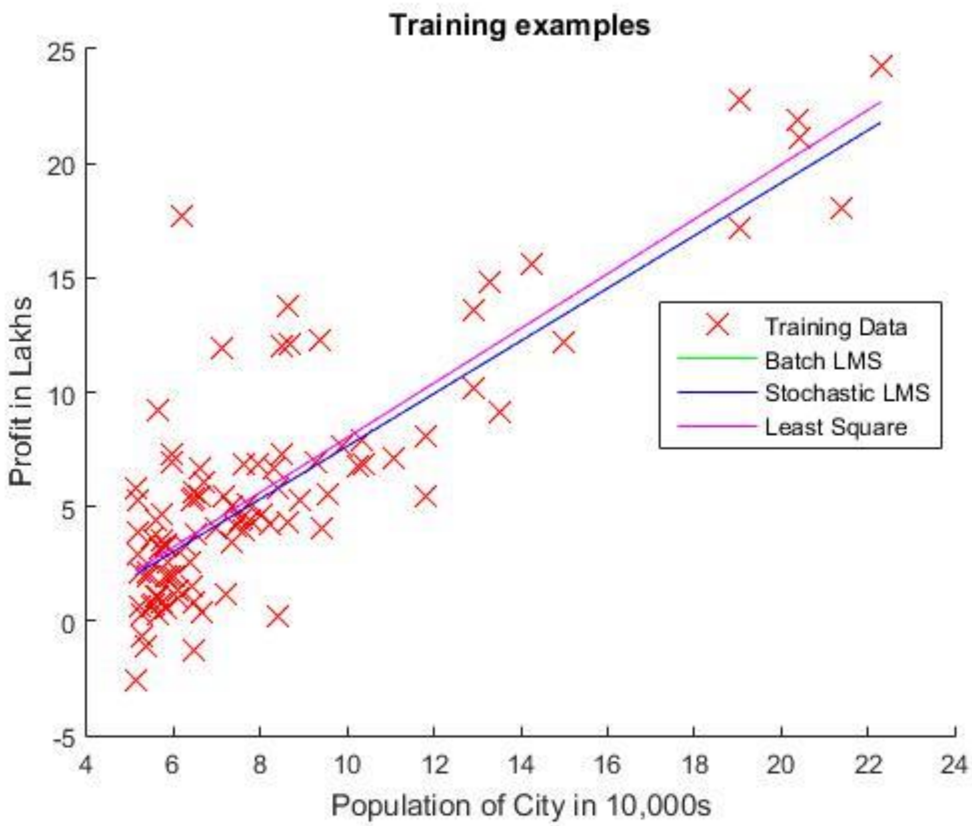
CLL 788
Assignment 1
Report

Bikash Gupta
2014CH70163

Plot of various methods over training samples



Least Square Closed Form solution overlaps with Batch LMS. However, the convergence is much faster in case of Least Square method.

Comparison of Convergence and Time Complexity

Method	Number of Iterations	Time
Batch LMS	7836	0.10938
Stochastic LMS	7777	1.9375
Least Square Closed Form	N/A	0

We can verify from the above table that Least Square Closed Form is the fastest among the three methods.

However, though Stochastic LMS converges with less number of iterations, accuracy is often held by Batch LMS. Time complexity however, tells the different story as Stochastic took more time than Batch even though the number of iterations were less because I've implemented batch LMS in vectorized form.

Theta Comparison over different methods

Hypothesis $h_\theta(x)$ is given by $h_\theta(x) = \theta_1 + \theta_2 X$

Method	Theta(1) or θ_1	Theta(2) or θ_2
Batch LMS	-3.9151	1.1930
Stochastic LMS	-3.8641	1.1503
Least Square Closed Form	-3.9151	1.1930

