COL774 : Assignment 1

Aditi Singla
2014CS50277

10th January 2017

1. Linear Regression

   (a) The values of x are first normalised. The cost function J(θ) is then optimised by batch gradient descent algorithm. The value of θ obtained is then transformed to get θ for the given unnormalised data.

   The parameters used are as follows:

   - Learning rate, $\eta = 0.02$
   - Stopping Criteria : $||(\nabla\ \theta)|| <= 0.0000001$
   - No. of iterations = 681
   - Final θ (normalised) = [5.8391288,4.5930362]
   - Final θ (unnormalised) = [-3.8957767,1.1930323]

   (b) Two-dimensional graph for data and hypothesis learned:

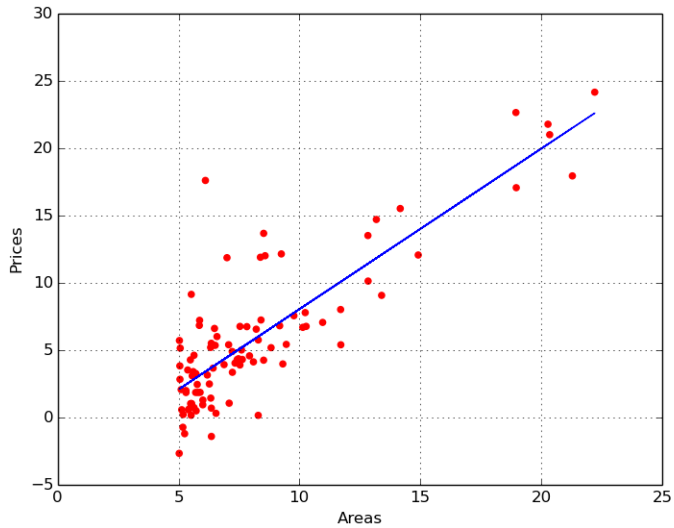

Figure 1: Data Points and Hypothesis Function

(c) Three-dimensional mesh/surface for the cost function J(θ):

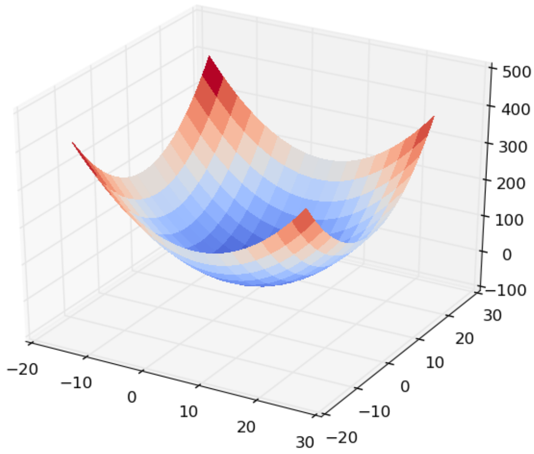

Figure 2: 3D mesh for J(θ)

(d) Contour Graph of error function

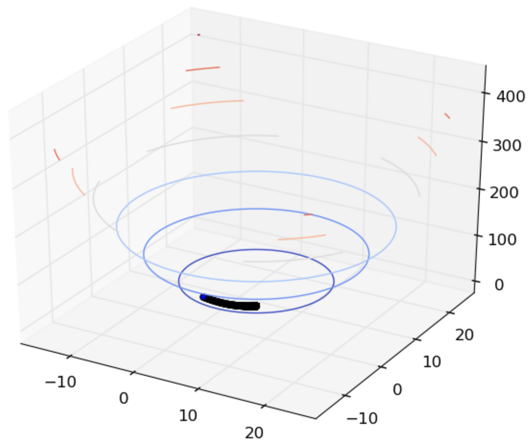

Figure 3: Contour Graph for J(θ) at $\eta = 0.02$

(e) Contour Graph for different learning rates:

