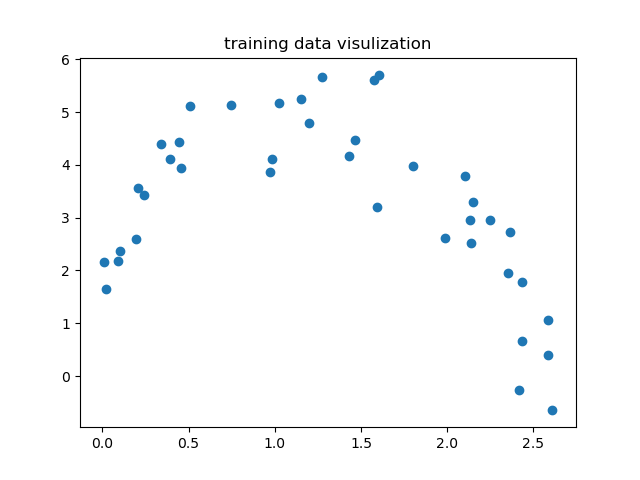
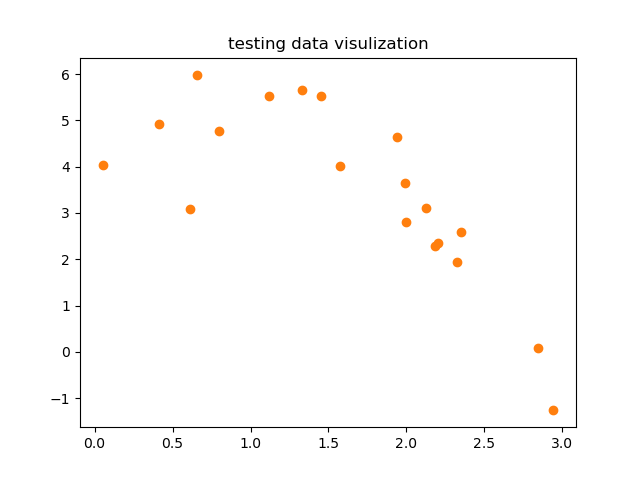


Figure - Testing Set

Figure - Training Set

1. Average Training Error = **4.3232**



Figure - Linear Regression Train Set

1. Average Testing Error = **4.65330**

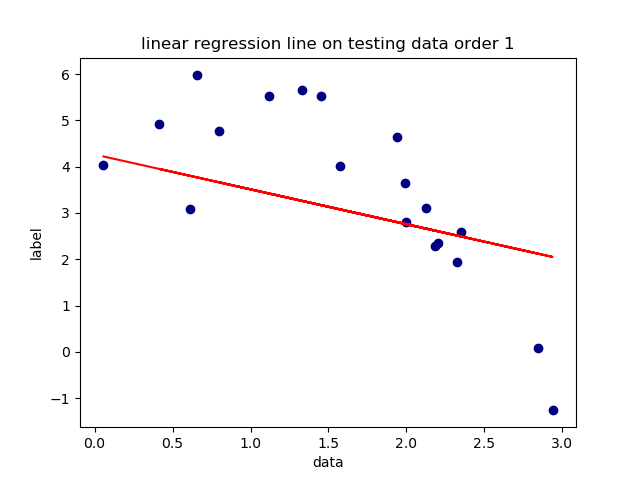


Figure - Linear Regression Test Set

1. 2nd-order polynomial regression; Since error on both train and test decreased, we can deduce that it is a better fit to our data. This can be observed in below figures as well.

Average Training Error = **0.942** Average Testing Error = **1.573**

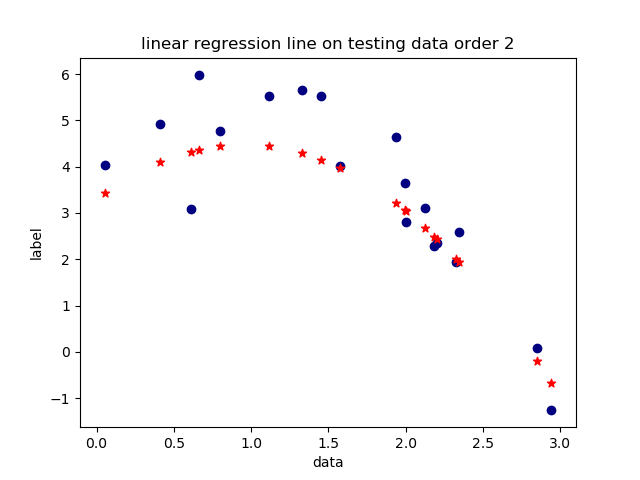


Figure - 2nd-order regression test set

Figure - 2nd-order regression train set

1. 3rd order polynomial regression: It seems to be better fit than 2nd-order regression. Because both errors decreased.

