CS 584 – Machine Learning

Assignment 1

Guru Prasad Natarajan(A20344932)

1. Problem statement:

The objective of the assignment is to implement the techniques used in parametric regression and to test the performance of the implementation.

2. Proposed solution:

Parametric regression techniques are implemented programmatically and comparing the results obtained with the in-built python functions to assert the correctness of the developed algorithm. The performance of the implemented models using mean-squared error technique. Parameters are calculated using explicit and iterative solution.

The algorithm implemented has got the following steps:

Transform the given X into Z based on the degree supplied.
Split the data into training and testing set.
Using 10-fold cross validation find the value of theta for each and every step.
With the theta computed, predict the values of x_train and x_test.
Using mean-squared error calculate the training and testing error.

Algorithm to calculate theta explicitly:

With the obtained transformed matrix, calculate its pseudo inverse and dot product it with the true values to get theta.

3. Implementation details:

The coding part is done in ipython note book. The program is better modularized for easy understanding and reusability.

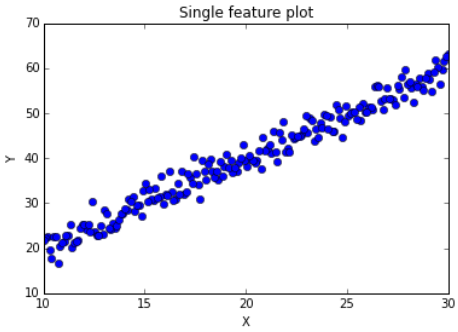
Instruction details:

The file can be run on any system which has ipython installed. No special instructions required.

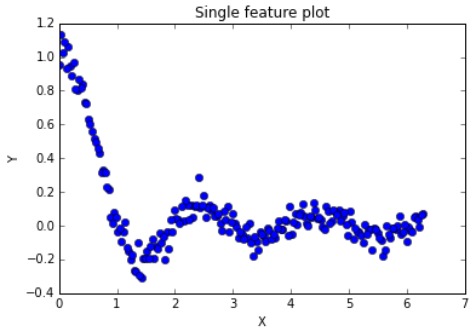
4. Results and discussion:

In order to get an idea on the complexity of the data, the data is plotted and below are the results:

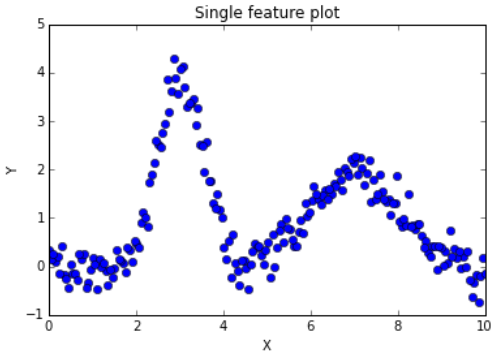
Dataset 1: svar-set1



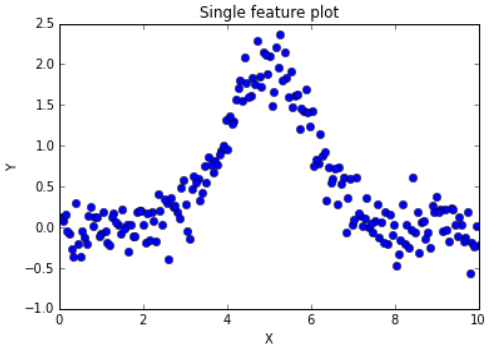
Dataset2:svar-set2



Dataset 4: svar-set4



Dataset3: svar-set3



b) Testing and training error are computer using mean squared error technique and the mode is plotted on top of the data.

