CSL341 – Machine Learning

Assignment 1

Submission Date	**Nipun Shrivastava**
	2011cs50288
11/09/14	**Group 5**

1. Linear Regression

For pre-processing, we normalised the data and then added a column of ones to account for intercept term. We chose the learning rate as per given in the assignment question. J_theta(x) was the same function as discussed in class. Stopping parameter was when the change in value of J_theta(x) between two consecutive iterations became less than a certain threshold. Gradient of J_theta w.r.t. theta is given by

grad(J_theta) = -x_data'*(y_data - x_data*theta)

With increasing value of learning rate, program took less and less iterations to converge, however, 1.3 took more iterations than 0.9 because 1.3 was big enough leaning rate for our program to actually go to the other side of optima with each iteration. This was clear from the contour plot. Any learning rate greater than 2 caused the program to diverge.

LEARNING RATE	ITERATIONS
0.1	72
0.5	14
0.9	6
1.3	9
2.5	Diverges
2.1	Diverges

For learning rate = 0.1, value of Theta(2nd term is the intercept term): **4.614357,5.836172**.

Learning rate equal to 1 seems the best bet for this case as it only takes 4 iterations and doesn't even oscillate around the optimal.

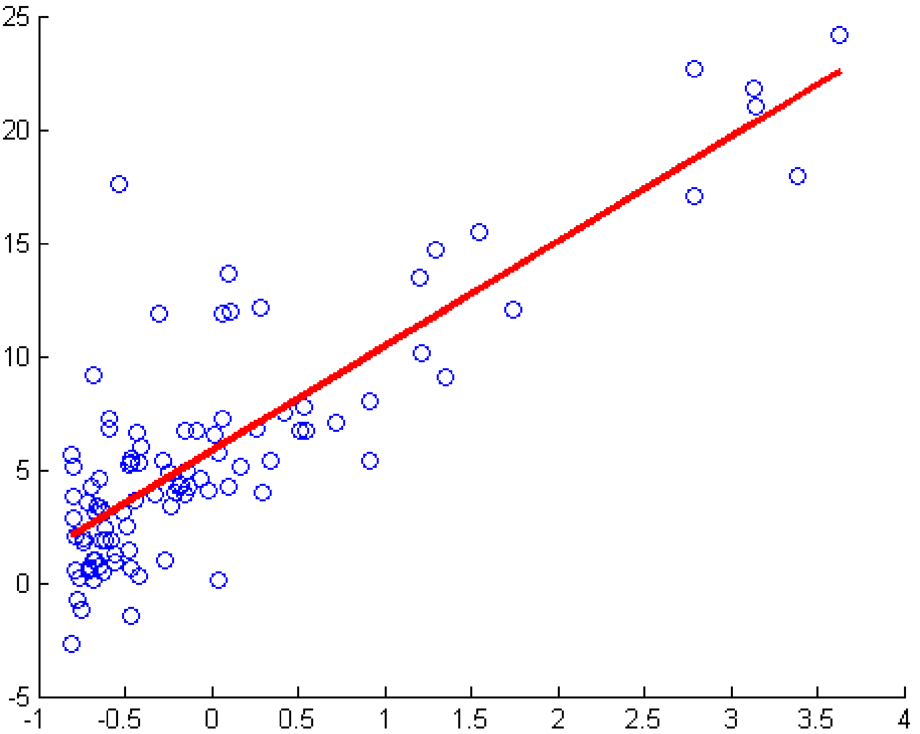


Figure 1 Decision Boundary - Scatter Plot

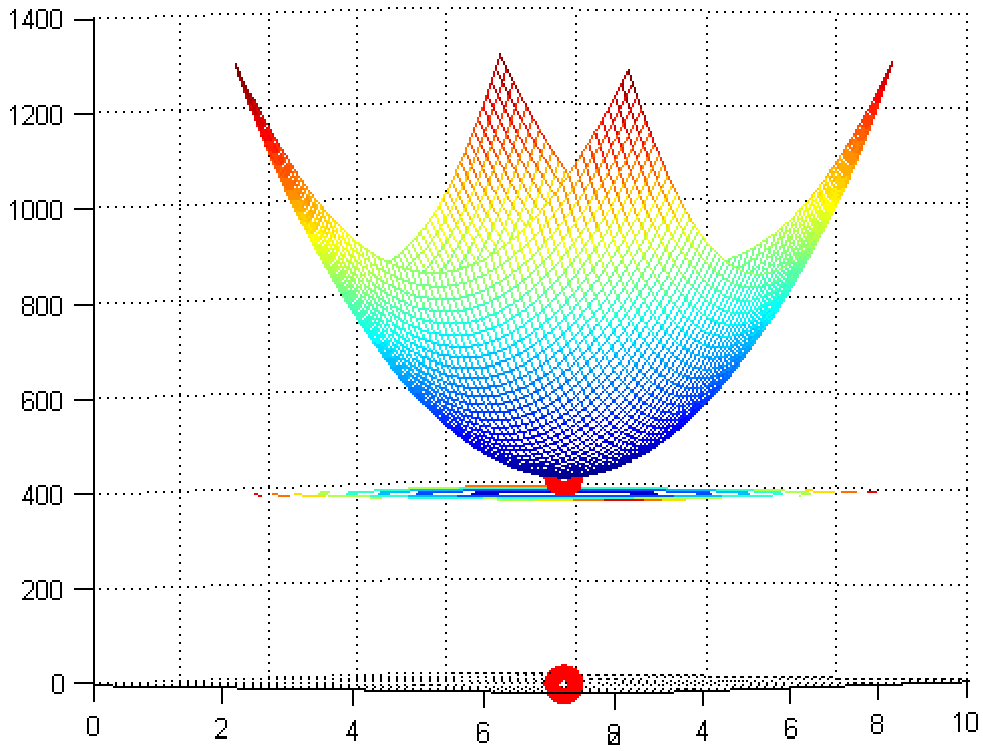


Figure 2 3D Surface Plot - Optimum Theta

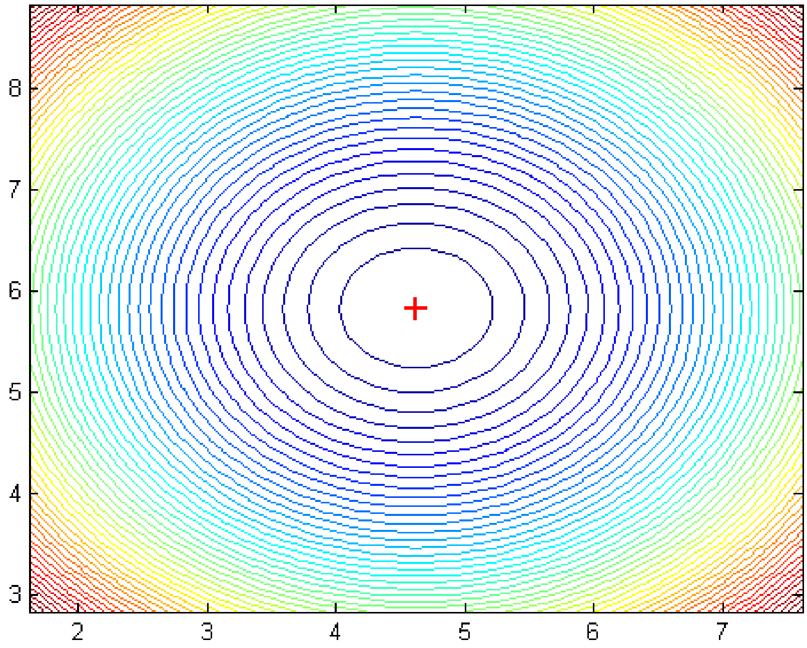


Figure 3 Contour Plot - Optimum Theta

2. Logistic Regression

Equations used

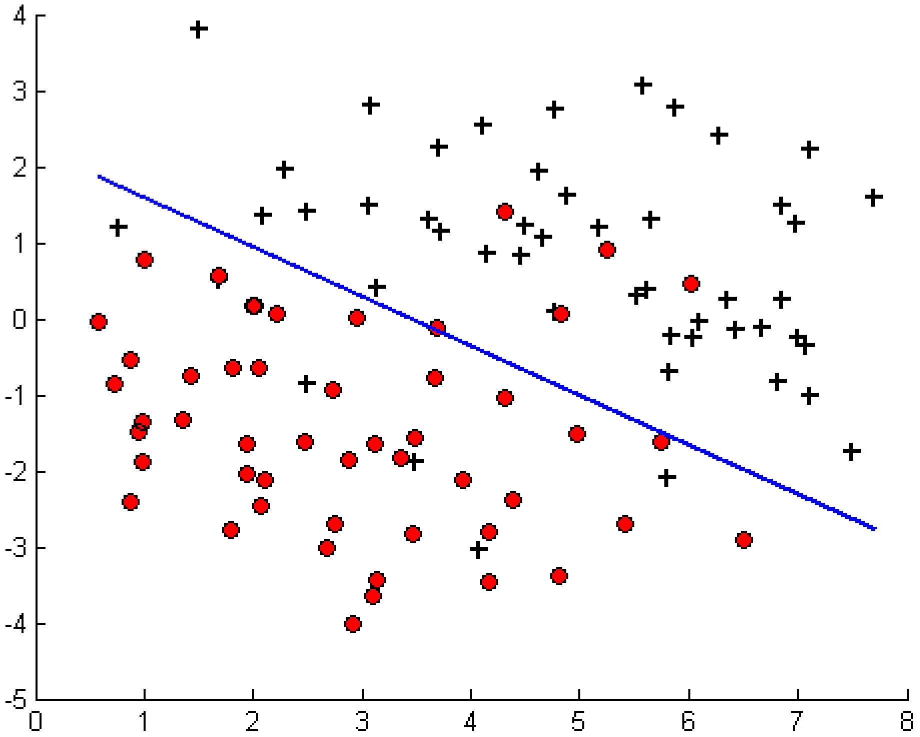
grad(L) = x'*(h_theta' - y)

hess(L) = -x'*diag(h_theta .* (1-h_theta))*x

theta = theta - inv(hessian)*gradient;

The last equation is the trademark equation of Newton's method.

Value of Theta(2nd term is the intercept term) obtained using this method: **0.760372,1.171947**



.

Figure 4 Decision Boundary - Scatter Plot

3. Locally Weighted Linear Regression

a) Equation used in case of unweighted linear regression:

theta = inv(X'X)*(X'Y)

Value obtained by this equation: **0.175311,0.327675**

b) Equation used in case of weighted:

theta = (X'WX)\(X'WY)

where **W** is the weight matrix.

c) **0.8** which was given as part of the question seems to work best for Ʈ. When Ʈ was made too small the curve resulted in the case of overfitting with various kinks in the curve. While large values of Ʈ did not seem to encapsulate the basic reason behind weighted regression and was unaffected to a great extent even when large cluster of points appeared on one side of the decision boundary. A classic case of underfitting. The curve will tend to unweighted linear regression straight-line when tau->infinity.