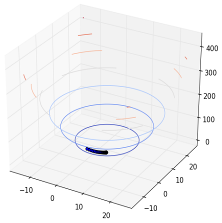

Figure 4: Contour Graph at $\eta = 0.1$

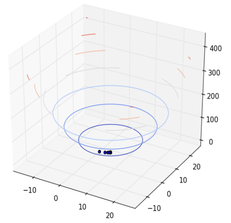

Figure 5: Contour Graph at $\eta = 0.5$

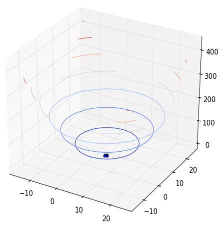

Figure 6: Contour Graph at $\eta = 0.9$

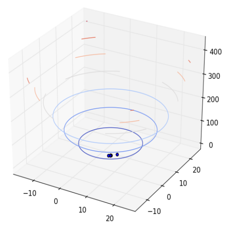

Figure 7: Contour Graph at $\eta = 1.3$

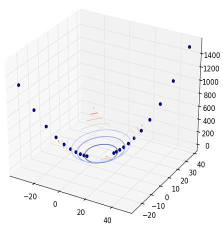

Figure 8: Contour Graph at $\eta = 2.1$

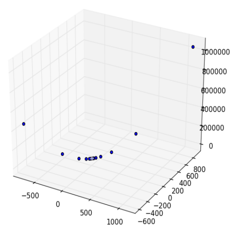

Figure 9: Contour Graph at $\eta = 2.5$

The graphs have been plotted for various values of η. We can observe that for small values of η, the cost function converges to the minimum value in considerable number of iterations (which is way too less than that for unnormalised data). As we increase the value of η, the error function converges faster, as the number of iterations decreases. For the value of $\eta = 1.3$, we observe an oscillation initially, but the overshoot being small, the cost function eventually converges to the minimum. But, when we reach $\eta = 2.1$ and $\eta = 2.5$, the overshoot is large and hence the value of cost function fails to converge.

2. Locally Weighted Linear Regression

(a) (Unweighted) Linear Regression on the dataset (using normal equations):

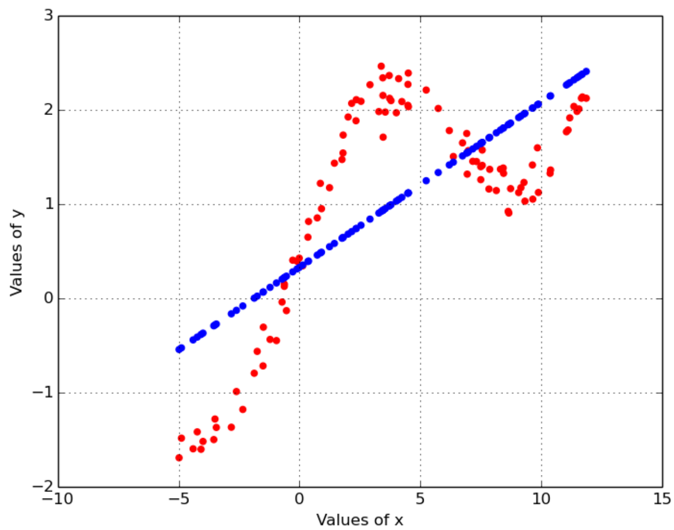

Figure 10: (Unweighted) Linear Regression

(b) (Weighted) Linear Regression with $\tau = 0.8$:

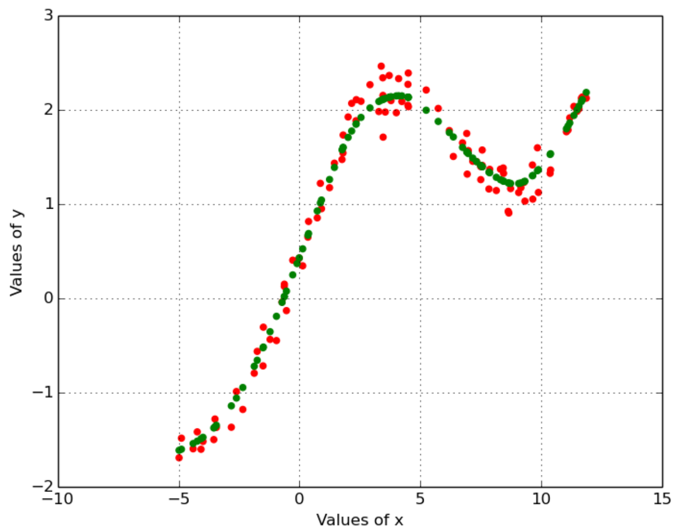

Figure 11: Locally Weighted Regression with $\tau = 0.8$

(c) (Weighted) Linear Regression with different τ values:

