

---

# Hierarchical Confidence Based Clustering

---

**Richard Otero, Shane McDaniel, Charles Pippin**

College of Computing

Georgia Institute of Technology

Atlanta, GA 30332

{rotero, shanemcd, cepippin} @cc.gatech.edu

## Abstract

In order to ease the complexity of asking one learner to accomplish the task of prediction, multiple learners are often used. Our approach towards this is to split the data set into clusters based on the confidence a learner can classify them. This repeats until some predetermined hierarchical level. Each cluster of data then gets its own specialized learner. This will allow the algorithm to concentrate more on the data points that it is unsure of, resulting in a higher confidence rate on them. We then use a centroid and neural network to learn which data points go into which clusters such that new data points can be matched up with the best classifier. We have applied this approach to the problem of financial classification for the Dow Jones Industrial Average, as well as handwritten digit recognition. Our hypothesis is that this approach can be used to create a committee of learners, each specialized towards a subset of the data space. Our results show that we can improve upon traditional strategies for each data set, while offering the ability to parallelize learning.

## 1 Introduction

The task of training a neural network to learn a large complex data set is challenging from the possible interactions between separate components. While it may be possible for the network to eventually learn the data set there is no practical way to know how long the process will take, which then makes validation of ones method difficult and time consuming. A common approach is to make such learning easier is to divide the data set into logical groupings. We expect the groupings to have more obvious trends, allowing neural networks to train more quickly and accurately. Determining these divisions by hand can be difficult, however, especially if the data is of a high dimension. Additionally, determining them by hand requires additional analysis for each data set. We strive to create a method of automatic clustering based on a particular data set's hidden features, as determined by a neural network.

A large issue with financial data is determining what trends exist. Given the vast amount of information that is available about a single stock, this proves to be a very cumbersome task. Additionally, trend knowledge is not necessarily transferable; what is a trend to one stock may be nothing to another. We expect that our system will be able to discern the relevant features of the financial data.

## 2 Methods

The general framework for our system consists of three steps as shown in Figure 1, and consists of three steps. First, the data set is fed into the splitter which groups together data points based on how well a neural network can classify them. This process is repeated on each resulting cluster until a threshold is encountered. The resulting clusters are then assigned a unique neural network which learns the subset to a high confidence level. We then have a set of learners, each of which is specialized for a given cluster.

The second step is to determine how to classify a new data point as belonging to one of the clusters. We consider two different methods: calculating the centroid of each cluster and then calculating the distance from that center, and having a neural network learn the cluster assignments.

The final step is to predict the value of a data point. For the centroid method we employ a distance metric against the center of each cluster and use that as a weight. For the neural network method we take the cluster it most confidently assigns the point to.

For our neural networks we employed the Joone Open Source Java package [5]. The overall algorithm and simulation experiments were implemented by the authors.

