Parametric Regression

Karthik Bangalore Mani
Department Of Computer Science
Illinois Institute of Technology

February 21, 2016

Abstract

The goal of this assignment is to fit the best Model for Single Variate and Multi-variate Datasets. The models performance are tested using 10-fold cross validation

Problem Statement

Given a Dataset 'D' , fit a Model 'M' in such a way that it is neither overfit nor underfit. For Polynomial Models, the best order of the polynomial is that , which has the least Root Mean square error.Care should be taken such that the datasets are not over-fitted.

Proposed Solution

For the Explicit solution, start with a polynomial order *d* , perform 10-Fold cross validation, compute the Root Mean Square error of the Model . Repeat the above step by increasing *d* to *d+1*, until a minimum value of RMS error is obtained. The model with minimum RMS is chosen as the best model.

For the Numerical solution, Gradient Descent has been chosen. Theta_j is recursively computer until, J(theta) becomes approximately equal to J(theta – 1). The algorithm for Gradient Descent is :

$$\text{Repeat until convergence } \{$$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\}$$

*Pic from coursera.org

Implementation Details

1. **Program Design Issues**
   The size of Z matrix grows exponentially as the degree of the polynomial is increased for multi-variate data sets. This especially affects processing of large data-sets such mvar-set3.dat which is of size **10MB.** Inverting the whole matrix in-memory, and computing the Z-matrix for this data-set which has 5 features makes it hard to go beyond a polynomial degree 5.

2. **Problems faced**
   Problems were faced in identifying the learning rate in Gradient descent and identifying the best order of polynomials. This was mainly due to bugs in the code. Once it was fixed, no issues were found.

3. **Instructions to use the program**

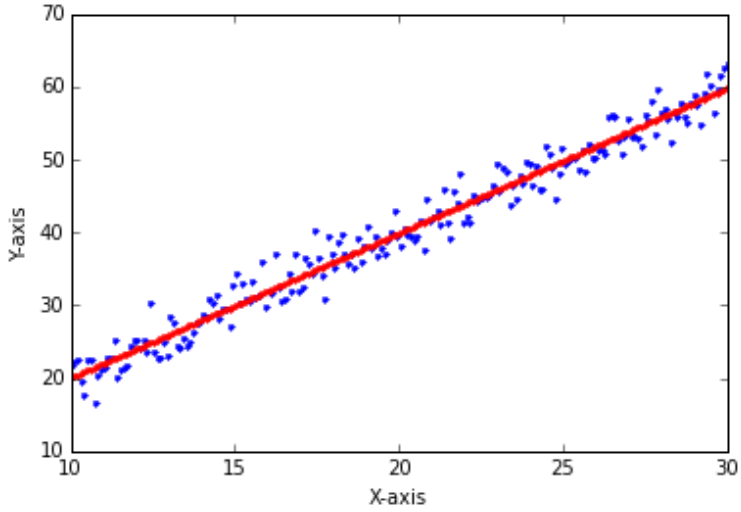
   Open the Regression.ipynb file in iPython notebook, and execute each cell to get the desired output. The datasets must be placed in the same folder as that of the Regression.ipynb file.

Results and discussion

1. **Uni-variate Regression :**

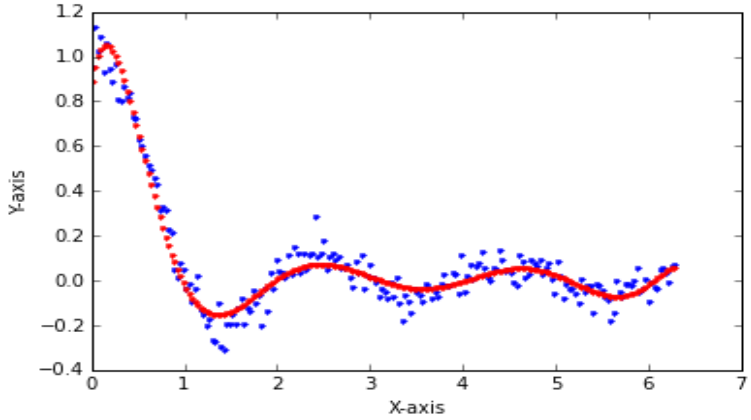
   svar-set1.dat was fit as shown below : It's Training and Testing RMS errors were 0.00360451045653 and 0.00391965983096 respectively. It was verified with scikit learn's inbuilt function, and found to be the same.