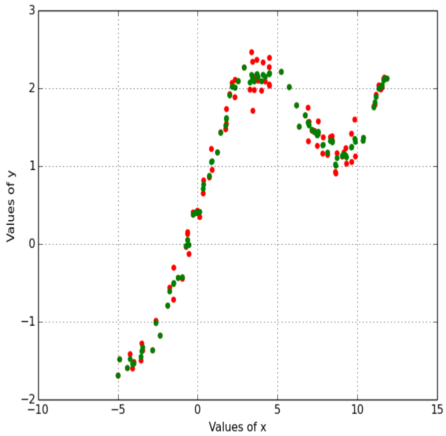

Figure 12: LWR with $\tau = 0.1$

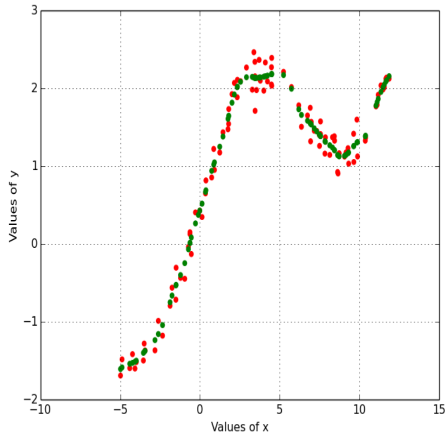

Figure 13: LWR with $\tau = 0.3$

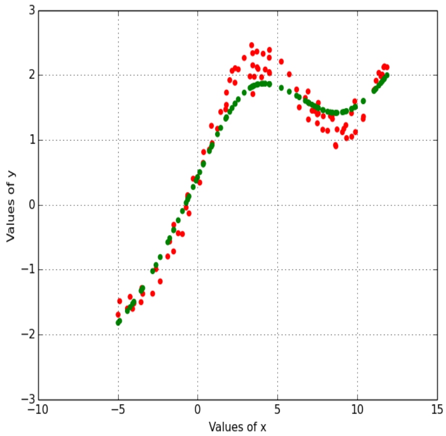

Figure 14: LWR with $\tau = 2$

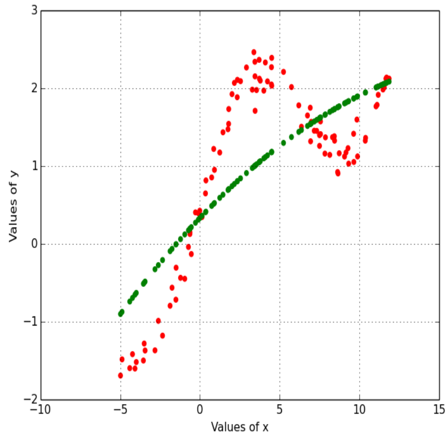

Figure 15: LWR with $\tau = 10$

The graphs have been plotted for locally weighted regression on various values of τ. For very small values of τ, the algorithm seems to overfit on the data as it tries to fit on every training data point. As the value of τ increases, the algorithm fits nicely to the data (good fit). For large values of τ, the weights tend to become 1, which is actually unweighted linear regression. So, for τ=10, we observe an almost straight line. For the given values of τ, we observe a good fit at τ=0.3.

3. Logistic Regression

(a) The value of cost function converges to a minimum value in a very few iterations, using Newtons method. The parameters used are as follows:

- Stopping Criteria : $||(\nabla\ \theta)|| <= 0.000001$
- No. of iterations = 7
- Final θ = [-2.620511, 0.760371, 1.17194]

(b) Two-dimensional graph for data and decision boundary:

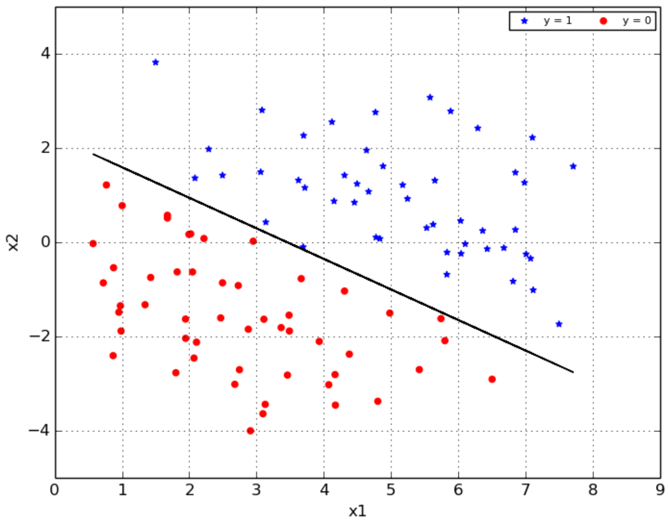

Figure 16: Logistic Regression

4. Gaussian Discriminant Analysis

(a) The values of μ_0, μ_1 and Σ obtained are

- $\phi = 0.5$
- $\mu_0 = \begin{pmatrix} 98.38 & 429.66 \end{pmatrix}$
- $\mu_1 = \begin{pmatrix} 137.46 & 366.62 \end{pmatrix}$
- $\Sigma = \begin{pmatrix} 287.482 & -26.748 \\ -26.748 & 1123.25 \end{pmatrix}$

