Time-Series Clustering and Association Analysis of Financial Data

Todd Wittman
CS 8980 Project
December 15, 2002

Abstract: *Each stock sold on the New York Stock Exchange is classified by industry. This paper describes applying data mining techniques to reach two basic goals. First, we wish to determine the industry classification given the historical price record of a stock. To achieve this, we experimented with hierarchical agglomerative clustering and a feed-forward neural network. Issues with outliers, pre- and post-processing, and distance measures are discussed. Second, we wish to determine the relationships among the various industries. We applied association analysis on the Dow Jones industrial indices to generate rules describing the stock movement across industries. We conclude the paper by discussing possible improvements and areas of future research.*

1. Introduction

Data mining of financial data has proven to be very effective and very profitable [1]. The goal of this project is to apply data mining techniques to an interesting, albeit not very lucrative, area of finance. Every stock sold on the New York Stock Exchange is classified into an industrial category, or just called an "industry." A company is identified with an industry based on its primary activities, which usually means the area from which it derives the largest share of its revenue. These categories include Chemicals, Biotechnology & Drugs, Retail, Healthcare, Utilities, and so on. In addition, each industry is further divided into sub-categories. For example, the Media & Advertising industry consists of four sub-categories: Advertising, Movies & Music, Publishing & Printing, and TV & Radio. However, these sub-categories are not standardized. In this paper, we will work with the standard industrial categories, as given by [2].

Two natural problems arise from industrial categories: classification and association. We wish to develop methods that can answer the following two questions:

1. *Can we determine a stock's industrial category given a historical record of the stock's prices?*
2. *How are the movements in stock prices across the various industries associated?*

In Section 2, we describe clustering and neural network approaches for answering the first question. In Section 3, we describe association rule mining to answer the second question. We conclude in Section 4 by discussing the merits of each approach and areas for future research.

2. Time-Series Clustering

Our goal is to properly identify the industrial category to which a stock belongs given only the historical price record. Gavrilov, Anguelov, Indyk, and Motwani accomplished just this by using hierarchical agglomerative clustering [3]. For comparison, we refer to [3] throughout this paper, pointing out where our approach differs from the techniques of Gavrilov et. al. It should be noted that this problem is not just an exercise in clustering. There are important applications to finance and interesting conclusions that can be drawn about specific stocks. Stock clustering has applications to quantizing the effect of trends within and between industries, identifying misclassified stocks, and portfolio evaluation.

Any attempt at clustering the stocks is based on the following crucial assumption:

A time-series clustering will be valid if and only if the price fluctuations of stocks within a group are correlated, but price fluctuations of stocks in different groups are uncorrelated or not as strongly correlated.

This statement can be interpreted two ways. The forward version of this statement says that if we obtain a good clustering with respect to our distance measure, then stocks tend to move as a group. This implication in and of itself would be useful to the financial world. Moreover, clustering statistics can quantize to what extent stocks move as a group. External clustering statistics such as entropy, purity, and cohesion tell us how closely stocks within an industry resemble each other. Internal statistics such as separation and the silhouette coefficient can tell us to what degree the industries' behaviors are separate from each other. The reverse version of the assumption says that if stocks tend to move as a group, then we should be able to obtain a sensible clustering. Without this assumption, any clustering would be meaningless even if the clustering was perfect with respect to the chosen distance measure. Also, we need to see differentiation between the groups or else our clustering will fail. This also points out a possible application of financial clustering. Since our clustering is based solely on the historical price record, the clustering will determine which group the stock most behaves like, which is not necessarily the group the NYSE categorized it as. The stock market is based on perception, not reality. Perception has become even more important in recent years with the advent of on-line trading, which released a flood of anxious and often ill-informed day traders. For example, AOL Time Warner is categorized by the NYSE in the Media & Advertising group, because that is where it earns the largest portion of its revenue. However, if most investors perceive AOL Time Warner as being an Internet stock, then its price fluctuations will track with those of the Internet group. By comparing the results of our clustering to the groupings given by the NYSE, we can actually learn more from the misclassifications than from those that match the NYSE standard. Also, by examining class statistics such as purity and entropy, we can say to what degree a stock follows each group. For example, we might determine that AOL Time Warner's behavior is 20% a Media stock and 80% an Internet stock. Clustering could also be very helpful in analyzing the time series of a portfolio including several stocks. The behavior of an investment portfolio is not necessarily determined by the stock that makes up the largest monetary share of the investment. Clustering a portfolio with stock data could identify which industrial groups have the greatest influence on the portfolio.

