CS289_HW04_Prob1

March 5, 2017

```
In [29]: import numpy as np
         import scipy.special as spsp
         import math

In [23]: lam = 0.07

In [24]: X = np.array([[0,3,1],[1,3,1],[0,1,1],[1,1,1]])
         y = np.array([1,1,0,0])

In [25]: w0 = np.array([-2,1,0])

In [26]: def Omega(X,w):
             Om = np.identity(len(X))
             for i in range(len(Om)):
                 s_i = 1/(1+math.exp(-np.dot(X[i],w)))
                 Om[i,i] = s_i*(1-s_i)

             return Om
```

For the first iteration, we can define our variables $\Omega^{(0)}$ and $s^{(0)}$ as

```
In [31]: Omega0 = Omega(X,w0)
         s0 = spsp.expit(np.dot(X,w0))
```

Then, we can solve for $e^{(0)}$ in $(X^T\Omega^{(0)}X + 2\lambda\mathbb{I})e^{(0)} = X^T(y - s^{(0)}) - 2\lambda w^{(0)}$

```
In [43]: def solve_e(lam,X,y,OmegaN,sN,wN):
             d = len(X[0])
             HessianJ = np.dot(np.dot(X.T,OmegaN),X) + 2*lam*np.identity(d)
             negGradJ = np.dot(X.T,(y-sN)) - 2*lam*wN
             e = np.linalg.solve(HessianJ,negGradJ)

             return e

In [39]: e0 = solve_e(lam,X,y,Omega0,s0,w0)
         print(e)

[ 1.61323601  0.40431761 -2.28417115]
```

```
In [44]: w1 = w0 + e0
         print(w1)
```

```
[-0.38676399  1.40431761 -2.28417115]
```

We can then calculate $\Omega^{(1)}$ and $s^{(1)}$ similarly,

```
In [45]: Omega1 = Omega(X,w1)
         s1 = spsp.expit(np.dot(X,w1))
         print(s1)
```

```
[ 0.87311451  0.82375785  0.29320813  0.21983683]
```

And repeat the iterative process to find $w^{(2)}$

```
In [46]: e1 = solve_e(lam,X,y,Omega1,s1,w1)
         w2 = w1 + e1
         print(w2)
```

```
[-0.51222668  1.45272677 -2.16271799]
```