(b) Graph for the given data

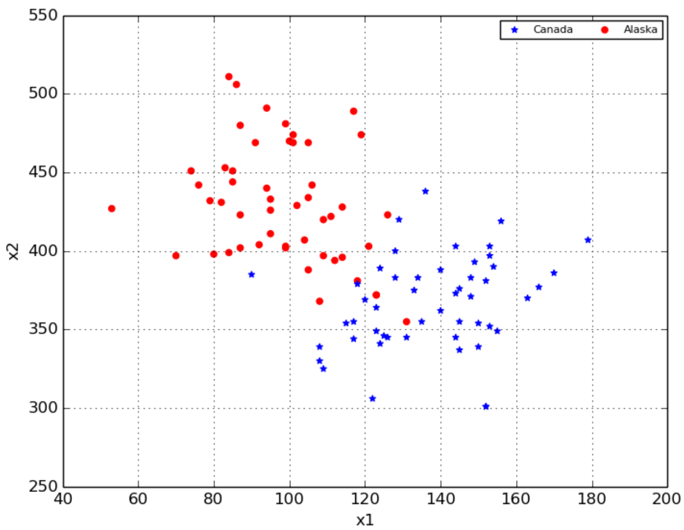

Figure 17: Given Data

(c) Equation corresponding to the decision boundary given $\Sigma = \Sigma_0 = \Sigma_1$ (Linear Equation)

$$(\mu_0 - \mu_1)^T \Sigma^{-1} x + \frac{\mu_1^T \Sigma^{-1} \mu_1 - \mu_0^T \Sigma^{-1} \mu_0}{2} - \log \frac{\phi}{1-\phi} = 0$$

(d) The values of μ_0, μ_1, Σ_0 and Σ_1 obtained are

- $\phi = 0.5$
- $\mu_0 = \begin{pmatrix} 98.38 & 429.66 \end{pmatrix}$
- $\mu_1 = \begin{pmatrix} 137.46 & 366.62 \end{pmatrix}$
- $\Sigma_0 = \begin{pmatrix} 255.3956 & -184.3308 \\ -184.3308 & 1371.1044 \end{pmatrix}$
- $\Sigma_1 = \begin{pmatrix} 319.5684 & 130.8348 \\ 130.8348 & 875.3956 \end{pmatrix}$

(e) Equation corresponding to the decision boundary given $\Sigma_0 \neq \Sigma_1$ (Quadratic Equation)

$$x^T \frac{(\Sigma_0^{-1} - \Sigma_1^{-1})}{2} x + (\mu_1^T \Sigma_1^{-1} - \mu_0^T \Sigma_0^{-1}) x - \frac{\mu_1^T \Sigma_1^{-1} \mu_1 - \mu_0^T \Sigma_0^{-1} \mu_0}{2} + \log \frac{\phi}{1-\phi} \frac{||\Sigma_0||^{0.5}}{||\Sigma_1||^{0.5}} = 0$$

(f) Observations:

- In case of same covariance matrices, linear decision boundary is observed, while for different covariance matrices, a quadratic decision boundary is observed. When $\Sigma_0 = \Sigma_1$, first term gets cancelled, eliminating the quadratic element.
- The quadratic decision boundary is actually a hyperbola.
- The points near the two decision boundaries are wrongly classified in case of same covariance matrices. So, quadratic decision boundary classifies the data better.

The plots for the decision boundaries are as follows :

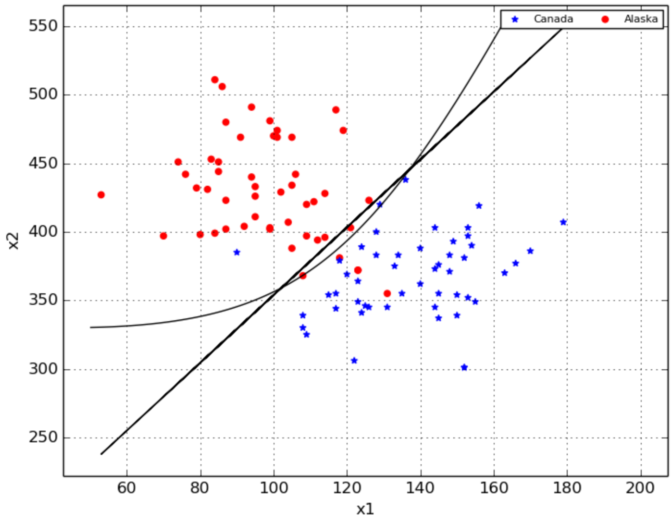

Figure 18: Decision boundaries for the two cases