Midterm

Philip Hasse

10/13/16

Solution to Problem (1):

(a) The worst way at finding both the best E_{out} and the lowest amount of computation required is using the Pocket algorithm starting at w=0 because it starts in an arbitrary location and tests each line from there, basically starting at nothing. In this problem, the error starts at approximately ~16.5% and after approximately 300-400 iterations the pocket algorithm gets approximately to the error the linear regression line started at. This method has a lot of wasted computation time.

(b) A slightly better way to find both best E_{out} and the lowest amount of computation required is using linear regression. This method is the easiest to perform computation wise, and will generally be fairly close to the true error, starting at around 10.8%. Therefore taking into account the extreme benefit in computation time it is a better method than the pocket algorithm starting at 0.

(c) The best way out of these three methods to find the best E_{out} with the lowest amount of computation required is using the Pocket algorithm starting for the solution given by linear regression. This method starts out with ~10.8% error (the linear regression error) and after a large number of iterations the pocket algorithm finds the error to be approximately 10.6%. This error would be found much quicker than the pocket algorithm starting at w=0 and is more accurate than the linear regression method, therefore making it the best method to classify the data.

Solution to Problem (2):

```
import numpy as np
import matplotlib.pyplot as plt

def solveN(N):
    for k in range(0,10):
        N = (8/.05**2)*np.log((4*((2*N)**10)+1)/.05)
        plt.scatter(k,N)
    plt.savefig('2.12.plot.pdf', bbox_inches='tight')
    plt.show()
    print N

solveN(1000)
```