   The theta vector was found to be [0.26120329 1.98610257]



   svar-set2.dat was fit as shown below :

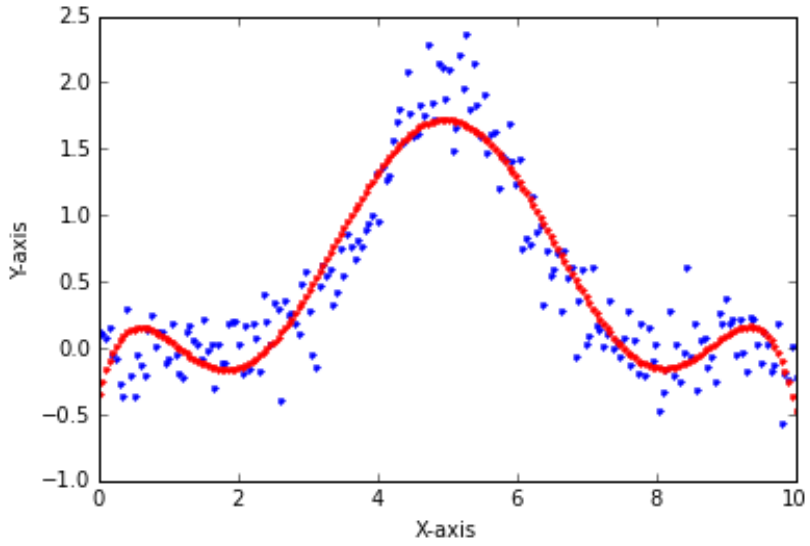
   The theta vector was found to be [8.91700506e-01 2.29959588e+00 -9.78716952e+00 1.10430917e+01 -5.97836005e+00 1.77747676e+00 -2.97140213e-01 2.62041644e-02 -9.48726519e-04]



svar-set3.dat was fit as shown below :

The theta vector was found to be :

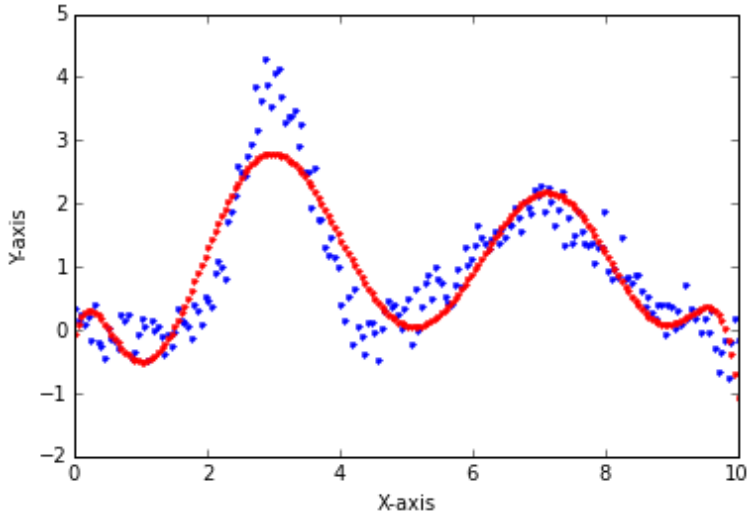
[-3.38612474e-01 2.01456984e+00 -2.66391030e+00 1.29405906e+00
 -2.66026712e-01 2.42605377e-02 -8.13729024e-04]



svar-set4.dat was fit as shown below :

The theta vector was found to be :

[-5.59856523e-02 3.84188757e+00 -1.22744855e+01 1.23206342e+01
 -5.40239384e+00 1.21766630e+00 -1.47822306e-01 9.20058377e-03
 -2.30588252e-04]



2. Multi-variate Regression

For **mvar-set1.dat ,** below table corresponds to Degree and its corresponding RMS test error.

```
2 --> [ 6910.17513645]
3 --> [ 8141.42098504]
4 --> [ 8311.45972367]
5 --> [ 8494.36361514]
6 --> [ 7729.23896228]
7 --> [ 8780.13261025]
8 --> [ 9178.35405161]
9 --> [ 9140.27554912]
```

Since **Degree 2** has the least error rate, it was chosen as the best order of the polynomial.

For **mvar-set2.dat ,** below table corresponds to Degree and its corresponding RMS test error.

```
2 --> [ 42128.44898433]
3 --> [ 24332.20361968]
4 --> [ 27266.76737233]
5 --> [ 9145.27889403]
6 --> [ 8466.82895311]
7 --> [ 2804.2126203]
8 --> [ 3102.39305476]
9 --> [ 4203.36036869]
```

Since **Degree 7** has the least error rate, it was chosen as the best order of the polynomial.

For **mvar-set3.dat ,** below table corresponds to Degree and its corresponding RMS test error.

```
2 --> [ 29887.97639526]
3 --> [ 27071.0971735]
4 --> [ 30204.57531415]
```

Since **Degree 3** has the least error rate, it was chosen as the best order of the polynomial.

For **mvar-set4.dat ,** below table corresponds to Degree and its corresponding RMS test error.

```
2 --> [ 7890162.15180832]
3 --> [ 8072458.32487905]
4 --> [ 28800061.89242591]
```

Since **Degree 3** has the least error rate, it was chosen as the best order of the polynomial.