Average Training Error = **0.940** Average Testing Error = **1.495**

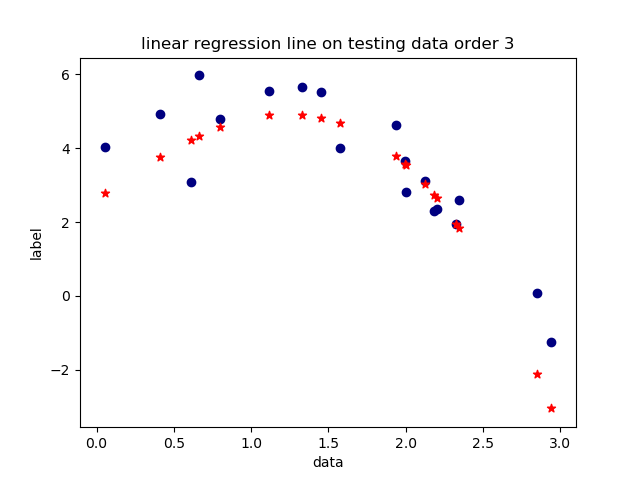


Figure - 3rd-order regression test set

Figure - 3rd order regression train set

1. 4th-order regression: In this case training error decreases while testing error increases significantly which means we have **over-fitting**; therefore, **we can conduct that 3nd order polynomial regression is the best regression choice for our data**.

Average Training Error = **0.866** Average Testing Error = **3.162**

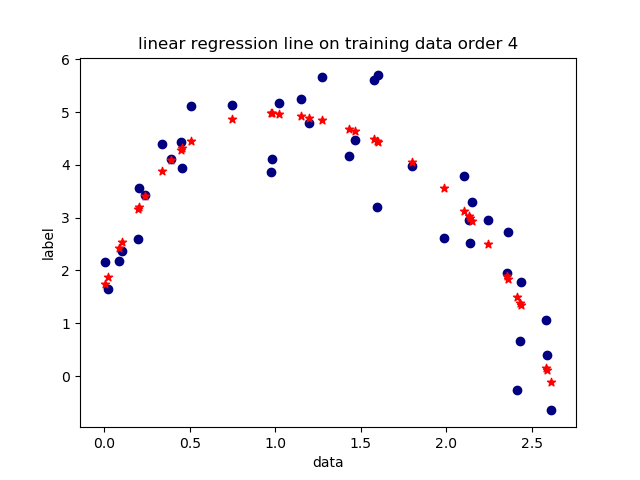
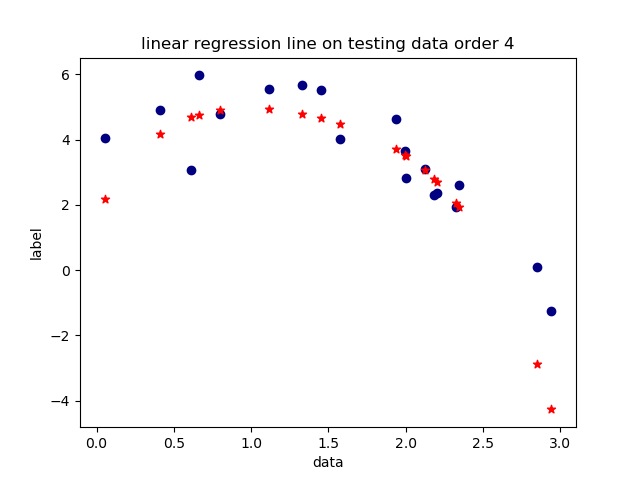