2.1. Distance Measures

Let $s_i(t)$ denote the price in dollars of stock i at time t. During trading hours, the price of a stock is constantly fluctuating, so time is a continuous variable. For our purposes, we can treat time as discrete since we will only look at the adjusted closing price for the day or week. Since the price of a stock does not necessarily reflect the revenue or size of a company, we cannot compare stock prices directly. Also, it is difficult to look at deviations, or first differences, of prices due to the wide range of possible stock prices. We assume that the percentage change is a good comparative measure of stock performance at a fixed time. This assumption is valid, because if it were not then the stock market would be biased. For example, if low-priced stocks showed a larger percent change on average than high-priced stocks, then no one would ever buy high-priced stocks. To say that another way, suppose Stock A cost $100 per share and Stock B cost $1 per share. We cannot say which stock is better based purely on their price. Also, the deviation is not a consistent measure, because if both Stock A and Stock B went up $1, then Stock B would be a better purchase since we could with a fixed amount of money we could buy 100 times more shares of Stock B than Stock A. However, if both Stock A and Stock B went up 10%, then they would be equally good stocks. We should note that [3] worked on the price deviations, but they performed extensive normalization on their data to correct for different prices. Letting t+1 denote the increment of time period, we have that the percent change $P_i(t)$ of stock i is

$$P_i(t) = \frac{s_i(t+1) - s_i(t)}{s_i(t)} \times 100.$$

This is only a good comparative measure of performance at a particular time t. We need to develop a distance measure to comparatively measure the time-series.

The natural and most basic distance measure for a time series is the Euclidean distance, or L2-norm. Then the distance between stocks i and j is given by:

$$d(i,j) = \sqrt{\sum_t \left(P_i(t) - P_j(t)\right)^2} = \left\| P_i - P_j \right\|_2$$

However, this distance measure is sensitive to noise and outliers. It should be noted that outliers may pose an issue for financial data, since we may see an abrupt jump in stock prices following a merger, bankruptcy, or scandal. Also, we know the Euclidean distance shows less differentiation as the dimension increases. Since we will be looking at price fluctuations over a long time period, we can expect the dimension to be very high. The authors in [3] used the Euclidean distance, but the bulk of their paper was reduced to reducing the dimensionality of the data.

Another possible time-series measure is one based on predictive models, such as ARIMA [4]. However, an underlying assumption in ARIMA time-series is that the series contains some pattern or cycle. Hence, the authors in [3] were able to successfully cluster data involving temperature, US population trends, and physiological conditions. However, the stock market is inherently unpredictable, so a similar approach based on moving averages might not be appropriate.

One of the difficulties in clustering stock market data is that the market tends to move as a whole. That is, we assumed that stocks move as a group, but they also move on a larger scale as an entire set. To see the differences in groups, we wish to remove the overall movement so

that we can concentrate on the subtler trends. One way to do this is to normalize the data. There are two basic approaches to normalizing stock market data: stock-based normalization and time-based normalization. In stock-based normalization, we would calculate the mean percent change μ_i and standard deviation σ_i for each stock i. We then normalize the percentage change $P_i(t)$ by

$$\tilde{P}_i(t) = \frac{P_i(t) - \mu_i}{\sigma_i}.$$

In time-based normalization, we instead use the mean percent change $\mu(t)$ for a time interval, say a week, across all stocks and the standard deviation $\sigma(t)$ for that time period:

$$\tilde{P}_i(t) = \frac{P_i(t) - \mu(t)}{\sigma(t)}.$$

Our preliminary experiments with hierarchical agglomerative clustering indicate that time-based normalization is superior. In a sense, this normalization removes the overall trend of the stock market. As an analogy, if our time series represented the daily temperature over the course of a year at different locations on the earth, time-based normalization would help remove the effect of the seasons that would be present in all the time series. Unless otherwise noted, all data discussed in this paper was normalized in this manner. The authors in [3] also used this time-based normalization approach, although their "piecewise normalization" involved time intervals, or "windows," of varying size. The authors did not discuss how they determined these "windows."

Another possible pre-processing step is to attempt to reduce the dimensionality of the data. The authors in [3] experimented with three basic techniques: Principle Component Analysis, Fourier Transform truncation, and aggregation. Principal Component Analysis (PCA) uses Singular Value Decomposition (SVD) to determine the eigenvalues and eigenvectors of the covariance matrix. These eigenvectors, also called principal components, form an orthonormal basis which is used to map the data vectors to a lower-dimensional vector space. The second technique uses a Fourier Transform to convert each vector from the time-scale space into the frequency-scale space. Each vector is then truncated, so that each time-series is represented by only a few of it lowest frequencies in the Fourier space. The third approach, aggregation, is by far the easiest to implement and probably the most effective dimensionality reduction technique. Each set of values over a specified time period is replaced by its mean value. Theoretically, we could measure stock prices to the second. However, the daily closing price is probably the easiest statistic to obtain. Although this price does not represent the daily average, we could look at instead the weekly or monthly average closing price. Gavrilov et. al. chose to work with the average over every 10 day period. Although averaging is the simplest and most natural data compression approach, this issue of the time scale needs to be handled carefully. For example, suppose we gathered the hourly temperature in Minneapolis over the course of one year. If we wanted to detect the trend of the seasons, it might not be obvious from the hourly data due to local fluctuations (heat waves, cold snaps, rain, etc.). If we were to compress the data set down to the monthly temperatures, then we could very easily pick out the seasons by seeing the cold January moving into the hot July and back again. However, we have compressed our entire year's worth of data down to just 12 data points and we may have erased any subtler trends, such as a heat wave in August, that may have occurred in the year. If the time scale is too small, the computational costs become burdensome and it is more difficult to recognize global trends. If

the time scale is too large, we may have erased local trends and these local differences may be important to properly differentiating the clusters.