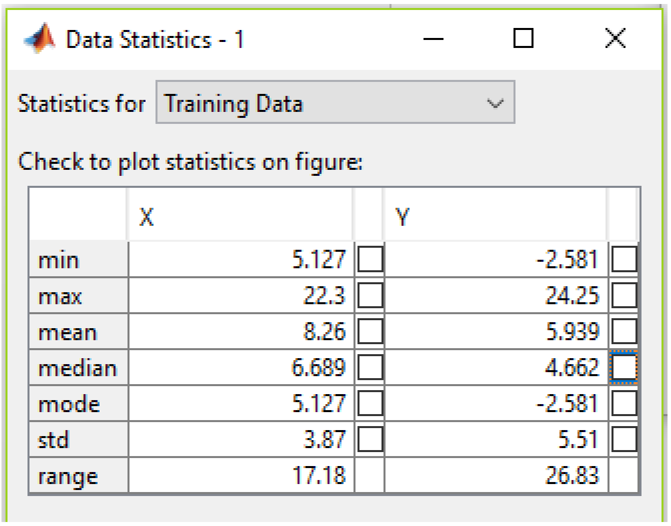
The theta obtained from Batch and Least Square were same when learning rate was taken to be 0.0001.

I took the learning rate to be small to avoid overshooting. I observed this phenomena during this assignment as when I took the learning rate to be 0.1, it did not converge.

Theta for Stochastic LMS was different. This can be explained from the fact that in Stochastic LMS, each time when a training example is encountered, the parameters are updated according to that single training example only. Stochastic LMS helps to reach θ to the minimum much faster than batch LMS (in terms of iterations). However that it may never converge to the minimum, due the update based on single example.

Data Statistics

Training Data Statistics



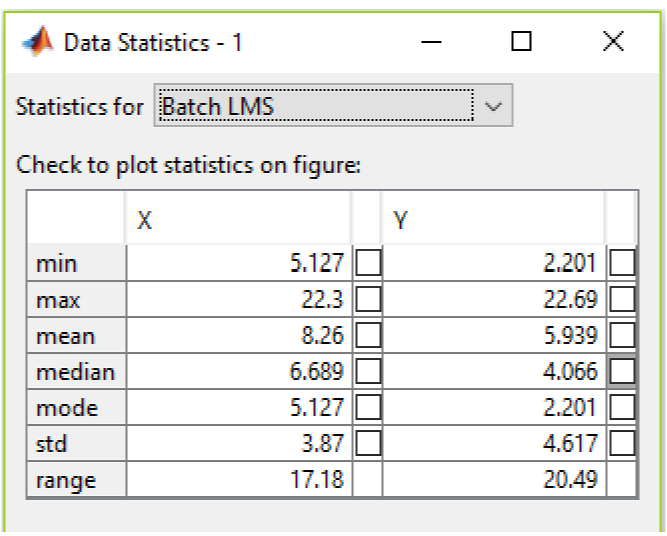
Data Statistics - 1

Statistics for Training Data

Check to plot statistics on figure:

	X	Y
min	5.127	-2.581
max	22.3	24.25
mean	8.26	5.939
median	6.689	4.662
mode	5.127	-2.581
std	3.87	5.51
range	17.18	26.83

Batch LMS Data Statistics



Data Statistics - 1

Statistics for Batch LMS

Check to plot statistics on figure:

	X	Y
min	5.127	2.201
max	22.3	22.69
mean	8.26	5.939
median	6.689	4.066
mode	5.127	2.201
std	3.87	4.617
range	17.18	20.49

Stochastic LMS Data Statistics

Data Statistics - 1

Statistics for Stochastic LMS

Check to plot statistics on figure:

	X	Y
min	5.127	2.033
max	22.3	21.79
mean	8.26	5.637
median	6.689	3.831
mode	5.127	2.033
std	3.87	4.451
range	17.18	19.76

Least Square Data Statistics

Data Statistics - 1

Statistics for Least Square

Check to plot statistics on figure:

	X	Y
min	5.127	2.201
max	22.3	22.69
mean	8.26	5.939
median	6.689	4.066
mode	5.127	2.201
std	3.87	4.617
range	17.18	20.49

Root Mean Square Error for the three methods

Methods	RMS Error
Batch LMS	2.9923
Stochastic LMS	3.0120
Least Square Closed Form	2.9923

As we can see, Batch and Least Square both perform well and has lesser RMS error when compared to Stochastic LMS due to the reasons as stated above.