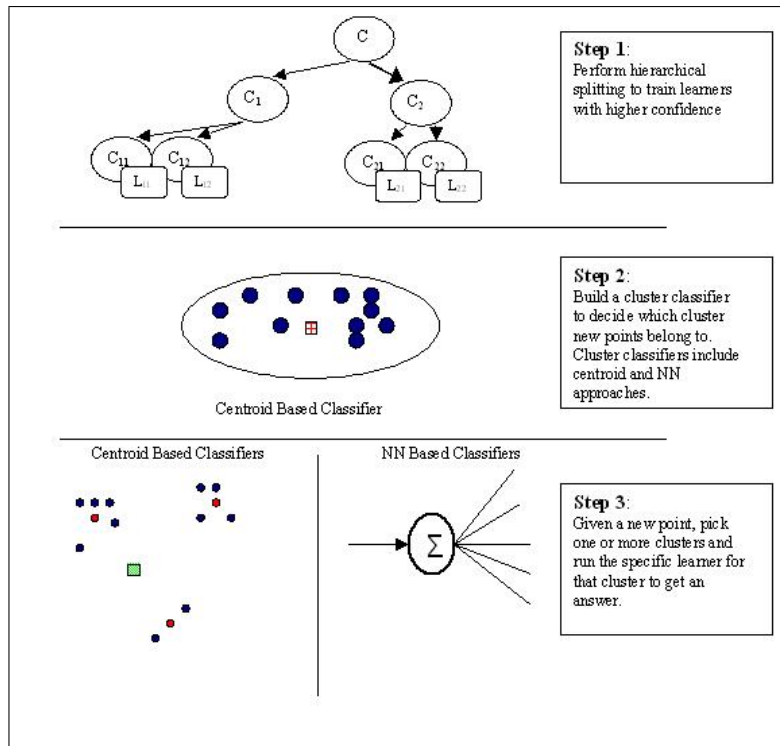


Figure 1: Splitting the data into clusters

### 2.1 Splitter

The splitter is used to generate a number of unique subgroups from the original set of points. This is done by developing a weak hypothesis that is able to correctly classify a subset of the data. This subset is assumed to have a relationship that can be leveraged by

the classifiers. The correctly classified points are placed into one group and the remaining points are used to generate another simple hypothesis.

The split from the second simple hypothesis generates another unique group of correctly classified points and an incorrectly classified points. This cycle is continued until a tree is generated of a predefined depth. The leaf-nodes of this tree are the groups of instances that were pulled from their parent data set by a simple hypothesis.

Each of the leaf-nodes is then trained separately to generate their own high accuracy hypothesis over the cluster of data points on which it was trained. As such, the final hypotheses each have a high confidence over their learned sets.

These clusters of related points and their associated hypotheses are then passed to the classifiers. It is important to note that we have used one measure, the ability to be easily classified correctly, as the metric of similarity between two points.

By removing these similar points from the set, relationships within the more difficult group left behind may become more pronounced, as a new round of training begins. The larger complex problem of financial prediction has been broken into smaller sub-problems.

## 2.2 Classifiers

The purpose of the classifiers is to learn the data points belong in each cluster. That is, to determine some trend in each cluster of points such that we can most accurately determine the cluster a new point belongs in. We we experimented with two classifiers in our system: a centroid based classifier and a neural network based classifier. Both performed well.

### 2.2.1 Centroid

Given that each cluster contains a specific subset of points, the learner associated with each cluster will have been trained on a subset of the sample space. Intuitively, this subset is defined by characteristics that can be captured using a similarity metric. By applying such a metric to the new point, our hypothesis is that the appropriate learner relevant to that point can be selected.

As such, after the points have been partitioned, the centroid approach treats each set of points as a cluster and calculates the centroid as a function of the mean across all dimensions. When new points are presented for classification, the distance from the new point to each cluster centroid is calculated. A number of different similarity metrics were explored across both data sets, including Euclidean distance, hamming distance, mahalonobis distance and the cosine of the angle.

Each cluster in the list then returns a classification result by querying the learner trained on the points in that cluster. Rather than selecting the closest cluster we enlist a weighted voting mechanism. The weight applied to the vote is a function of the training accuracy of the cluster's learner and the inverse of the distance,  $d$ . Each weighted vote is then summed. The classification with the most votes is the answer returned, see Equations 1 and 2.

$$answer = \sum \alpha \frac{1}{d^2} Ask(L_i) \quad (1)$$

$$\alpha = trainaccuracy \quad (2)$$

### 2.2.2 Neural network

The neural network classifier uses a standard neural network with one hidden layer of nodes. This hidden layer was typically set to contain half the number of nodes as the input

layer. The output layer consists of one node for each cluster. This output layer method was used instead of a single normalized output as it showed to give an accuracy increase on the order of 20% on the handwritten digit data set. A validation set of data was chosen by randomly picking 20% of the data set, the other 80% was used as training data. The network was then trained for 300 epochs as this value resulted in accuracy increases on the order of 20% over our initial value of 30 epochs. During the training process the network with the best validation accuracy was remembered and ultimately used for classifications.

