Intermediate Status Report

CS 6350-001: Machine Learning, Fall 2018

James Brissette

My initial two runs used the perceptron algorithm and the averaged perceptron. This proved to be rather challenging as the size of the data set became a limiting factor in terms of execution speed. Fortunately, however, during my initial implementation of perceptron at the beginning of the semester I created a useful method of dealing with sparse matrices and loading in the data appropriately.

I found initially that since our feature space in the perceptron homework was much smaller (~19 or so) that I had implemented full feature vectors, which of course had to be changed to deal with this new data set. This, however, introduced a new issue and that was that my implementation of the perceptron algorithm was perhaps not as fast as it could be. This taken in combination with now needing to add and manipulate features with a dictionary made working with the full dataset rather cumbersome.

Immediately I saw that it was not feasible to iterate over all 25000 examples for 20 epochs and so in order to test and optimize, I used only small subsets of the data. Noticing that the algorithm was taking a great deal of time to go through each example, I thought it might be a good idea to also augment the feature space. While I haven't yet developed a good set of feature transformations, this is worth mentioning in this intermediate status report.

The accuracies I achieved for each algorithm are below, and I likely will attempt to implement various forms of ensembles in concert with simple feature transformations. I imagine I will also need to refine my algorithm for the Average Perceptron to speed up execution time by finding a more clever way to manage the average weight vector, as I may likely encounter similar storage and speed issues with other algorithms.

Simple Perceptron
Number of Epochs: 1
Accuracy on Evaluation Set: 0.84064

Averaged Perceptron
Number of Epochs: 1
Accuracy on Evaluation Set: 0.63727 @ Rate = 1; 0.62416 @ Rate = .1