3. **Gradient Descent**

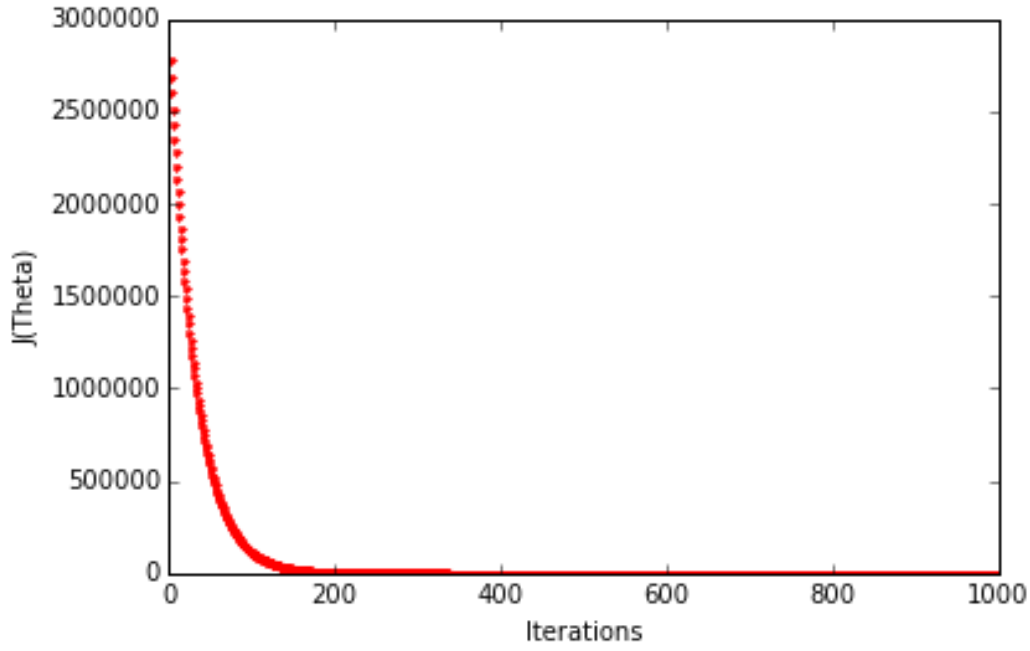
Gradient Descent were performed on 2 datasets : multi_var_1.dat (3 features) and multi_var_2.dat (4 features).

The Gradient Descent was found to converge at the below theta vectors :

multi_var_1.dat : [4.73663699 3.99999972 3.26336244]

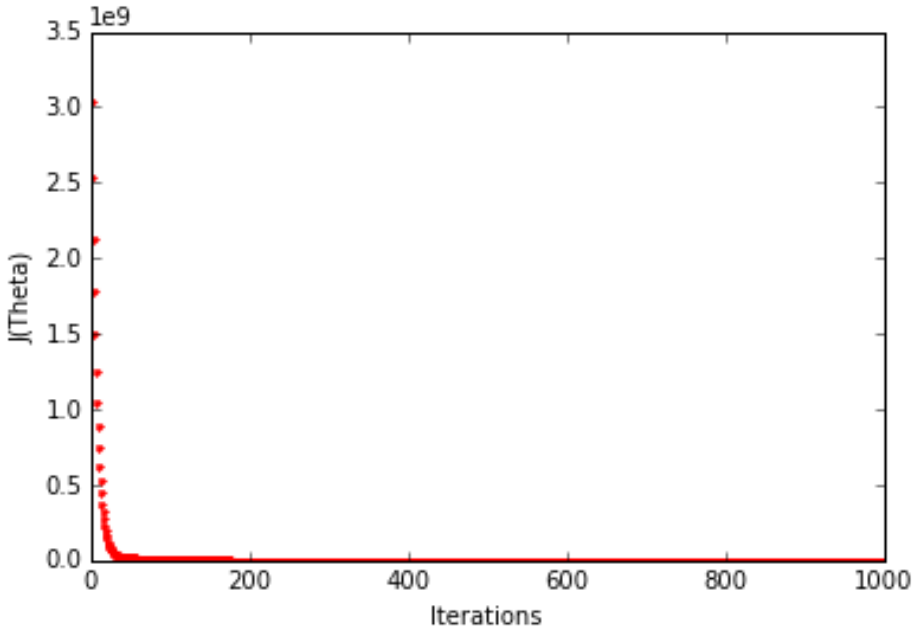
multi_var_2.dat : [4.99756564 2.00486872 3.00243436 5.99513128]

Below is the Plot of J(theta) v/s the number of iterations for the above 2 datasets :



At the end of 1000 iterations, the parameters or the theta vector was found to be :

[4.73663699 3.99999972 3.26336244]



At the end of 1000 iterations, the parameters or the theta vector was found to be :
[4.99756564 2.00486872 3.00243436 5.99513128]

Weakness : For very huge datasets with sizes >10 MB and higher order polynomials >5, Computation of the theta vector requires lot of time.

References

1. **Coursera.org**
2. **Stackoverflow.com**