MACHINE LEARNING

Assignment-1

Parametric Regression

SPRING – 2016

submitted to

Gady Agam

by

Darpan Patel
A20345898

Problem Statement

Here, In this assignment I have implemented parametric regression for Single variable(one – feature) and Multi- Variate (many features – fixed value) regression.

1.) Single Variable regression

- plot the graph of given data and decide the type of regression.
- I have tried linear model for datasets and polynomial models for other datasets.
- I changed degree of polynomial (From linear regression – 1 to polynomial regression – 2,3,4,5…) along with the training data and testing data set according to the K-Fold(cross validation).
- Measure the performance in terms of “testing error” and “training error” .
- Choose the appropriate model according to performance

2.) Multi-Variate Regression

- I have tried data to higher dimensions using generalized linear regression.
- Measure the training error and testing error for different K_fold(cross validation)and choose appropriate model.
- Tried gradient-decent method to compare my result.

Analyzed different experiment's results using different training and testing datasets and compare the results.

Proposed Solution

- I have implemented two separate program for Single Variable linear regression(single_var.py) and Multi-Variate Regression (multivariable.py).
- I have used linear regression for 1st datasets of single variable and polynomial for rest other. Used generalized linear regression for Multi-Variate Regression.

- I have implemented all the function from scratch using some basic python libraries like numpy - for array and matrix functionalities, matplotlib – for plotting 2D graph and sklearn – for just check for correctness of outputs.

Algorithm for Single Variable Linear Regression

- First of load the data into **numpy** array and split it into two different array of Training and Testing. Find the co-efficients for linear regreslsion using degree = 1 on based of training dataset so that model is trained. Predict the value of testing data using trained model. Find the training error and testing error from predicted values. Use K-fold cross validation for all training datasets. default value of K_fold is 10(User can change it.). At the end find the mean-square-error for all the fold and then take average of it for better results and efficiency.

- For other datasets that don't have linear relationship between data then have to implement polynomial regression. I used same algorithm foe different degrees. I tried multiple experiment for different degrees (2,3,4,5…). The function returns mean-square-error for all K-Fold and then find the testing error and training error. According to that result I can choose best model for datasets.

Algorithm for Multi-Variate Linear Regression

- In this regression, I used approach same as previous . Find the required co-efficients from given formula of generalized linear regression. Predict the value for different test data sets and the return the MSE (mean-square-error) for testing and training.

- I used gradient-decent method for iterative solution and compare to my solution in terms of error.

References

- http://nbviewer.jupyter.org/github/justmarkham/DAT4/blob/master/noteboo

ks/08_linear_regression.ipynb

Implementations Details

Design Issue :

- difficult to implement and iterate list for matrix calculation so that I used numpy library for array and matrix manipulation.
- Make dynamic function for linear and poynomial regression.
- Difficult to find polynomial regression for Multi-Variate regression.
- Design issued with Gaussian kernel function for dual regression.

How Problem Solved :

- faced a problem this kind of function from scratch in editor so that I used PyCharm IDE for python coding.
- Comparision between different polynomial degree's output and how to choose best model so I used array functionalities to store intermeiate results and compare them later.

How to run program :