## 2.3 Data sets

### 2.3.1 Optical recognition of handwritten digits

The optical character recognition data set contains digits ranging from 0 to 9 and consists of normalized 32 x 32 bitmaps of handwritten digits (gathered from 43 people) transformed into 8x8 input matrices. This data set was created by E. Alpaydin, and C. Kayna[6].

### 2.3.2 Financial data

The financial data set consists of thirteen years of historical financial data for the Dow Jones Industrial Average as downloaded from the Yahoo Finance Website [7]. The original data was transformed to pull time series information into each row. In the transformed data set, each row consists of input values as presented in Table 1. Two output values, increase and decrease, were used. These were derived from the change in price at day t+1 (closing price the following day.) The data set was split into separate training and testing sets with a 2/3, 1/3 distribution.

Table 1: DJIA data attributes

Attribute	Description
t-20	percent change in price from 20 days prior
t-10	percent change in price from 10 days prior
t-2	percent change in price from 2 days prior
t	percent change in price from day before
swing	difference between high and low price that day
volume	volume that day

## 2.4 Simulation

To provide a benchmark for our algorithm, we created a simple trading simulation that executes over a period of 1200 trading days ranging from October, 1999 through April, 2004 for the DJIA. The simulation exercises different trading strategies, including mean-value, random, and buy-hold. The mean-value trading strategy assumes a mean as a function of volume and price and trades around that value. The random trader simply selects from buy, hold and sell choices; the buy-hold trader simply buys in 100 share increments until no account funds remain. Each of the centroid classifiers (using different similarity metrics) and the neural network classifier participated as traders. A typical back propagation neural network was used as an additional benchmark trader. Each trader started with an opening account balance of \$100M. At each day in the simulation, all traders were given the option to buy, hold or sell 100 shares at the closing price for that day. On the final day of simulation, all shares were sold at market price.

### 3 Additional considerations

The immediate problem with adding more steps into the process of classification is that we are also adding additional complexity. The first concern is the number of settings and thresholds that are used to split the data set on. One issue with splitting data is that there really is no way to validate the generated clusters by themselves; they can only be evaluated in the scope of the larger system. This leads us to the classifiers. Each classifier has its own set of variations. In the centroid classifier, we have different distance metrics to consider; in the neural network classifier we have more settings and thresholds. Each variation we test for compounds the number of combinations needed to validate the system as a whole and the clustering in specific. Just varying the learning rates for the splitter and the neural network classifier gives a squared increase in testing combinations.

## 4 Results

### 4.1 Data sets

#### 4.1.1 Optical recognition of handwritten digits

For the handwritten digits data set our classification mechanisms performed adequately, but on the same level as a single neural network. The centroid classifier had its highest accuracy at 76.5%, and the neural network classifier had its highest accuracy at 76%. Both methods were comparable in performance. As a benchmark we ran a single neural network on the entire data set and achieved an accuracy of 74.2%. Both methods were able to beat the single neural network.

Table 2: Financial method gains

Method Name	% Gain
Random	2%
Buy & hold	1.3%
Mean value	22.6%
Centroid Classifier	23.3%
Neural Net Classifier	12.8%

As a benchmark, we compared our machine learning prediction method to a couple of traditional methods. All accuracies are shown in Table 2. We see that the mean value algorithm does the best among these methods and sets a high achievement bar. As for our classifiers we see promising results. The centroid based classifier peaked at 1.233 times the initial \$100M investment. This beats the mean value algorithm by 0.7%, or an additional gain of \$700,000 over mean value, as shown in Figure 2. The return rate for the centroid however ranged greatly with a minimum value of 0.808. The neural network classifier did not surpass the mean value algorithm; however, it did result in a max gain of 12.8%. The variation in the neural network's return was small, with the smallest return being 7.0%, it still achieves a profit.