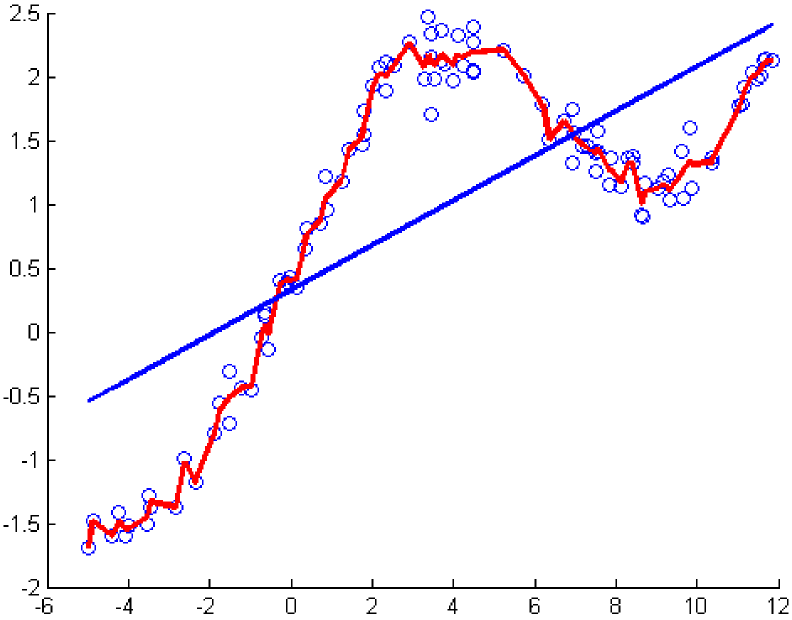


Figure 5 Weighted Linear Regression – Ʈ=0.1

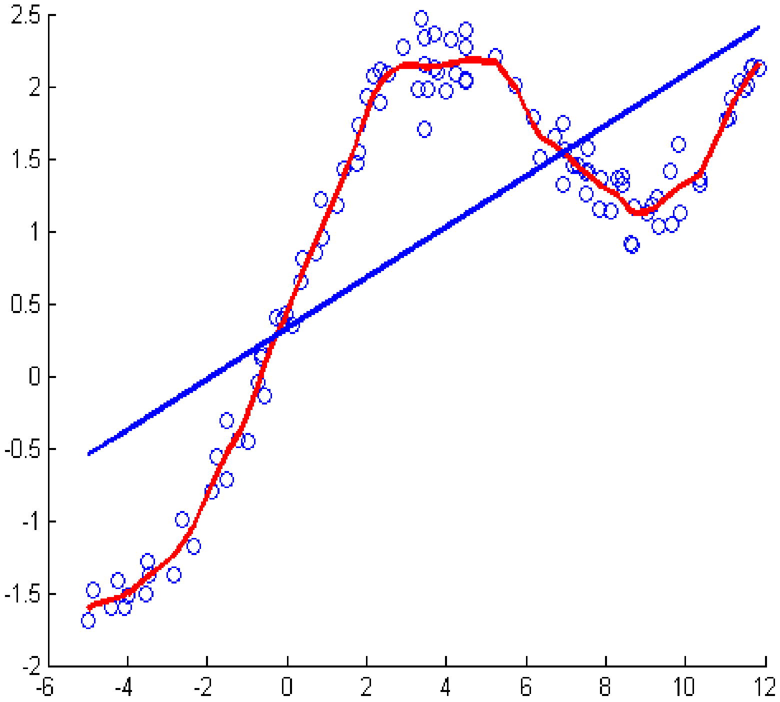


Figure 6 Weighted Linear Regression – Ʈ=0.3

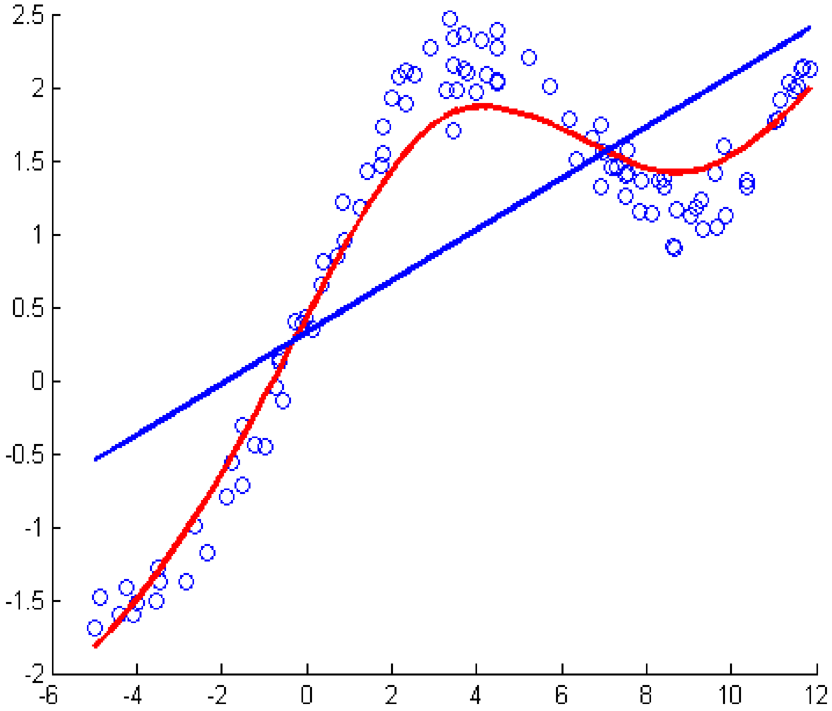


Figure 7 Weighted Linear Regression – Ţ=2

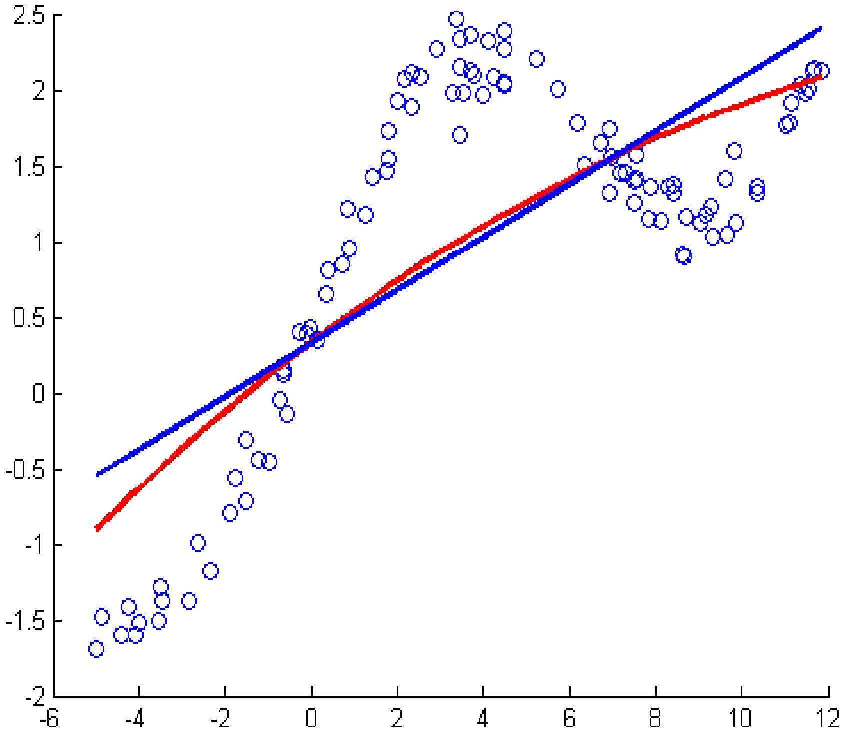


Figure 8 Weighted Linear Regression – Ţ=10