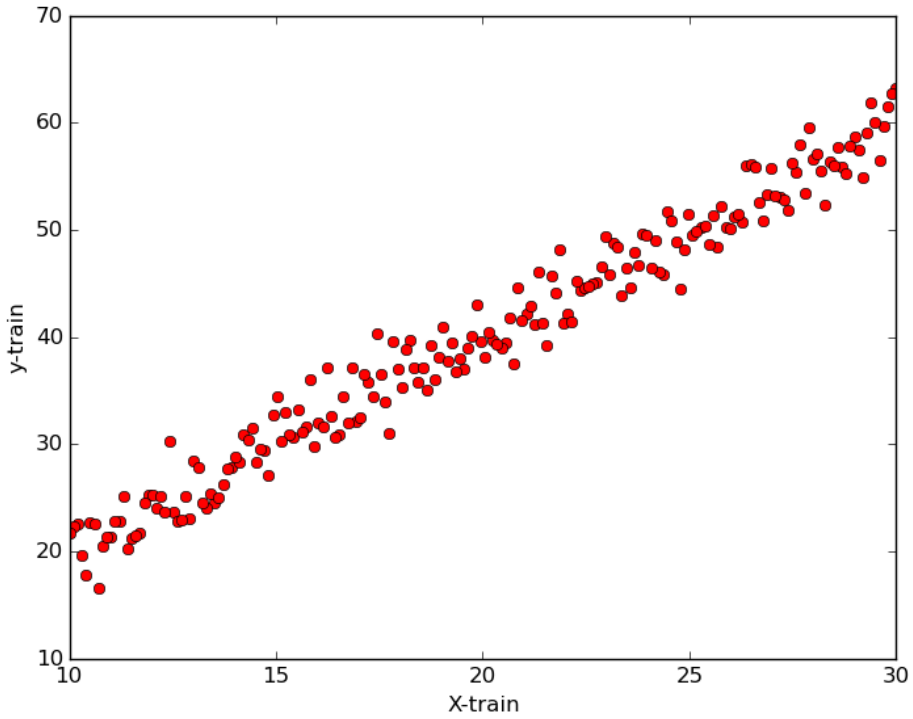
- Run single_var.py
- Enter the value of file number (that you want to test.)
  ex: Enter File Number : 1 (or 2, 3, 4)
- Enter polynomial degree (up to which degree you want to see the comparision value.)
  - ex : Enter polynomial degree : 1 (linear regression) and else for poynomial regression
- For Degree = 1, program works as linear regression and Degree >1, program works as polynomial regression.
- Enter cross validation Fold : 10 (20 or any interger value less than no. of rows in file)

- Run multivariable.py
- Enter the value of file number (that you want to test)
  - ex: Enter File Number = 1 (or 2,3,4)

Results and Discussion

Single Variable Regression

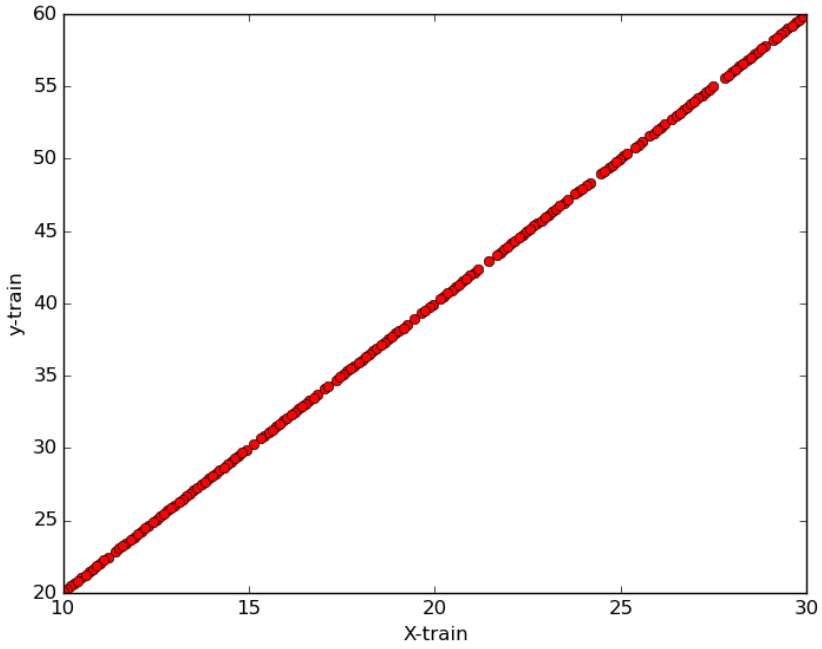
- Loading and plotting of given data of data sets – 1 (svar-set1.dat) and datasets – 2 (svar-set2.dat)