2.3. Clustering Results

Using the comprehensive historical stock price record available on-line at [2], we gathered data for 91 different stocks. The stocks are listed in Appendix A. The stocks covered the three year period from November 1, 1999 to November 1, 2001. To reduce the dimension of the data, we looked at the weekly closing price. This aggregation is valid because financial experts generally look at the weekly prices when judging the long-term trends of stocks. Using the industry classifications given by [2], the 91 stocks span 10 industrial categories: Aerospace & Defense, Automotive, Chemicals, Energy, Financial Services, Computer Hardware, Health Care, Internet, Media & Advertising, and Retail. We will view these classifications as the "ground truth" and use them as the standard to evaluate our clusterings. These specific ten clusters were chosen with the hope that there is some interdependence, but not to the point that it becomes too difficult to differentiate the stocks. For example, we chose Computer Hardware and Internet as two categories, but we did not include stocks from the Computer Software industry as well. The idea is that Hardware and Internet companies exert some influence on each other, but not to the extent that Software and Internet companies are related. In most cases, the stocks picked were the ten largest companies in the industry, as judged by market capitalization. However, three years of data was not available for all companies in the top ten, but each cluster contains at least seven of the top ten stocks. To try to find interesting (mis)classifications, a few "cross-over" stocks were added to the mix. For example, the stock WebMD, although not of the ten largest Health Care stocks, is an interesting case. WebMD is an online prescription and referral service. Although technically a Health Care company, it could also be considered an Internet stock. The behavior of WebMD stock is determined by how investors view the stock, not by the NYSE classifications. Large conglomerates such as AOL-Time Warner (Media), Honeywell (Aerospace), and 3M (Chemicals) may also be interesting to focus on, since these mammoth corporations could be seen as spanning several industrial categories. To note for comparison, the authors in [3] used 500 stocks spanning 102 categories, but they only looked at one year's worth of data compressed to 10 day averages.

The simplest clustering technique is hierarchical agglomerative clustering. We build a cluster tree by greedily combining elements so that we minimize some cluster distance measure. This tree is best visualized as a top-down tree called a dendrogram, resembling a genealogy tree. To obtain k clusters, we simply "cut" the tree at the k-th level from the top. We experimented with the four most common cluster measures: single link (MIN), complete link (MAX), average link, and Ward's Method. The Euclidean distance measure of the weekly percentage price changes was used to compare stocks. To help remove overall market trends, we used the time-based normalization described in the last section.

To learn about our methods and parameters, we first looked at a subset of our data. This was a necessary step to understand our data visually, since large dendrograms are difficult to interpret and, indeed, MATLAB won't even draw them. We will look at just the stocks in three industries: Health Care, Internet, and Media. This choice was made to examine the "Internet cross-overs" AOL-Time Warner and WebMD and also to see the effect of the highly volatile Internet stocks on the clustering. The time-series plots for these three industries are shown in

Figure 1. It is difficult to derive any significant differences between the categories visually, but the Internet stocks appear to be more volatile than both the Media and Health Care stocks. Note the erratic light blue line in the Media group is AOL-Time Warner and the erratic dark blue line in the Health Care group is WebMD.

