Q3

February 11, 2018

0.1 Logistic Regression using Newton's Method

In this part we implement classification using logistic regression using Newton's method for optimization.

We calculate the Grad and Hessian matrices and use the following update rule to update our parameters (initialized at zero).

$$\theta_{t+1} = \theta_t - H(f(\theta_t))^{-1} \nabla f(\theta_t)$$

Following is the output of our optimization. The first parameter corresponds to the first feature, second to the second feature and last one is the bias. It is clear that our algorithm converges sufficiently in only one iteration and after this all the parameters are just scaled by some number. After five iterations overflow is encountered and the algorithm terminates.

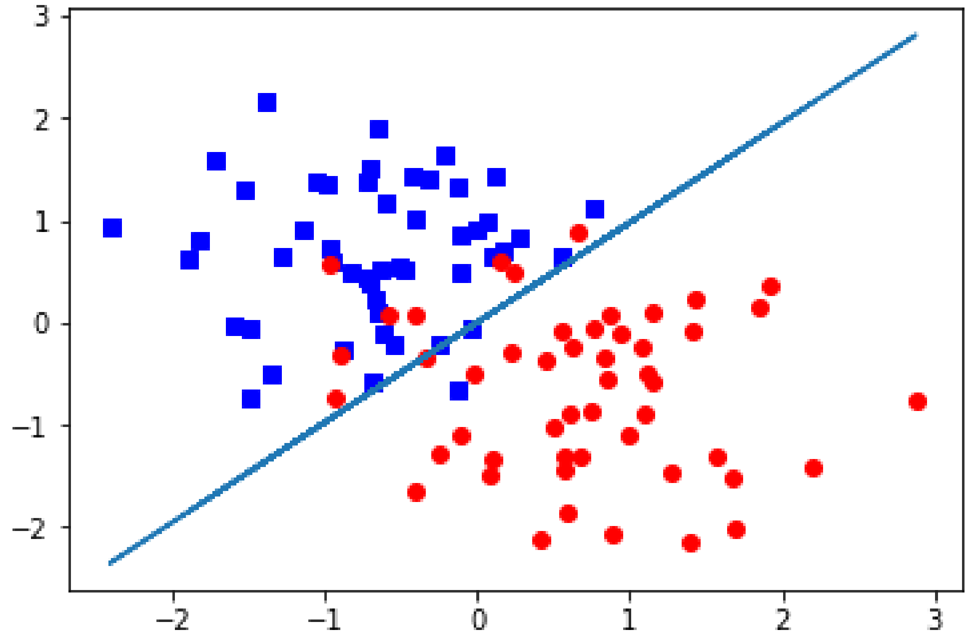
```
Iteration =  1
Theta array =  [ -8.97341408e-01   9.20112283e-01   3.85905135e-16]
Iteration =  2
Theta array =  [-4.88262336  5.02247361 -0.28086918]
Iteration =  3
Theta array =  [-326.74629425  321.51992179  -63.86126337]
Iteration =  4
Theta array =  [ -5.65776800e+10   5.55708777e+10  -1.15565907e+10]
Overflow in iteration 5
```

```
/home/madhur/.local/lib/python3.6/site-packages/ipykernel_launcher.py:2: RuntimeWarning: overfl
```

Thus a reasonable equation of the boundary is $ -8.97341408e-01 * x_0 + 9.20112283e-01* x_1 + 3.85905135e-16

Below is the plot of our decision boundary and the data points, it can be seen that our boundary separates the two classes reasonably well.

```
Out[9]: [<matplotlib.lines.Line2D at 0x7f356c952b00>]
```



3
2
1
0
−1
−2
−2
−1
0
1
2
3