Edward Venator (esv)
EECS 484 Problem Set 1
9/12/2011

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. $x' = \frac{x - x_{min}}{x_{max} - x_{min}}$. 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. Increasing this factor to greater than two favored outliers, which introduced volatility into the results—sometimes the picks were very profitable, and sometimes they did not even beat the index 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.

Edward Venator (esv)
EECS 484 Problem Set 1
9/12/2011

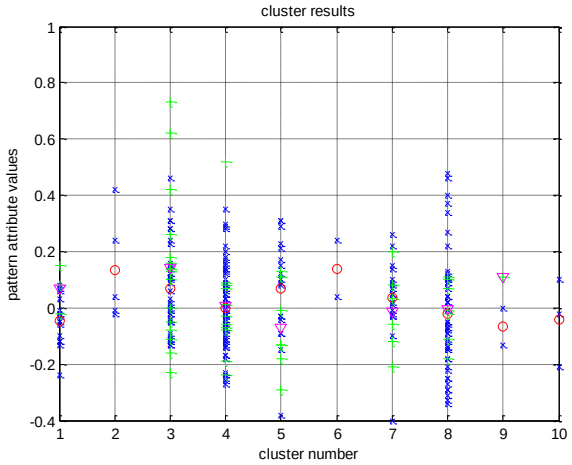

Figure 1: Clustering into ten clusters, using all features.

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 a best-case 11% return on investment.

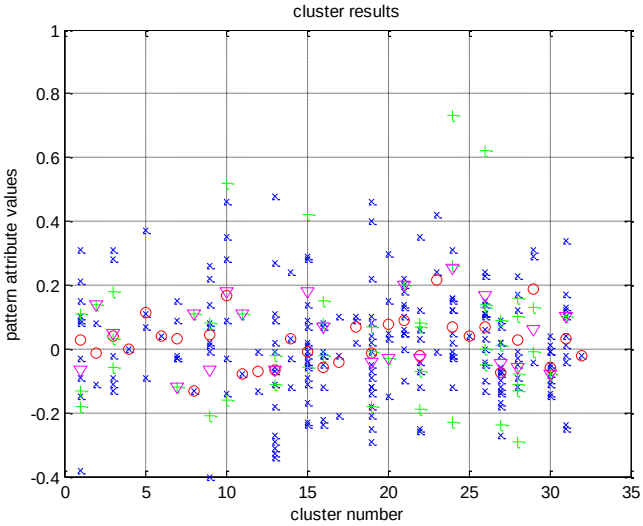

Figure 2: Clustering into thirty-two clusters, using all features.

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. I found that features 17, 20, and 21 (LT Debt to Equity, Sales/Turnover, and Price/Book Value) resulted in unprofitable picks if used individually

to choose stocks. Based on this, I eliminated them from my clustering feature set. This method turned out to be rather hit or miss. Some trials resulted in a return on investment as low as 7%, and some resulted in one as high as 19%. The reason why is clear from the plot in figure 3, which shows the wide spread of the clusters and that the validation set does not appear to match well with the training set.

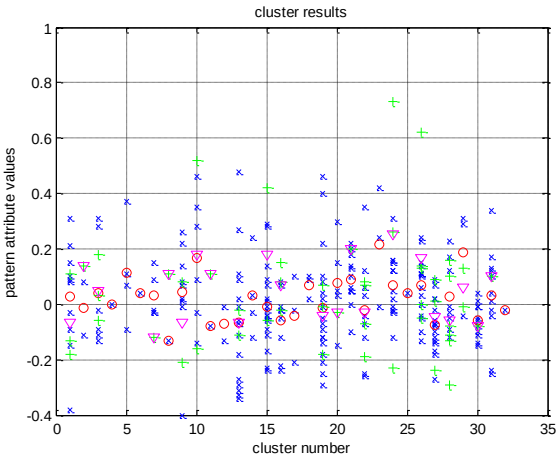

Figure 3: Clustering into thirty-two clusters, using some features.

I decided to keep experimenting with which features to include. I wouldn't be able to evaluate all 255 possible combinations, but I could try quite a few. I determined that feature 17 (LT Debt to Equity) was absolute poison to good stock picks. Feature 20 (Sales/Turnover) seemed to improve my picks when I added it to the features to check. Feature 21 (Price/Book Value) did not seem to make picks better or worse on average, but greatly decreased consistency, leading me to believe that it has little predictive power and was just adding noise. Feature 15 (P/E Close) did not appear to make much difference at all whether it was included or not.