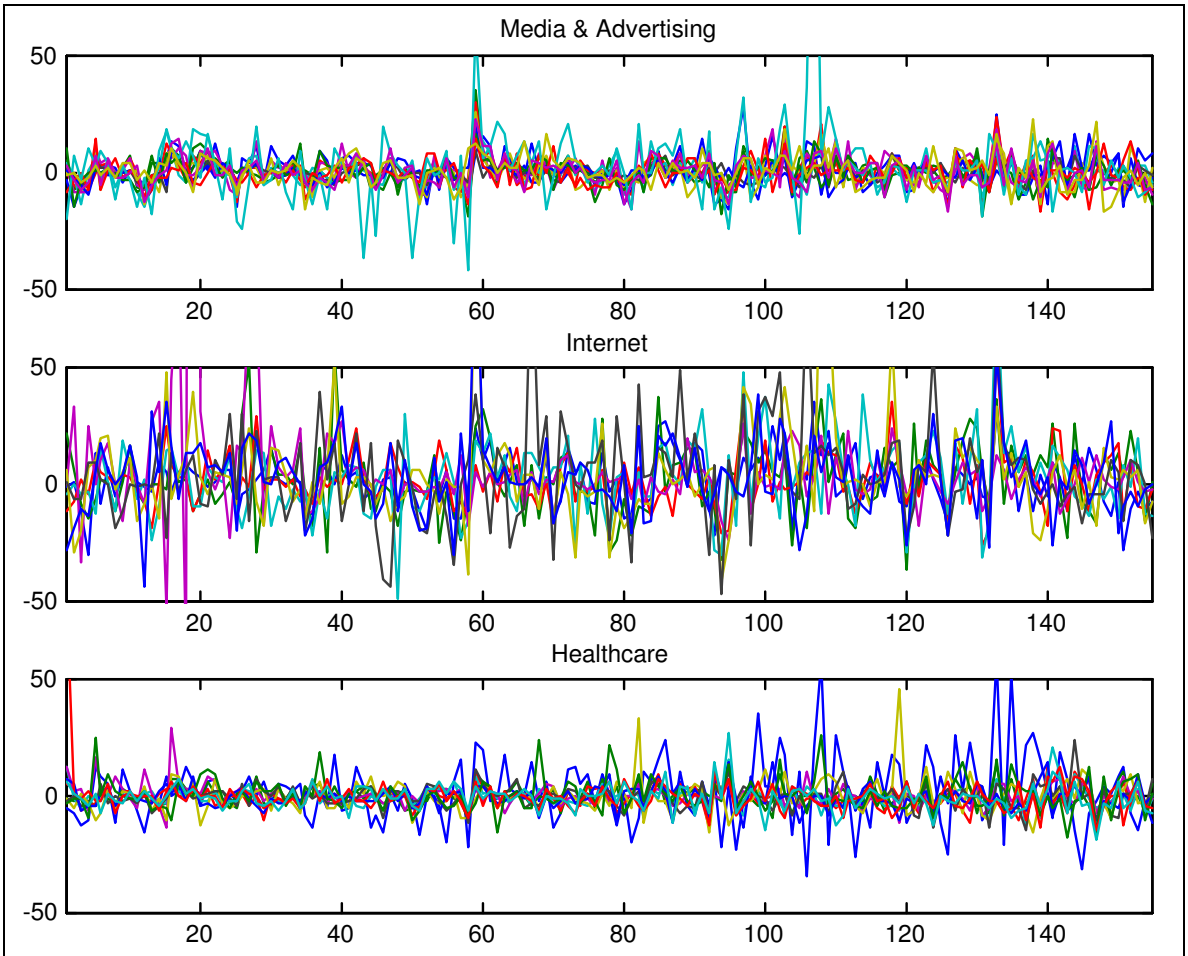

Figure 1. Time series plots of weekly percentage change of the training data stocks.

We ran hierarchical agglomerative clustering on this 3-industry data set under the single link, complete link, average link, and Ward's Method linkage schemes. The dendrograms are shown in Figure 2. As expected, the erratic Internet stocks were very troublesome to cluster accurately. In all four linkage schemes, we will get clusters containing only one stock regardless of where the cut is made. The single link method resulted in the most such size-one clusters. This is to be expected since the Internet stocks are essential noise points or outliers compared to the other industries and the single link method is particularly susceptible to outliers. If we ignore the presence of the Internet stocks, we can see the appropriate clusters forming with the Health Care and Media stocks under complete link and Ward's Method. So we should come up with a way to remove outliers from our final clustering. Gavrilov et. al. avoided this problem by not considering Internet stocks as one of their categories. Also, their paper looked at one year of data, 2000. Most stocks, particularly Internet stocks, experienced a sharp drop and subsequently became rather erratic following September 11, 2001.