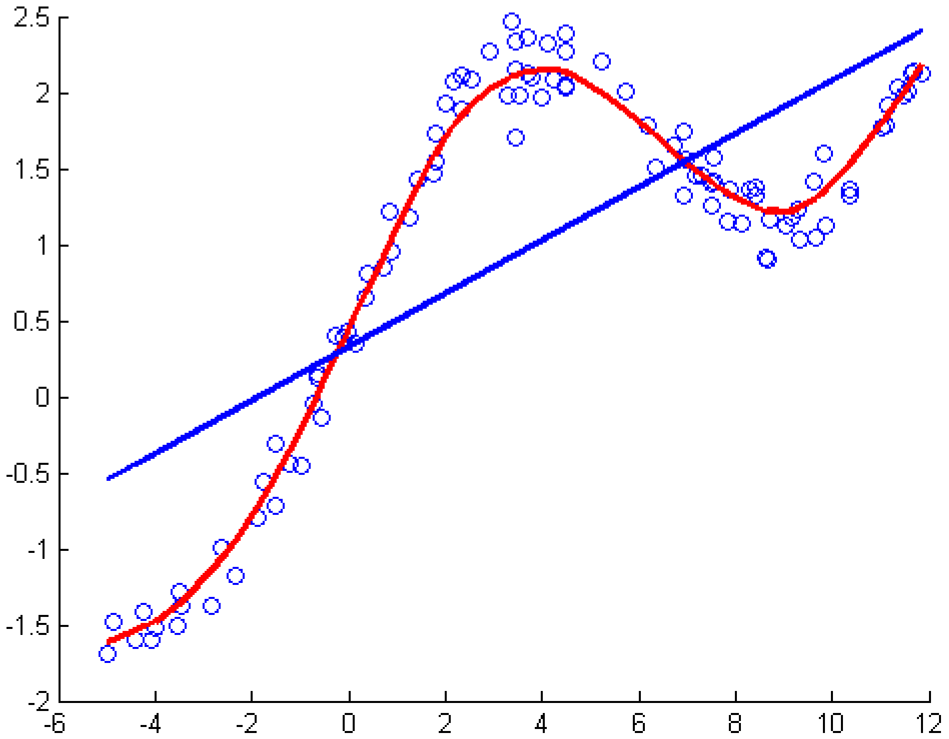


Figure 9 Weighted Linear Regression – Ʈ=0.8

4. Gaussian Discriminant Analysis

a) mu1 =

98.3800	429.6600

mu2 =

137.4600	366.6200

sigma =

1.0e+03 *

0.2875	0.0267
0. 0267	1.1233

c)

x'\sigma*(mu0-mu1) - 0.5*(mu0+mu1)\sigma*(mu0-mu1) + log(phi/(1-phi) = 0

is the equation used for LDA. It guves us a linear decision boundary.