Weighted Linear Regression

Locally Weighted Linear Regression (LWR)

This is a 1D case where $\theta = [\theta_0, \theta_1]$

The cost function $J(\theta)$ is a weighted version

Now

$$J(\theta) = \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right)^2$$

$$\frac{\partial J}{\partial \theta_0} = -2 \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right)$$

$$\frac{\partial J}{\partial \theta_1} = -2 \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right)$$

Cancelling the -2 terms, equating to zero, expanding and re-arranging the terms:

$$\frac{\partial J}{\partial \theta_0} = \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right) = 0$$

$$\sum_{i=1}^{m} w^{(i)} \theta_0 + \sum_{i=1}^{m} w^{(i)} \theta_1 x^{(i)} = \sum_{i=1}^{m} w^{(i)} y^{(i)} \rightarrow Eq(1)$$

$$\frac{\partial J}{\partial \theta_1} = \sum_{i=1}^{m} w^{(i)} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right) x^{(i)} = 0$$

$$\sum_{i=1}^{m} w^{(i)} \theta_0 + \sum_{i=1}^{m} w^{(i)} \theta_1 x^{(i)} x^{(i)} = \sum_{i=1}^{m} w^{(i)} y^{(i)} x^{(i)} \rightarrow Eq(2)$$

Writing eqn(1) & eq(2) in matrix form $A\theta = b$ allows us to solve for θ

$$\sum_{i=1}^{m} w^{(i)} \theta_0 + \sum_{i=1}^{m} w^{(i)} \theta_1 x^{(i)} = \sum_{i=1}^{m} w^{(i)} y^{(i)}$$

$$\sum_{i=1}^{m} w^{(i)} \theta_0 + \sum_{i=1}^{m} w^{(i)} \theta_1 x^{(i)} x^{(i)} = \sum_{i=1}^{m} w^{(i)} y^{(i)} x^{(i)}$$

$$\begin{bmatrix} \Sigma w^{(i)} & \Sigma w^{(i)} x^{(i)} \\ \Sigma w^{(i)} x^{(i)} & \Sigma w^{(i)} x^{(i)} x^{(i)} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \Sigma w^{(i)} y^{(i)} \\ \Sigma w^{(i)} y^{(i)} x^{(i)} \end{bmatrix}$$

$$A \quad \theta = b$$

$$\boxed{\therefore \theta = A^{-1} b}$$

Now,

$x^{(1)} = 6.2101$ $\quad y^{(1)} = 17.692$

$x^{(2)} = 5.6277$ $\quad y^{(2)} = 9.2302$

$x^{(3)} = 8.6186$ $\quad y^{(3)} = 13.762$

$x^{(4)} = 7.1032$ $\quad y^{(4)} = 11.954$

Given $x = 7.576$

$\tau = 0.5$

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right) = \exp-\left(\frac{(x^{(i)} - 7.576)^2}{2\times(0.5)^2}\right)$$

$\therefore w^{(1)} = 0.02396$

$w^{(2)} = 5.04 \times 10^{-4}$

$w^{(3)} = 0.11371$

$w^{(4)} = 0.63949$

Now

$\Sigma w^{(i)} = 0.777664$ $\quad \Sigma w^{(i)} x^{(i)} = 5.67407$

$\Sigma w^{(i)} x^{(i)} x^{(i)} = 41.65215$

$\Sigma w^{(i)} y^{(i)} = 9.63789$

$\Sigma w^{(i)} y^{(i)} x^{(i)} = 70.44584$

Now,

$$\begin{bmatrix} 0.777664 & 5.67407 \\ 5.67407 & 41.65215 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 9.63789 \\ 70.44584 \end{bmatrix}$$

$$A \quad \theta = b$$

$$\theta = A^{-1} b$$

Solving, we get

$$\begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 8.7832 \\ 0.4948 \end{bmatrix}$$ Ans.

Using same analysis (for 4 points) using stochastic LMS we get the following $\begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$ values.

iterno:	θ_0	θ_1
iter 1	0.0018	0.0110
iter 2:	0.0027	0.0161
iter 3 :	0.0040	0.0279
iter 4 :	0.0052	0.0362
iter 5 :	0.0070	0.0477
iter 6 :	0.0075	0.0521
iter 7	0.0092	0.0636
iter 8 :	0.0103	0.0717