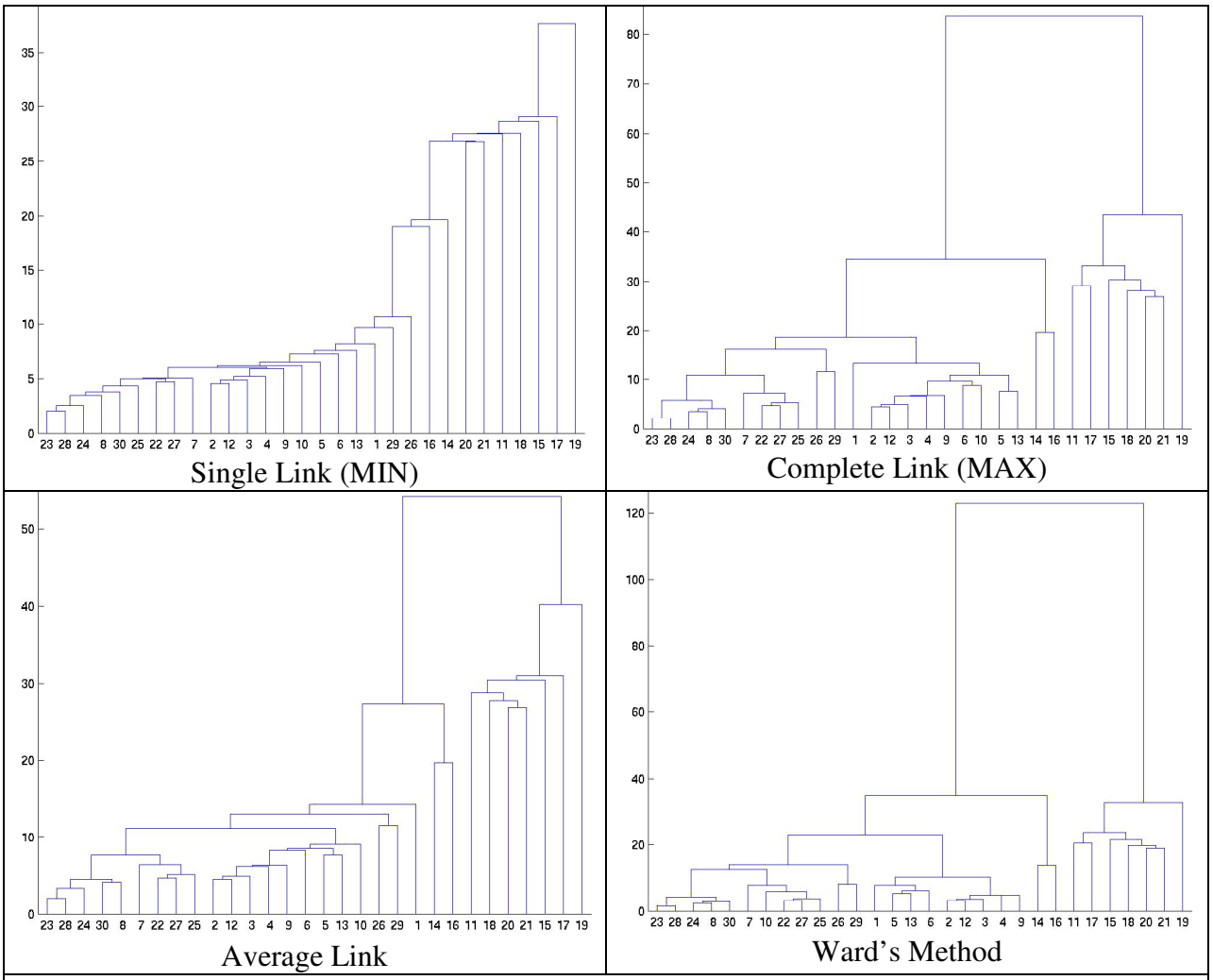

Figure 2. Dendrograms of the 3-industry training set under different linkage schemes.

There are many techniques for filtering outliers, most involved in the pre-processing of the data. One scheme is to calculate the average distance for every item to every other item and remove all items with an average distance exceeding a specified threshold. However, this requires domain knowledge and we are trying to develop a general heuristic for clustering stock data. A related pre-processing approach is to remove all items whose normalized average distance exceeds a threshold, say 3. That is, we remove stock i if its average distance $\bar{d}_i$ satisfies

$$\left| \frac{\bar{d}_i - \mu_d}{\sigma_d} \right| > 3$$

where μ_d is the mean average distance across all items and σ_d is the standard deviation. However, this approach assumes that the distances take on a normal distribution. A histogram of the average distances for all 91 stocks showed that the distribution was highly left-skewed. Since we do not want to incorporate domain knowledge into our outlier filter, we propose a post-processing approach based on the cluster tree produced. We cut the tree at the desired number of clusters and remove all stocks put in their own cluster. The cut and pruning needs to be repeated until each cluster has at least two elements. This approach would not be appropriate, of course,

if the ideal clustering contained a size-one cluster. But for data sets with a large number of items and few clusters, this assumption is reasonable. For our purposes, we use the knowledge that we selected several stocks from each industry. This scheme may remove a large number of items, but in our particular data set we expect most or all of the Internet stocks to be extremely volatile so we hope to remove a large number of stocks from our clustering.

We ran hierarchical agglomerative clustering on the entire 91 stock data set using the post-processing outlier detection scheme described above. We repeated the cut and pruning until we obtained 10 clusters of size at least 2. The single link scheme performed extremely poorly, as expected, and more than half of the stocks were pruned. The results for complete link, average link, and Ward's Method are shown in Appendix B. Similarly, the average link scheme performed poorly. The pruning had to eliminate 33 stocks, more than one-third of the data, before the data could be clustered into 10 groups. Fortunately, complete link and Ward's Method performed reasonably well. Complete link filtered out 18 stocks and achieved an overall purity of 0.6027. Note [3] chose complete link as their clustering scheme. Ward's Method performed even better, filtering out only 11 stocks and achieving a higher overall purity of 0.7875. Furthermore, 8 of the 11 outlier stocks removed were Internet stock, with a ninth stock being WebMD. If we view the set of pruned stocks as another cluster, then we have created a cluster that is 8/11 Internet stocks, lowering the overall purity only slightly to 0.7272.

As a final note on clustering, any clustering scheme would have difficulty properly grouping all 91 stocks, particularly the Internet stocks. An image of the normalized distance matrix is shown in Figure 3. The stocks are ordered as in Appendix A. The red regions indicate stocks that are "far" from other stocks. The dark blue regions indicate that stocks are very similar. Of course, there are dark blue dots along the diagonal. Ideally, we would see dark blue squares along the diagonal and red elsewhere. But our matrix is far from ideal. The Internet stocks are particularly distressing, showing dark red bands throughout. However, when compared to the un-normalized distance matrix, this matrix appears more suited to clustering.