Results

Edward Venator (esv)
EECS 484 Problem Set 1
9/12/2011

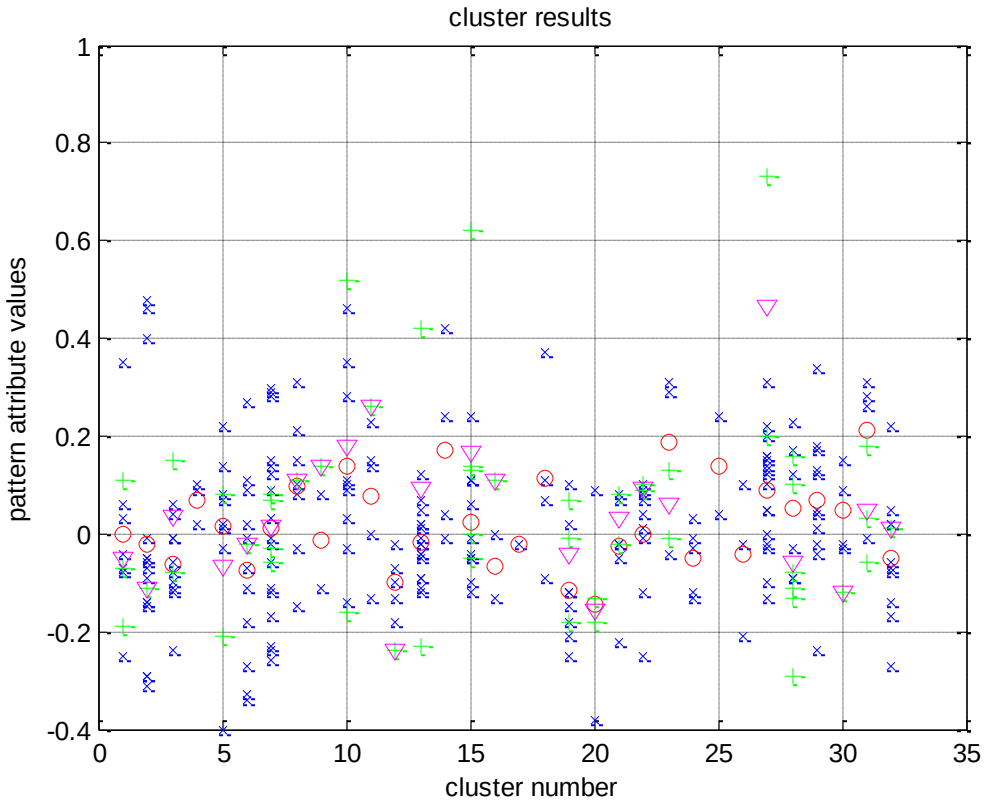

Figure 4: Typical clustering results with my algorithm.

I decided to cluster based on P/E, Price/Sales, Net Profit Margin, Return on Equity, Sales/Turnover, and Operating Margin. The following are the results of fourteen runs of this clustering method.

Table 1: Results of 14 trials of the final clustering algorithm.

Trial	Num Passes	Num Stocks Picked	Return on Investment
1	6	23	8.70%
2	6	15	9.53%
3	6	23	7.61%
4	7	11	9.55%
5	8	7	5.71%
6	5	15	15.53%
7	5	8	16.75%
8	6	18	5.61%
9	11	23	8.26%
10	9	19	11.05%
11	9	12	11.17%
12	6	28	7.36%

13	7	21	7.29%
14	11	13	19.08%
Average	**7.286**	**16.857**	**10.23%**

As you can see, the results were fairly consistent, and averaged about 10% return on investment, much better than the ~4% of the index fund. There were some outliers, but even the worst outperformed the index fund. This performance is not notably better than the performance of the same algorithm with all features included.

This method consistently converges in fewer than 12 passes, which means that my fifty pass limit was far higher than it needs to be. However, the run time of the software is acceptable, so I see no reason not to let it run to convergence.

Conclusions

This method is more effective than an index fund. However, I am not entirely satisfied with the loose groupings in the clusters. If the clustering method were a better indicator of stock profits, the clusters would be tightly grouped on the attribute plot, and they are not. In addition, a better clustering method would produce a more consistent return on investment. This method seems to be effective at picking better than average stocks, but it's no get rich quick scheme.