Dataset 1: svar-set1

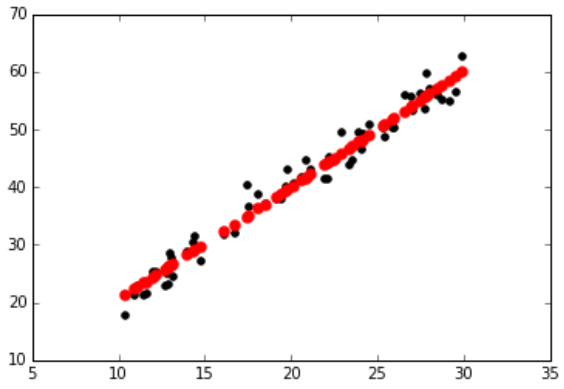
MSE Values		
Polynomial	Train	Test
1	4.225493	4.368753

The training error is less than the training error which says that most of the error is captured during testing and not during training. It is normal to have less training error when the data is split into multiple folds.

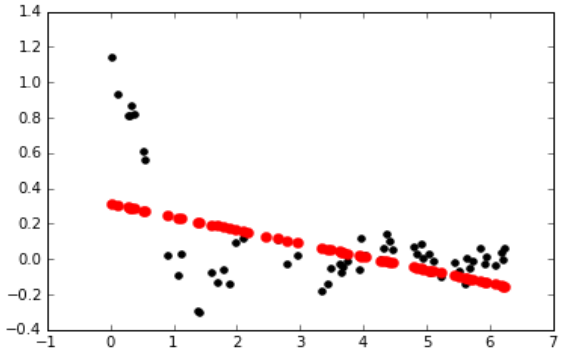
Plot on test data:

The data is split into 70:30 as training and testing data and the graph is plotted.

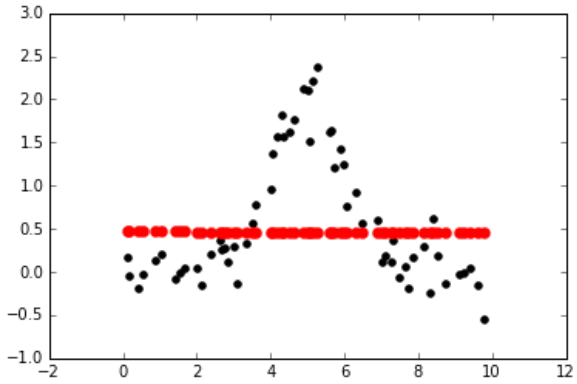
Svar-set1



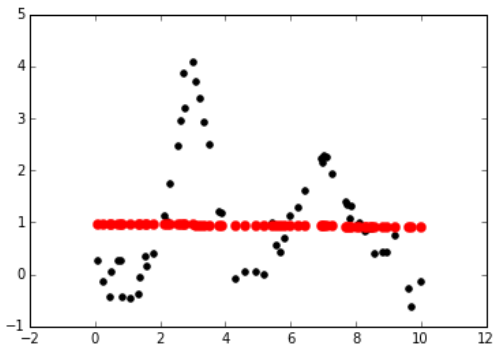
Svar-set2



Svar-set3



Svar-set4



From the above plots, we can conclude that linear model fits only the first data and we have to go for polynomial models for the remaining data sets.

c) By comparing the results of the implemented model, with the readymade inbuilt function. It is found that both the models yielded the same results which prove the correctness of the algorithm.

e.g., Both the models returned the same mean squared error and the same parameters. A sample is provided below:

Dataset: svar-set4(order 3)

MSE obtained through implemented algorithm

MSE Values		
Polynomial	Train	Test
3	0.918144	0.938203

MSE obtained through inbuilt algorithm:

MSE Values		
Polynomial	Train	Test
3	0.918143935202	0.938203377002

Moreover the intercepts and the co-efficient obtained through the in-built function and the theta values of the linear model computed was the same.

Linear model is fit to the data. Training and testing error is calculated using mean squared error and in order to test the performance of the model, 10 fold cross-validation is used.