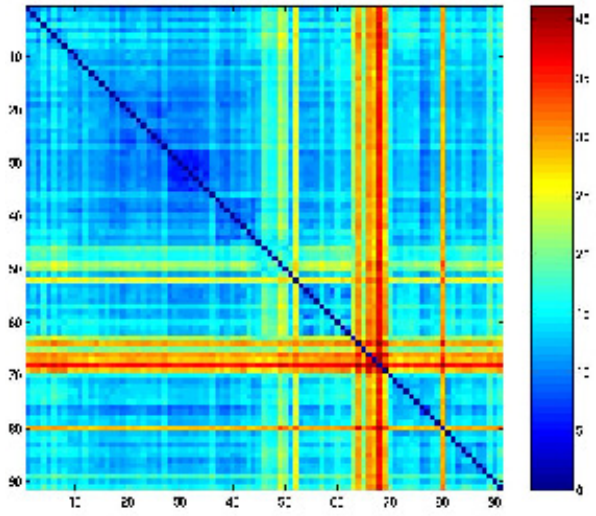

Figure 3. Time-normalized distance matrix.

2.4 Neural Networks

We briefly mention an alternative classification technique that does not work as well as hierarchical agglomerative clustering, but may be promising with refinement. A feed-forward neural network is a weighted graph that is "trained" so that a certain input or set of inputs X will result in a specified output or set of outputs T. Then if an input is given that is similar to an input in the training data, it should give rise to a similar output. A typical neural network has an input layer, one or more "hidden" layers consisting of p hidden nodes, and the output layer (see Figure 4). At the hidden node, we generally incorporate an activation function so that information will only be passed to the next node if a certain threshold is reached. The training will be computationally expensive since we must set $p(|X|+|T|)$ weights.

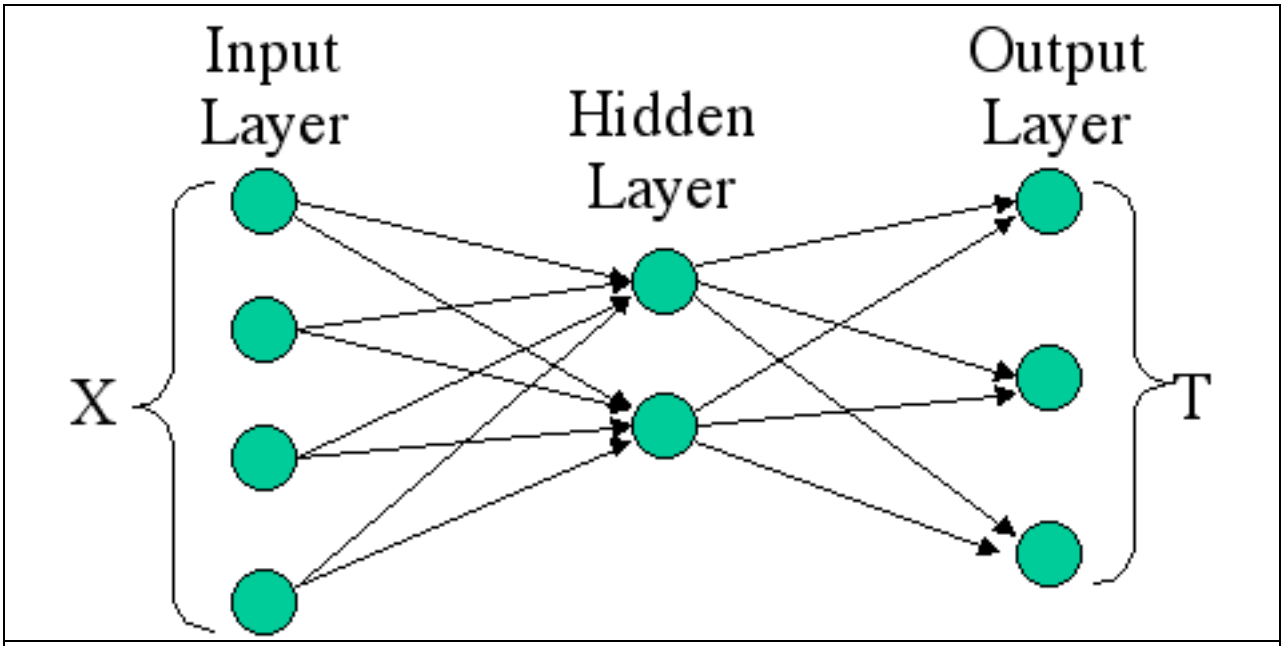

Figure 4. Diagram of a general feed-forward neural network.

Giles et. al. used a neural network for prediction of stock data, but we could also use a neural network for classification of stock data [5]. In our case, the three-year time series of percent changes can be fed in as input. We did not normalize the data, hoping that the network would "learn" how to normalize the data itself. The output will be a vector of length 10 indicating the probability that the stock belongs to each category. For the training data, the input will be given as a 1 in the desired category and 0's elsewhere. We used one hidden layer with p=20 hidden nodes and a sigmoid activation function. We trained the network on all 91 stocks using Levenberg-Marquad training, which can be thought of as a multi-dimensional steepest descent technique [6]. The training was very expensive computationally, since the network had to train $p(|X|+|T|) = 20(156+10) = 3320$ weights to match 91 pieces of data. One iteration of the training took 1,117 seconds. We trained the network for 200 iterations, roughly 3.2 hours. Figure 5 shows a plot of the residual on the training set, which is defined as the difference in the desired network output and the actual network output:

$$\text{Residual} = \|\text{Desired Output} - \text{Actual Output}\|_2 .$$

The plot of the residual versus the iteration number is shown in Figure 5. Note the residual is a monotone decreasing function, which indicates that the learning process only improves the

network. To determine the result of a forward run of the network, we simply look at the output node that take on the maximum value. This should indicate the cluster that the stock belongs to with the highest probability.

