ISCO630E-ASSIGNEMNT-4

Conclusion

By Nairit Banerjee-IIT2016505

Question 1

We need to apply Logistic regression using Newton's method on the **Two exam results dataset** to whether a student gets admitted or not.

The regularized cost function is defined as

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2.$$

We then apply gradient descent as follows

$$\theta^{(t+1)} = \theta^{(t)} - \mathrm{H}^{-1} \nabla_\theta \mathrm{J}$$

$$\text{where Gradient: } \nabla_\theta \mathrm{J} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

$$\text{and Hessian: } H = \frac{1}{m} \sum_{i=1}^{m} [h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})) x^{(i)} (x^{(i)})^T]$$

where

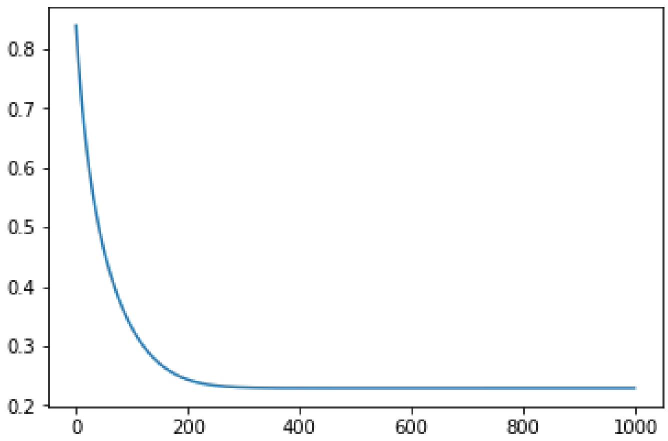
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

The data is first normalized and splitted in 70:30 ratio for training and testing respectively.

We apply newton's method **regularization parameter 0.1**.

The **final value of cost function is 0.229**.

The cost vs iterations graph is as follows

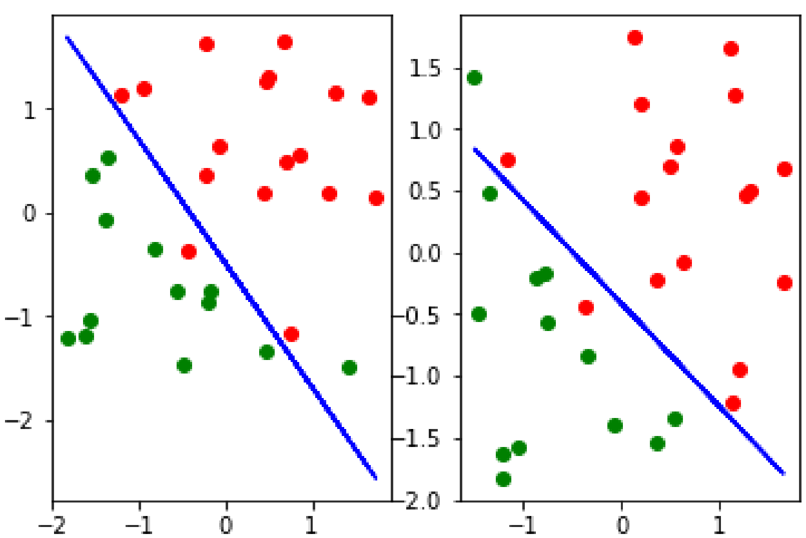


We observe that the cost decreases much more quickly when Newton's method is used over gradient descent.

The **final values of the coefficients are : [1.46307305 3.57748299 3.00206375]**

We also got an **accuracy of 93.33% on the testing data**.

The decision boundaries for both features individually can be visualized as



Question 2

We need to apply Logistic regression on the **Microchip Quality dataset** to predict whether a chip gets accepted or rejected.

The regularized cost function is defined as

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2.$$

Now we apply Gradient Descent as follows

$$\theta^{(t+1)} = \theta^{(t)} - \mathrm{H}^{-1}\nabla_\theta \mathrm{J}$$

$$\text{where Gradient: } \nabla_\theta \mathrm{J} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

$$\text{and Hessian: } H = \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))x^{(i)}(x^{(i)})^T]$$