d) sigma1 =

1.0e+03 *

0.2554	0.1843
0. 1843	1.3711

sigma2 =

319.5684	130.8348
130.8348	875.3956

Mean in this case will be same as in part (a).

e)

```
2*log(phi/(1-phi))-log(det(sigma0)/det(sigma1))-([x;y]-mu0)'/sigma0*([x;y]-mu0)+([x;y]-mu1)'/sigma1*([x;y]-mu1) = 0
```

is the equation used for QDA. This will give us a decision boundary corresponding to a quadratic function.

f) Quadratic boundary seems to better encapsulate the nature of data given to us. Evan the misclassifications in quadratic case is less as compared to linear. Though this was not a good example to see the advantages of quadratic discriminant analysis as the data seemed more or less linearly separable. Quadratic case might lead us to slight overfitting. But no assumptions on the covariance matrices in case of QDA reduces one constraint on input data for us.

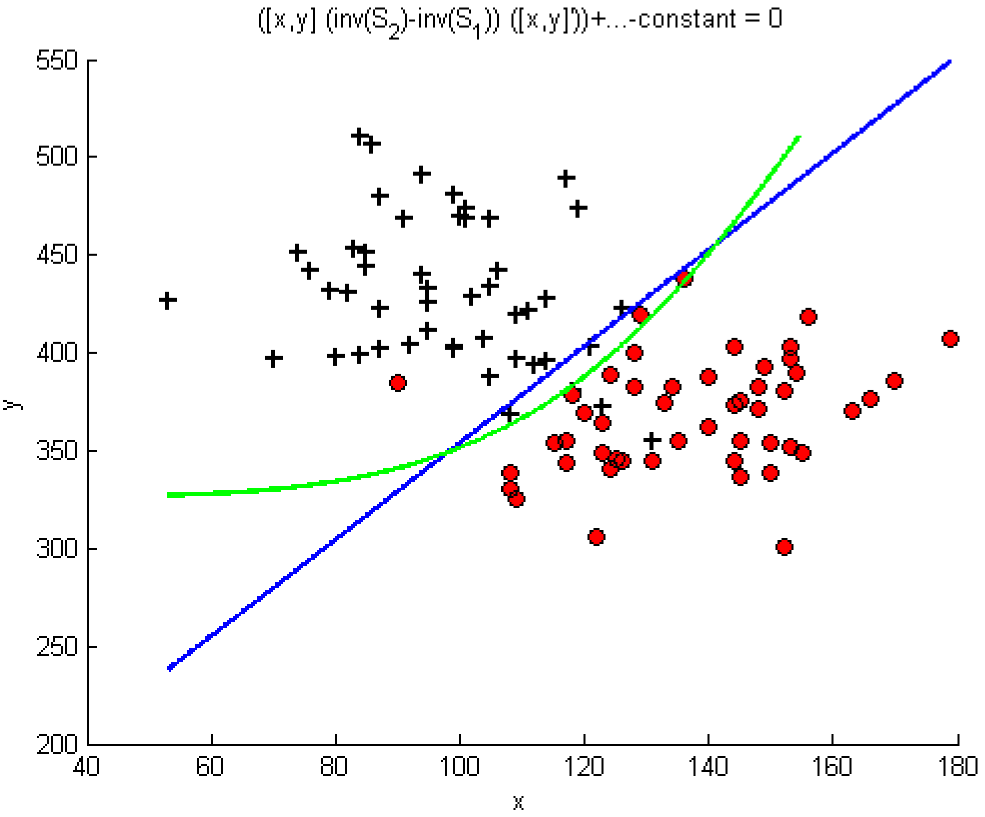


Figure 10 LDA and QDA