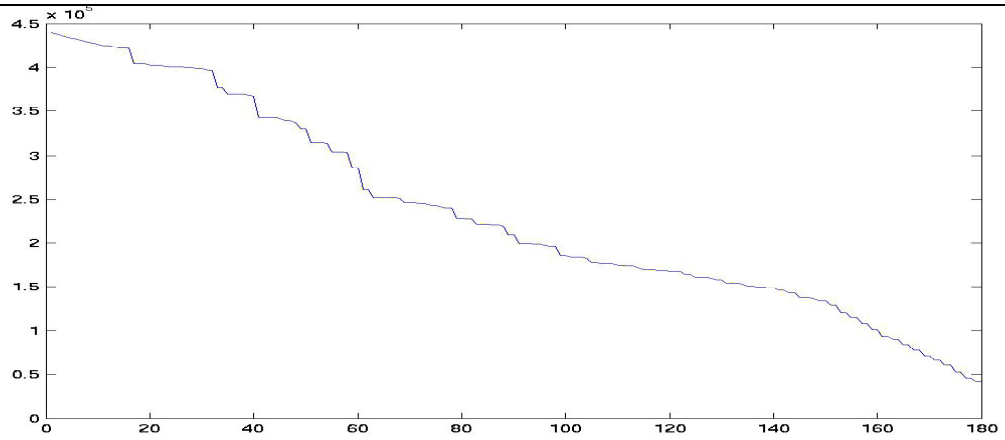

Figure 5. Plot of residual vs. time for neural network training.

Unfortunately, the results were rather disappointing. The neural network was able to correctly classify only 35 of the 91 stocks, roughly 38%. Since there are 10 clusters random guessing would only result in 10% accuracy, so the neural network is better than random but it is still far from being an acceptable clustering scheme. Interestingly, the network preferred clusters 4 (Energy) and 10 (Retail), since all of the misclassified stocks were assigned one of these numbers. Perhaps the network could achieve better accuracy with more training, more hidden nodes/layers, and normalized data.

A major drawback of the neural network approach is that it can only identify industries that is has been trained on. Agglomerative clustering, on the other hand, does not assume beforehand what industries will appear in the data. Although the training is excruciatingly slow, a forward run of the network is very fast, since it only involves the multiplication and addition of weights. Neural networks are not set up to handle missing values, while we could adapt our Euclidean distance measure to handle missing values in our clustering schemes. Neural networks also have difficulty with discontinuous data, which are stock data might be. Despite these drawbacks and the network's poor performance, training a neural network to identify a stock's industry is an intriguing prospect.

3. Association Analysis

We saw in the last section that clustering stocks is very difficult because stocks are correlated across industries. So a natural extension of the attempt at clustering is to try to determine what the relationships among the industries actually are. Association analysis attempts to generate rules that describe such behavior.

3.1. Rule Generation

By way of a "market basket data" analogy, we think of association analysis acting on shopping baskets in a grocery store. Each basket can contain one of several possible items. We record the presence of or absence of each item in each basket. The goal is to determine which

items occur together in shopping baskets frequently. We say a rule $X \Rightarrow Y$ is true if the presence of item X implies the presence of item Y in a random market basket. For example, if many customers who purchase milk also purchase eggs, we could say the itemset {Milk, Eggs} is frequent and we could deduce the rule $Milk \Rightarrow Eggs$. To quantize how often these items occur together, we use the support statistic:

$$\text{supp}(X \Rightarrow Y) = \frac{\sigma(X,Y)}{N}$$

where $\sigma(X)$ denotes the frequency of item X and N is the total number of market baskets. To describe the probability that the presence of item X implies the presence of item Y, we use the confidence statistic:

$$\text{conf}(X \Rightarrow Y) = \frac{\sigma(X,Y)}{\sigma(X)}.$$

The best rules will have both high support and confidence. The simplest way to generate such rules is to enumerate all possible rules, remove all rules with support less than minimum support threshold T, and sort the remaining rules based on confidence. To save on computation, we do not need to enumerate all rules. The A Priori Principle states that if an itemset is frequent, then all of its subsets must be frequent as well. Using this principle, once we detect an itemset with support less than T, we can prune all supersets of that itemset from consideration as well.