Dataset 1: svar-set1

MSE Values		
Polynomial	Train	Test
1	4.225493	4.368753
2	4.221782	4.430751
3	4.148673	4.449766
4	4.114483	4.477184

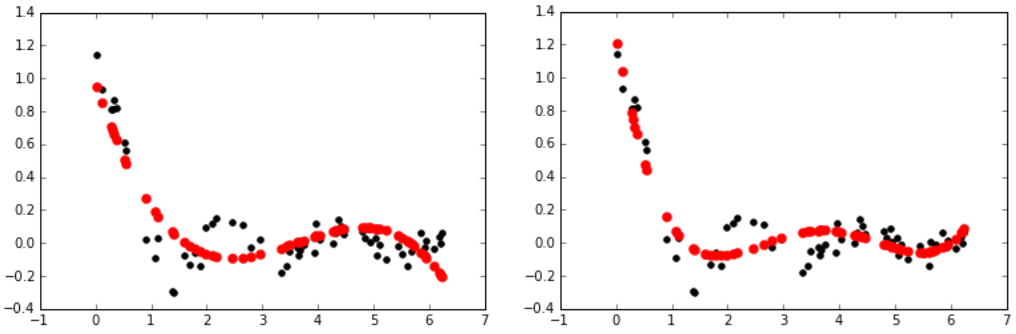
Since from the above, for data set 1 a straight line will be well fitting the data on either side of the line.

From the errors obtained, as the polynomial is increase the error on testing error seems to be increasing. **So, linear model is a better fit**

Dataset 2: svar-set2

MSE Values		
Polynomial	Train	Test
1	0.059487	0.061201
2	0.038604	0.040108
3	0.020292	0.021370
4	0.011462	0.012200
5	0.011042	0.011906
6	0.010855	0.011881

Deg 3

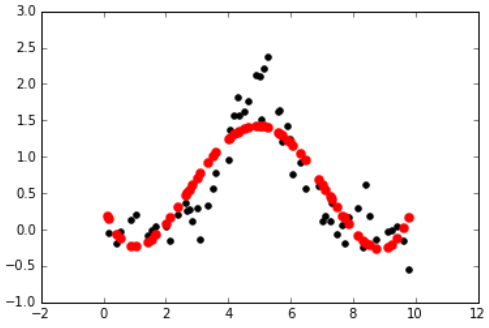


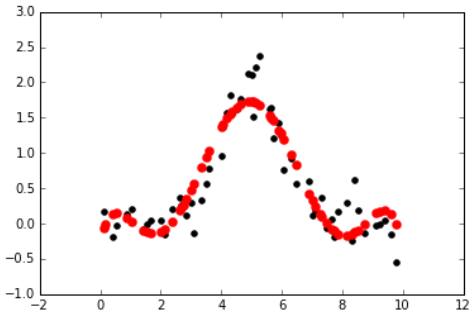
From the above results, polynomial of degree 4 or higher offer fits the data. So model with degree 3 is the optimal fit.

Dataset 3: svar-set 3.dat

MSE Values		
Polynomial	Train	Test
1	0.498417	0.504339
2	0.252896	0.260586
3	0.252810	0.262183
4	0.126369	0.133380
5	0.125981	0.133889
6	0.062818	0.069861

Deg 4

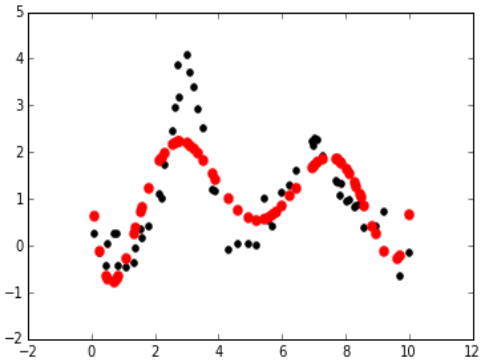




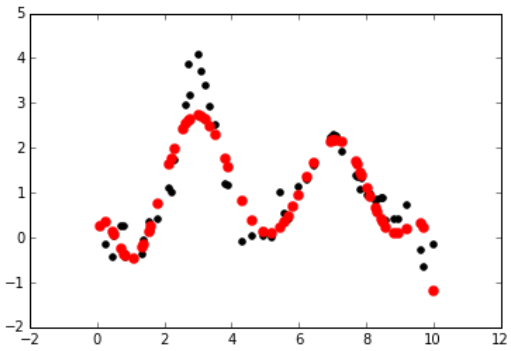
From the above results, polynomial of degree 4 or higher offer fits the data. So model with degree 3 is the optimal fit.