where

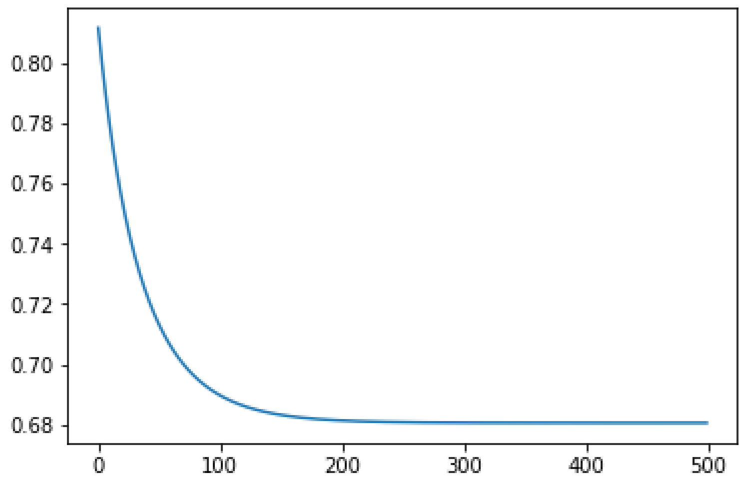
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

The data is first normalized and splitted in 70:30 ratio for training and testing respectively.

We apply Newton’s method **regularization parameter 0.1.**

The **final value of cost function is 0.680**.

The cost vs iterations graph is as follows,

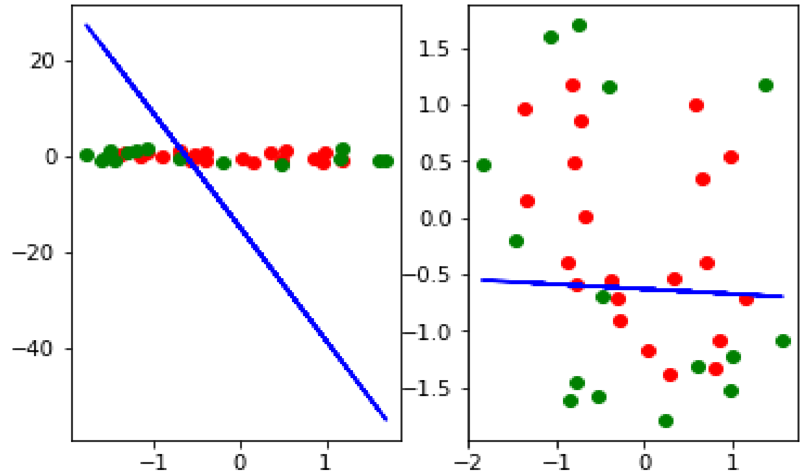


Again, Newton's method minimizes the cost faster than gradient descent.

The **final values of the coefficients are : [-0.16522306 -0.26187431 -0.01112865]**

We also got an **accuracy of 36.11% on the testing data**.

The decision boundary for both features individually looks like,

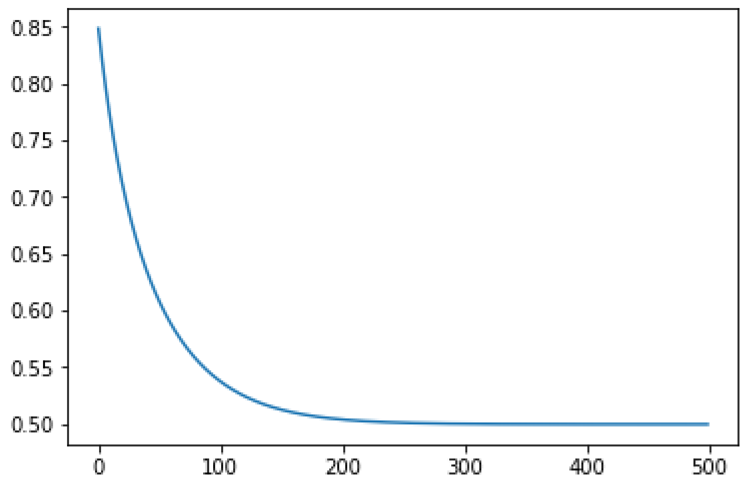


Clearly the model is underfitting the data. Hence we **introduce new features**, which are **X1^2, X1^3, X2^2 and X2^3** where X1 and X2 were our initial features.

We apply gradient descent and final cost came out to be **0.499**.

The coefficients after training were : **[-0.02272692 -0.88678597 -0.47770449 -1.47250695 -1.5499411 0.97223947 1.52790875]**

The cost vs iterations graph is as follows,



Using the same parameters as above, we now get an **accuracy of 88.89%** on the testing split, which is much better than when only two features were used and also marginally better than Logistic Regression applied with gradient descent.