## 5 Related work

Ada-Boosting (AB) is related in its conception of developing a large number of different hypotheses which are all offered a vote in deciding the final classification of a new point. The original data set has a hypothesis developed for it that captures slightly better than 50% of the points or instances. This initial hypothesis is stored and the weighting of each

Table 3: Financial classification accuracy

Method Name	Accuracy
Simple NN	91.7%
Centroid Classifier	98.3%
Neural Net Classifier	98.3%

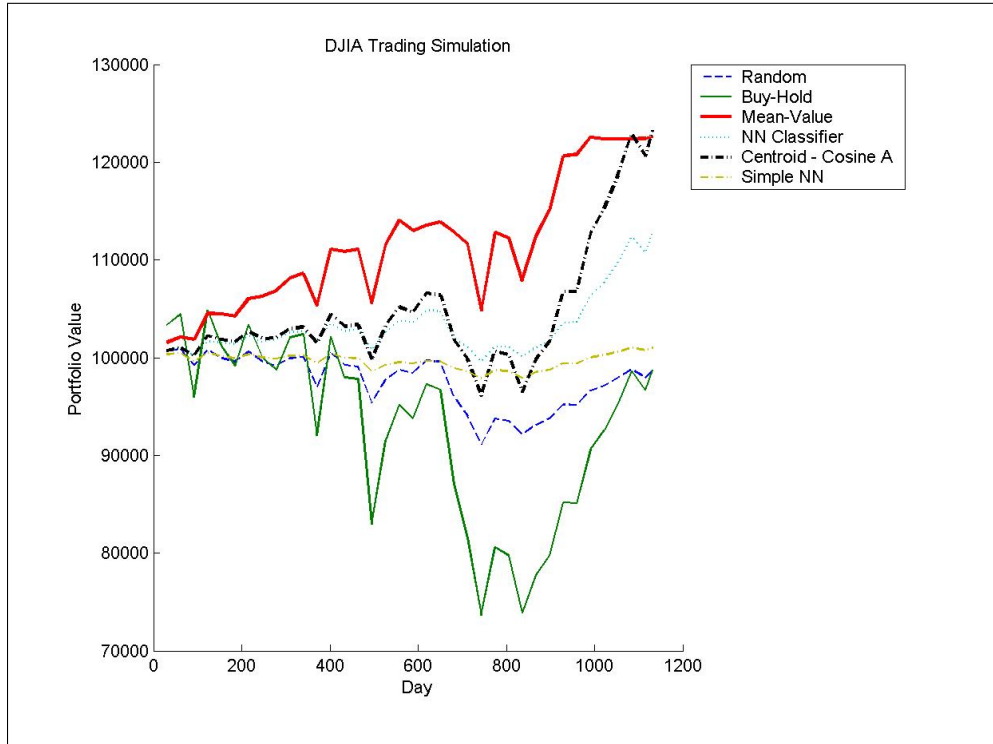


Figure 2: Financial Trading Simulation

point is adjusted accordingly. This weighting affects the importance a future hypothesis will place in classifying the data set. The correctly classified instances are given a lower relative weight and the incorrectly classified instances have their relative weights increased.

This method adjusts each successive hypothesis to focus on the incorrectly classified instances, to a greater and greater degree. In AB, all of the points from the original data set are always present, even if their relative weights become vanishingly small. This is important as every hypothesis is trained within the domain of all of the original points, though a subset of those points have been given lesser weights. When a set threshold is met, a committee of found hypotheses is used to vote on the classification of new points.

Conceptually, our design follows the ideology of AB: to focus on what we were not able to correctly classify. The differences with the binary classifier is that the data set is initially split into smaller groups. This is based on the concept that points correctly classified from the larger group, by a simple hypothesis, have a similarity that can be exploited for classification. These groups are pulled from the original set before a final high confidence hypothesis is developed for each. These final hypotheses are not trained within the domain

of the original data set, as in AB, but in the subset discovered through the use of a simple initial hypothesis. The final hypotheses are collected and each allowed a vote based on the probability that a new point belongs within its subset.

The committee approach has been used before for financial prediction. It has been shown to be useful in reducing a financial models complexity and for improving performance. Prior work [8] has been done in splitting the full data set into smaller clusters that could be used to form a committee of subset experts. In their work, all of the input variables were first grouped on the basis of mutual information. Statistically similar variables were then assigned into the same cluster.

