Machine Learning HW 3

Authors: Lior Shachoach 206284960 and Moriya Barel 207304353

Question 1:

Num of mistakes until no error: 18

Final weights vector: [1, 0, 1, 0, 0, 32, 0, 0, 0, 0, 32, 4, 0, 0, 0, 32, 0, 0, 0, 4]

Question 2:

Results Report:

k = 1 empirical error: 0.2476, true error: 0.3887999999999999
k = 2 empirical error: 0.2764000000000001, true error: 0.38200000000000006
k = 3 empirical error: 0.104, true error: 0.25439999999999996
k = 4 empirical error: 0.102, true error: 0.24080000000000001
k = 5 empirical error: 0.056799999999999976, true error: 0.22200000000000003
k = 6 empirical error: 0.04079999999999999, true error: 0.20120000000000007
k = 7 empirical error: 0.0204, true error: 0.18519999999999998
k = 8 empirical error: 0.021200000000000007, true error: 0.18480000000000005

1) First, for every **even** K, Adaboost does not seem to improve almost at all from the K-1 iteration (e.g. k=2 gives almost identical or worse results to k=1, same with k=4 compared to k=3 and so on…)

   Second, after observing the results of each Adaboost run separately, it seems that there are runs that are much better than others and some are much worse.
   For example, a specific run had with k=7 an empirical error equal to 0 and true error equal to 0.06.
   We could choose the top runs out of the 50 and exclude the others to gain a better average.

   Third, the above conclusion raises a question, can we find a separation of S and T to get good results? Instead of splitting randomly of course.

2) Do we see Overfitting? Yes, we do.
   The larger the K, the smaller the empirical error, almost 0 for k>=3, yet the true error does not improve by a lot from k>=3.
   So, the VC dimension is large, the empirical error is very small and yet the true error is not satisfying, the only conclusion is Overfitting.