Methodology

I began by modifying scale\_all\_feature\_values.m to scale all features on a 0-1 scale using a linear profile, i.e. . Rescaling the values like this meant that all features would have equal weight in the clustering space.

Next, I altered the method of seeding clusters. Initially, clusters were seeded with the first stocks in the list. I seeded the clusters randomly, checking to make sure that no pattern was seeded into multiple clusters. I also increased the number of clusters to 10 and the maximum number of passes to 50.

I wrote cluster removal code based on the cluster adding code. At this point, the utility functions of the code were completed. To test my code, I used a random cluster selector in find\_closest\_cluster.m and checked that patterns were in fact being added and removed from clusters.

Having confirmed that the utility functions were working, I modified find\_closest\_cluster.m to check the Euclidean distance between patterns and cluster centroids and select the cluster with the shortest distance.

I chose to invest in all stocks from clusters with an average return on investment greater than twice the average and a cluster population of five or greater. Choosing populations with greater than twice the average return on investment eliminates clusters that deviate only slightly from the average. Requiring that cluster populations be greater than five helps to eliminate clusters without large enough populations to have predictive power. To evaluate my picks, I calculated the average return on investment of all stocks in the validation set that were in my chosen clusters.

At this point, I was still clustering based on all of the features, and I was using ten clusters. This combination resulted in very poor clustering, and the average return on investment was only marginally better than the average (see figure 1), about 5% vs 4% for the full set.



Figure : Clustering into ten clusters, using all features.

My next move was to increase the number of clusters to 32, which I reasoned would still allow for ~8-9 training stocks per cluster. Although a smaller number of clusters might be just as affective, having more clusters would mean that each cluster was composed of more similar stocks, and hopefully the cluster would be a better predictor of profitability. Increasing the number of clusters substantially improved both the homogeneity of the clusters and the return on investment of my picks. This method resulted in about 11% return on investment.



Figure : Clustering into sixteen clusters, using all features.

Although this result was encouraging, I hoped that I could improve it further by eliminating some features from the calculation. I went through and clustered based on each feature individually to observe which ones might have a negative impact on clustering.