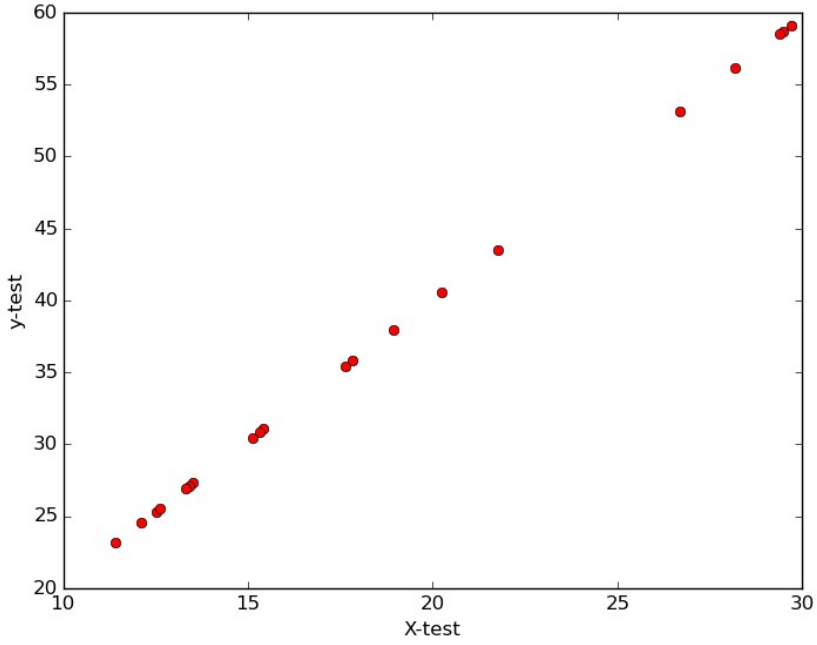
- From the graph of given data we can easily predict that Linear Model is fit for dataset – 1 and polynomial for rest of others. So other datasets needs some higher level polynomial regression

Linear regression Results

- Dataset – 1 , Polynomial – 1 , K-fold =10



(Up) Training values and Tesing values (Down)

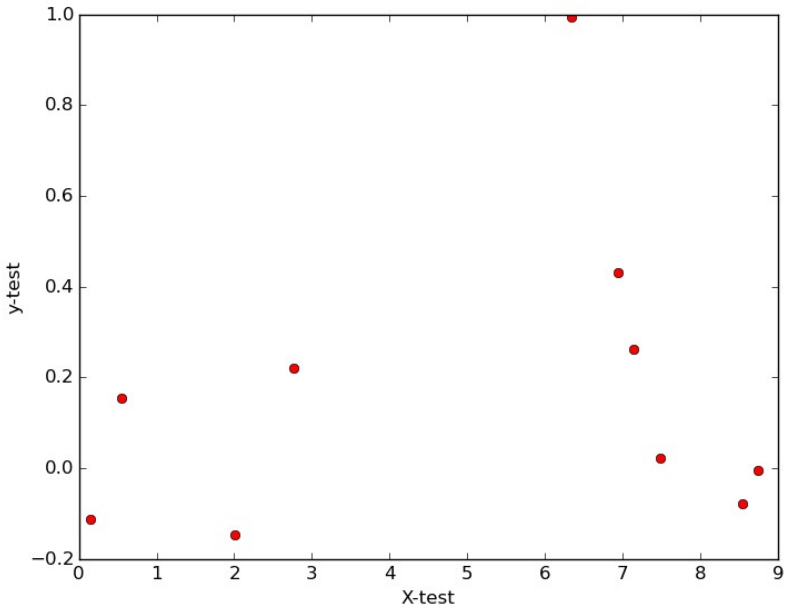


Experimental Error of Linear and Polynomial

(D – Dataset No. , P – Polynomial Degree , K – K-fold value)

Types of Experiment	Training error (MSE)	Testing error (MSE)
D -1, P- 1, K – 10	4.22549268	4.36875256
D – 3, P – 6, K - 20	0.06301617	0.06885697
D – 2, P – 3, K -15	0.02028562	0.02162703