Dataset 4: svar-set4

MSE Values		
Polynomial	Train	Test
1	1.200203	1.212287
2	0.927843	0.940830
3	0.918144	0.938203
4	0.848577	0.876431
5	0.802041	0.840307
6	0.449782	0.478090



Degree 6



Degree 7

From the above results, polynomial of degree 7 or higher offer fits the data. So model with degree 6 is the optimal fit.

By varying the amount of train data by introducing higher percentage split, mean errors vary as below:

Linear model

(Train:Test)	MSE Values	
Split(%)	Train	Test
70:30	4.24816637	4.1981058
80:20	4.12436545	4.71937777
25:75	3.22778353	5.11291284

Polynomial model

(Train:Test)		MSE Values	
Split(%)	Polynomial	Train	Test
25:75	3	1.05578422	0.90047475
	4	1.01782058	0.8251131
	5	0.91539946	0.80263854
70:30	3	0.88796329	1.01484804
	4	0.82347664	0.93090805
	5	0.7899927	0.85956034
80:20	3	0.87354475	1.11802253
	4	0.80294333	1.04728414
	5	0.76366791	0.97788023

From the above results we can conclude that both the amount of training data and the polynomial plays a major role in deciding the model to be used.

More the training data better the model will be, from the above error decreases with more training data and as the polynomial increases which tend to better fit the data.

Section 2

a) The given data is with multiple features are loaded and mapped to higher dimensional features using combination of features.

For e.g., Let X has two features x_1 and x_2. So, the combination of features would be x_1, x_2, $x_1.x_2$, x_1^2 , x_2^2.

b) Linear regression is done in the higher dimensional space and below is the testing errors calculated using 10 fold cross validation

Dataset 1: mvar-set1

MSE Values	
Polynomial	Test
2	0.259617
3	0.260660
4	0.260305
5	0.261528
6	0.261960

From the above results, testing error increases as we increase the degree of the polynomials. It would be better to choose degree 4 as it has got sufficient number of combinations of the feature vectors to produce a best fit.

The data set is partitioned into various subsets of train and test data and the below result is obtained.

(Train:Test)		MSE Values	
Split(%)	Polynomial	Train	Test
70:30	3	0.259756404814	0.253816418168
	4	0.258363502108	0.253983392208
	5	0.257756893012	0.255624728328

80:20	3	0.254190760043	0.273010886094
	4	0.252741035187	0.273413371962
	5	0.252205381503	0.275372137249

c) Theta values are observed using both explicit solution and iterative solution. Gradient descent is the iterative solution used. Below are the results.

Explicit Solution:

Dataset: mvar1-set1: Degree 2

```
Solution to the regression problem using iterative solution:
[ 1.02230218  0.99752589  0.99048461 -0.01260975 -0.00847393 -0.00643214]
```

Iterative solution:

```
Solution to the regression problem using iterative solution:
[ 1.02230218  0.99752589  0.99048461 -0.01260975 -0.00847393 -0.00643214]
```

From the above results, it is found that both the values are the same. So, it is going to have the same result as the previous implementation.

d) The dual regression problem is solved using the “Kernel trick” and Gaussian Kernel is used. In order to predict the values, Nadaraya –Watson kernel estimator is used which is below,

$$\widehat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(x - x_i) y_i}{\sum_{i=1}^{n} K_h(x - x_i)}$$

Based on the values of sigma the error values changes which can be used in selecting a better model. When it comes to calculation, it was found that dual problem takes more time to solve because of more number of computations involved. However, by changing the value of sigma we can get a model which has less error which makes it more accurate than the primal.