Figure - 4th order regression test set

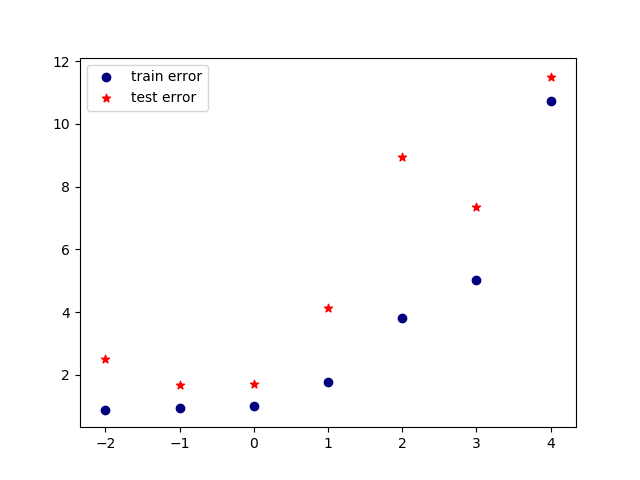
Figure - 4th order regression train set

**Regularization and Cross Validation [30 points]**

1. Here is the formula for regularization and the update function based on that:

Using this formula, I have calculated the loss for . The result for training and testing loss is indicated in the table below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.01 | 0.1 | 1 | 10 | 100 | 1000 | 10000 |
| Training Loss | 0.876 | 0.941 | 1.001 | 1.775 | 3.808 | 5.024 | 10.730 |
| Testing Loss | 2.493 | 1.688 | 1.724 | 4.125 | 8.949 | 7.358 | 11.490 |



Based on this table and figure 11 we can conclude that **lambda 0.1** works best for our data.

Figure - Error based on different values of lambda

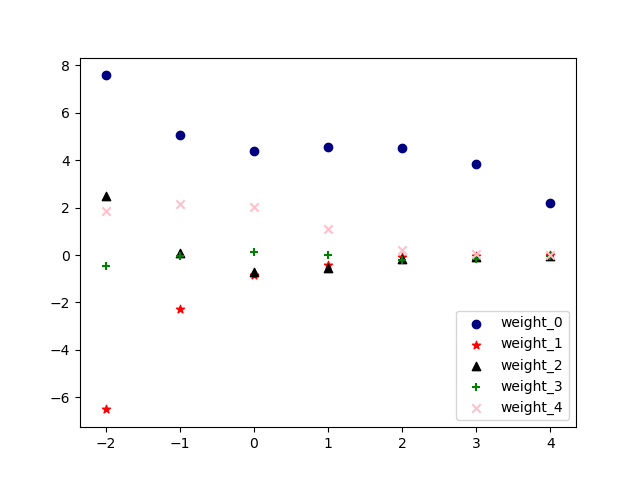
1. In this section, the value of each weight parameter as a function of is plotted.

Figure - weight parameter based on lambda

1. Five-fold cross validation:

The training and validation average error over 5 folds for each lambda is summarized in table below; The best lambda still seems to be **0.1** which is same as (a).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0.01 | 0.1 | 1 | 10 | 100 | 1000 | 10000 |
| Training Loss | 0.849 | 0.924 | 0.998 | 1.955 | 3.909 | 5.275 | 11.324 |
| Validation Loss | 1.150 | 1.144 | 1.168 | 2.141 | 4.090 | 5.462 | 11.795 |

l\_2 regularized 4th\_order polynomial regression curve with lambda=0.1 and test data are represented in the figure below.

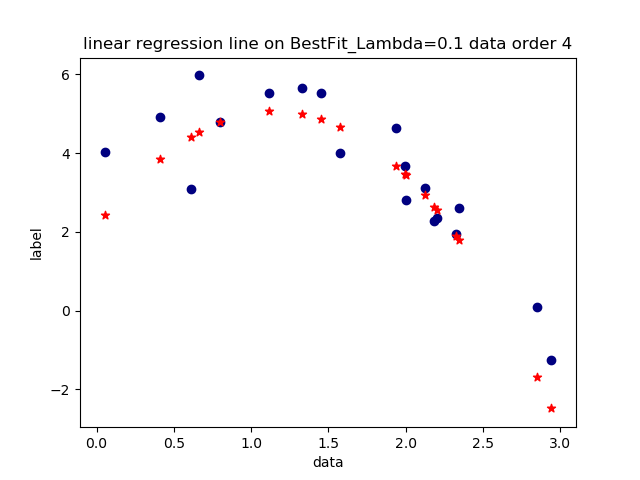


Figure - Best Fit on test data