CMPT 404
Midterm

Daniel Clark

October 13, 2016

1 Compare two algorithms on a classifications task: the Pocket algorithm (designed for classification), and linear regression (not designed for classification).

I ran the Pocket, Linear Regression, and hybrid Linear-Pocket algorithms on the given data set, terminating after 100 unimproved iterations. On average, the Linear Regression algorithm had the largest amount of classification error $E_{out} = 0.0154985598683$. The standard Pocket algorithm came next with an error of $E_{out} = 0.01261829653$. The hybrid Linear-Pocket algorithm had a similar error $E_{out} = 0.0123439857358$. Linear Regression is not designed for classification, and so, it should perform worse than algorithms tailored to that task. As these data are so noisy, the pocket algorithm and the hybrid algorithm performed similarly. This makes sense because even though the hybrid algorithm starts with a better guess, they both must still iterate many times before coming up with a sufficient solution, and these iterations decrease the improvement created by the hybrid algorithm.

The standard Pocket algorithm took 428 iterations to complete while the Linear-Pocket algorithm took 478 iterations. These numbers are interesting because they suggest that the standard Pocket algorithm starts with a better guess than the linear regression solution. I suspect, however, that the discrepancy here is merely due to randomness in the Perceptron algorithm.

2 For an H with $d_{vc} = 10$, what sample size do you need (as prescribed by the generalization bound) to have a 95% confidence that your generalization error is a most 0.05?

I created a short program to solve this problem by iteratively approximating it using equation (2.13). This program completes quickly and seems to approach the proper value exponentially quickly. It quite simply computes the equation using the most recent guess for N and compares that value to the previous value of N until they are equal. Hypothetically this would run forever, but due to the limited storage method used, there is a point where the improvements made are smaller than the least significant digit stored, at which point the algorithm terminates.