Our work approaches the problem by means of a simpler heuristic that is shown to perform well on challenging data sets, when compared to other common techniques for financial prediction.

## 6 Future work

Future work should be done to examine the effects of different methods for generating the initial cluster groups, for the splitter. Variance could be used as another measure of similarity for finding related points. One could also use higher order statistics to generate clusters based on how interesting, kurtotic, they are.

Apart from the research that could be done to develop heuristics for generating the initial clusters, additional methods should be examined for deriving the distance from a new point to each of entries in the cluster list. Principle Component Analysis (PCA) could be used as another method for deriving cluster distance, by generating distance measures along a new basis of principal components. Independent Component Analysis (ICA) could also be used to only consider distances to each cluster that exist along a new basis of interesting components; ignoring contributions to the distance measure that may simply be adding Gaussian noise.

Work should also be done to examine these techniques in the form of a ternary splitter. Instead of two groups, the full data set would be split into three groups; confident correct, confident wrong and unconfident. The confident wrong and unconfident groups could then be worked with or continually split until a group of correctly classified clusters are found. Initial work showed improved results from using a binary splitter but this is still an open area for continued research.

Testing should also be performed against implementations of other committee-based financial forecasting techniques [8] to examine the heuristic's simulated performance, relative to probabilistic and other methods of separating initial clusters of similar points.

## 7 Conclusions

Many straight forward and simple methods exist to solve the problem of classification for stock trading; some of these perform surprisingly well. Our results show that our method of splitting a data set into confidence based clusters and then learning the cluster assignments can improve upon traditional methods methods, resulting in larger financial gain. Additionally, we have shown that there are different classification options available based on the desired risk to gain ratio.

While the centroid classifying method worked better than the neural network classifier on both data sets an interesting trend arose. On the financial, data the centroid classifier beat the mean value method however it had a return range of -21.2% to 23.3%, meaning that in some instances it lost money. The neural network classifier, on the other hand, has a return

range of 7.0% to 12.8%. This provides us with a nice risk to gain trade off.

In addition to our promising results, we have also provided a method to make the job of learning a data set tractable, allowing for parallel learning. Given the nature of how many different attributes are available for any particular stock, in conjunction with the amount of stock history available, training a single network to learn all the data could be impractical. Through the use of parallelization, this approach produces a more efficient method.

Our method inherently adds additional complexity to the problem of classification. It is our belief, however, that simple methods are not always useful for classification. Our experiments show the gained accuracy in financial results can be worth the additional complexity in some cases. Additionally the added complexity is shown to increase efficiency by opening up the door to parallelization.

### **Acknowledgments**

We wish to thank Dr. Charles Isbell, in the College of Computing at the Georgia Institute of Technology, for his guidance and input.

### **References**

- [1] Ghosn, J. & Bengio, Y. Multi-Task Learning for Stock Selection.
- [2] Yao, J. & Tan, C.L. *Guidelines for Financial Forecasting with Neural Networks*.
- [3] Walczak, S. (2001) An Empirical Analysis of Data Requirements for Financial Forecasting with Neural Networks. In *Journal of Management Information Systems 17, No. 4*
- [4] Wu, L. & Moody, J. Multi-effect Decompositions for Financial Data Modeling. In *NIPS Proceedings, 1995*.
- [5] Joone. Java Object Oriented Neural Engine. At <http://www.jooneworld.com/>
- [6] Alpadin, E. & Kaynak, C. Data set: Optical Recognition of Handwritten Digits. At <http://www.ics.uci.edu/mlearn/MLSummary.html>.
- [7] Data set: Dow Jones Industrial Average. At <http://finance.yahoo.com>
- [8] Yuansong Liao and John Moody Department of Computer Science, Oregon Graduate Institute (1999) Constructing Heterogeneous Committees, Using Input Feature Grouping: Application to Economic Forecasting, *NIPS Proceedings, 1999*