Test result Graph for Experiment -2



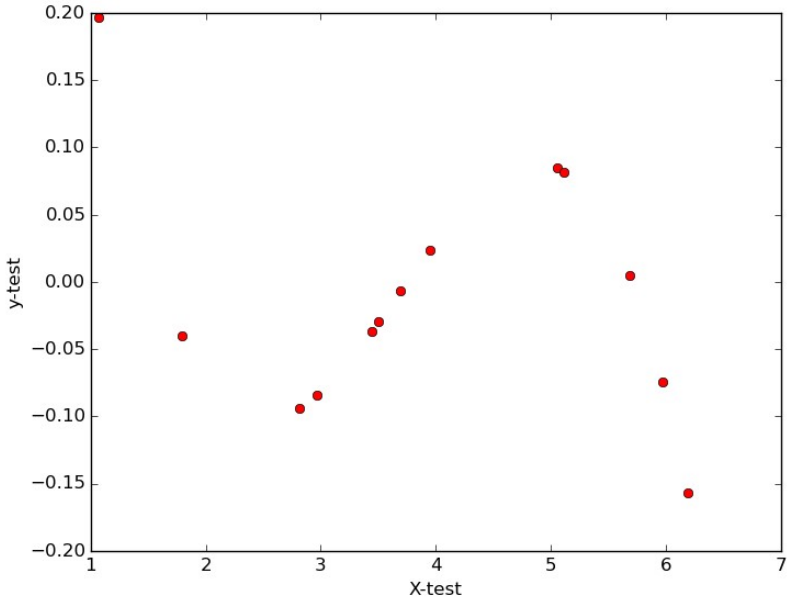
==========>

Dataset – 3 ,
Polynomial Degree – 6
K-fold =20

Testing Results
Best Model = 6 degree

==========>

Dataset – 2
Polynomial Degree – 3
K-fold =15
Best Model = 3 degree



- Training error is always less than Testing Error.
- Result is become more accurate for higher degree of polynomial
- According to result we can decide the model in which testing error is minimum.

Comparision to Ready made Python Library

- I used ready made python library named `sklearn import linear_model”`
- It gives me accurate co-efficients for my X-test data and give the MSE for testing error **4.2323123** and training error **4.1213543**
- It is as good as my results for linear regression of (D-1, P -1, K – 10) (refer table)

Reducing The Amount of training dataset

When we reduce the amount of training data, the training error rate usually increase as the model became more poor. Here is reducing training data from 90% to 50%

As we reduce the amount of training data then Error will increase.

Amount of Training data	Testing Error
90% K-Fold =10	4.36875256
80 % K-Fold = 5	4.49154502
50% K-Fold = 2	4.59320829

Multi-Variate Regression

- In a Mulivariate Regression I have calculated the test and train error for higher dimensions from 2 to 6 (file given by professor). But we can do calculate test error for any number of dimensions using my program.
- Results for each data set of training and testing error in below table. K_fold = 10

Data Sets	Training error (MSE)	Testing error (MSE)
D -1	5.1233664233	5.7129706868
D – 2	0.0314874998348	0.0314014795097
D – 3	12.1254353	12.5393123123
D – 4	0.00389072530841	0.00418917353293

- In this case training error is also lesser than testing error.
- Different errors for different datasets given by proffesor.

Iterative Solution

- I used gradient descent as a way to implement the iterative solution.

- Using this method I just calculate value of co-efficients and compare it to my predicted co-efficients which is below.

Predicted theta :

Predicted Theta (Co-efficients)	Gradient Decent Theta (Co-efficents)
1.00035924	0.99783983
0.99871656	0.99911763
0.99454442	0.99689591

- We noticed that it gives more accutrate theta value for 10 iterations in gradient-decent function. In gradient-decent we can reach a global optimal if we choose the appropriate parameters

Gaussian Kernel Methods

- I tried to find Gussian kernael method but there may be some problem to find it so I tried it with internal library named “`sklearn` **`import gaussian_process”`**
- I tried this libraries but some I could't get correct answer for it.

References

- http://docs.scipy.org/doc/numpy/reference/routines.sort.html
- https://gist.github.com/stober/4964727