Soft Computing:ISCO630E

Report

Assignment 2.2

Anant Chaturvedi(IIT2016506)

Question Description

1. Use HP data to implement LWR .You may take neighbouring batch size of data.Discuss what will happen when tau is very small.

Introduction

We are given a dataset comprising of housing prices with various features(continuous as well as categorical).We will be given a query point for which we will locally fit the data to predict the house price for the given query point.

Concepts Used

Hypothesis Function

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x, \qquad X = \begin{bmatrix} — (x^{(1)})^T — \\ — (x^{(2)})^T — \\ \vdots \\ — (x^{(m)})^T — \end{bmatrix}. \qquad \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}.$$

The above formula approximates y as a linear function of x, called hypothesis. The hypothesis considers both Θ and x as (n + 1) size vectors.

(1) X = m X (n + 1) matrix, containing the training samples features in the rows, where each $x^{(i)}$ is a (n + 1) X 1 matrix.

(2) m = number of training samples

(3) Θ_i = the parameters

(4) x_0 = 1 for every sample.

(5) n = features count, not including x_0 for every training sample

(6) Y contains all the m target values from the training samples, so it's m X 1 matrix

Cost Function

Replace

$$\sum_i w^{(i)}(y^{(i)} - \theta^T x^{(i)})^2.$$

In place of

$$(y^{(i)} - \theta^T x^{(i)})^2,$$

In the Original Regularized Cost Function Where

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

The smoothing parameter, τ , is the fraction of the total number *n* of data points that are used in each local fit. The subset of data used in each weighted least squares fit thus comprises the

n* τ points.In our Question we were supposed to fit around 20 points and there are 546 samples in our dataset so τ =float(20/546).

Normal Equation(Regularized)

$$\theta = (X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix})^{-1} X^T \vec{y}$$

Gradient Descent(Regularized)

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$$

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \qquad \text{for } j = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m}\theta_j \qquad \text{for } j \geq 1$$

α = Learning rate

The convergence arises when $J_t(\Theta) - J_{t+1}(\Theta) <= \text{ERROR\_THRESHOLD}$

Since the cost function is convex in nature, the local minima obtained for the cost function would be the global minima.

RESULTS

The convergence appeared to be faster for

α =10000000000

Error_Threshold=1e-30

λ =1e-22

The Original Value at [1, 4000,2 ,1 ,1 ,1 ,1 ,0 ,0 ,0 ,0 ,0] is 39000

For the approach of Gradient Descent it took 54 iterations to arrive at a final cost of 19.06

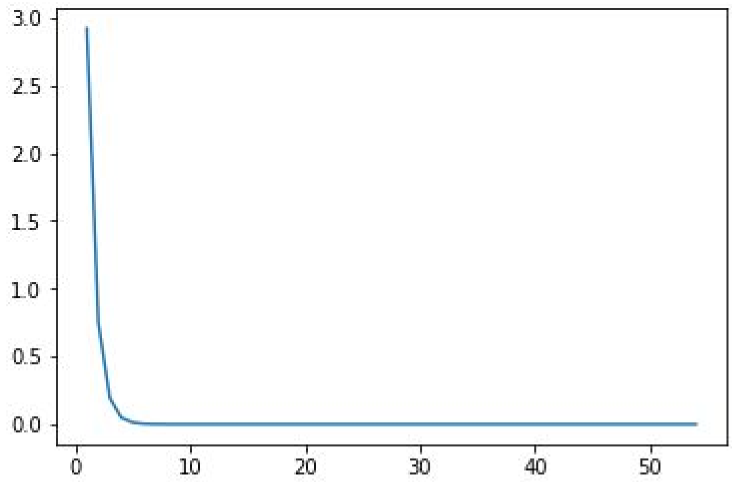
The final theta values using Gradient Descent obtained were

Final theta values : Final theta: [3101.57188523 -2362.88555452 -4063.51498634 -1766.32905256

-2888.0508516 1256.72655661 6672.96972752 -2275.01549192

-679.41165693 -2112.27513615 -2495.29173226 -1716.32170698]

And The Predicted Value at [1, 4000,2 ,1 ,1 ,1 ,1 ,0 ,0 ,0 ,0 ,0] came out to be 34454.59



Using Normal Equation the Final Cost came out to be 0.00144

And The Final theta values obtained were

Final theta values : [3.379200e+04 -3.303125e+02 -1.280000e+02 -2.560000e+02 4.725000e+01

-1.665000e+02 0.000000e+00 0.000000e+00 -1.275000e+01 -1.100000e+01

0.000000e+00 6.400000e+01]

And The Predicted Value at [1, 4000,2 ,1 ,1 ,1 ,1 ,0 ,0 ,0 ,0 ,0] came out to be 34144.29

It can be seen that the Final cost and the theta values using both gradient descent and normal equation have some difference and The predicted value by both methods is also not same.