These basic ideas of grocery store rule generation can be applied to time-series [7]. In our case, we wish to detect which industries are related. That is, we wish to know if a change in stock prices in one industry causes a change in stock prices in a separate industry. Suppose that for each industry we had a single time-series that described the general behavior of all or most of the stocks in that industry. For example, we could use a weighted average of stock prices within that industry. The "items" in this case would be a change in stock prices. Since we wish to track the industry behavior, each industry would lead to two items: prices going up and prices going down. A "basket" would be a specified time period, say a week. For example, a week might give rise to the itemset {Internet Up, Media Down, Aerospace Down}.

As described in Sec 2.1, the percent change $P_i(t)$ of the stock gives a consistent measure across stocks. So at a time t, we determine the behavior of stock i for the week according to:

$$I_i(t) = \begin{cases} \{\text{Stock i Up}\} & \text{if } \mathrm{P_i}(t) > 0 \\ \{\text{Stock i Down}\} & \text{if } \mathrm{P_i}(t) < 0 \end{cases}.$$

The itemset I(t) for the week at time t will then be

$$I(t) = \bigcup_i I_i(t).$$

Unfortunately, this implementation will result in dense data sets. If we consider a total of M stocks, then there will 2M possible items. Since a stock very rarely remains constant, then each week's itemset I(t) will contain M entries. To help "sparsify" our data, we set a minimum percent change threshold P>0. We then define our the item generated by stock i as:

$$I_i(t) = \begin{cases} \{\text{Stock i Up}\} & \text{if } \mathrm{P_i}(t) > P \\ \{\text{Stock i Down}\} & \text{if } \mathrm{P_i}(t) < -P \\ \varnothing & \text{if } -\mathrm{P} \le \mathrm{P_i}(t) \le P \end{cases}.$$

This concept is illustrated in Figure 6 on the next page.

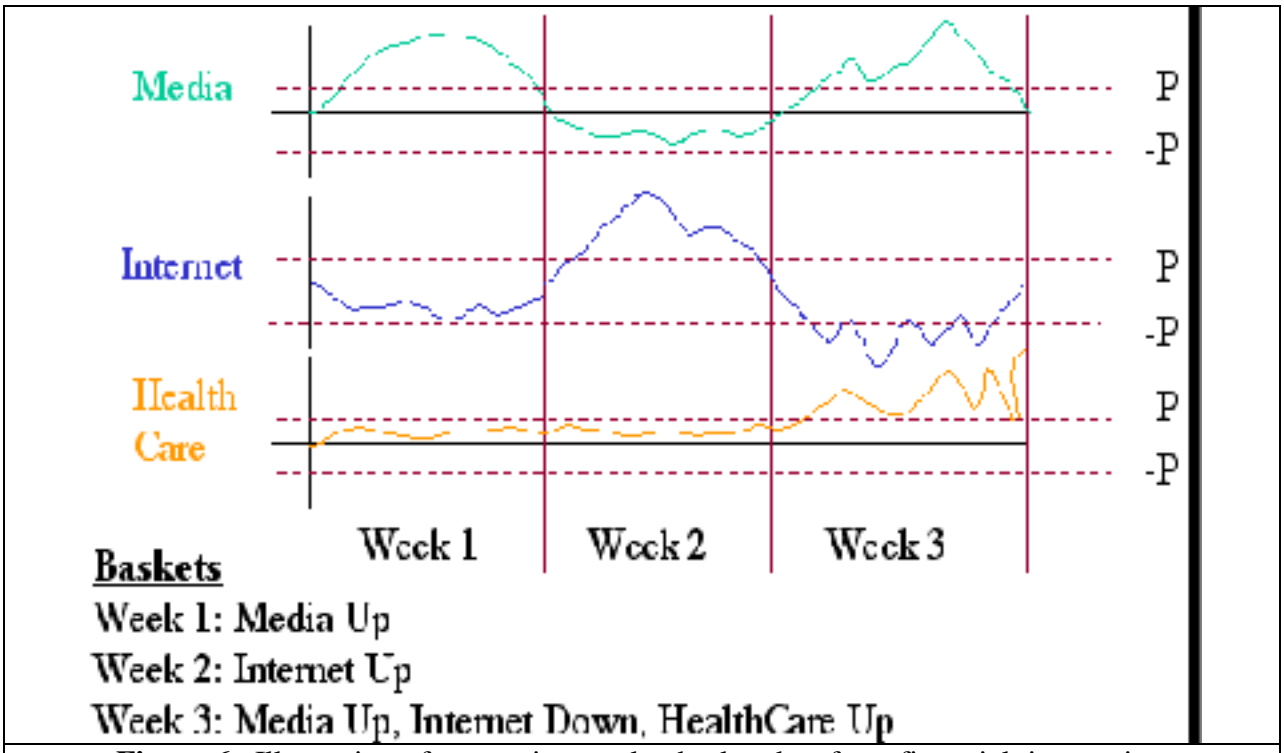

Figure 6. Illustration of generating market basket data from financial time-series.

The market basket set-up described above detects price changes that occur within the same week. Rather than detecting concurrent fluctuations, it would be more useful to develop a predictive system. We could adapt our market baskets by staggering the weeks. Suppose we wish to detect how a change in one week affects the following week. In generating the rule $X \Rightarrow Y$, we look at the precedent X occurring in the week *prior* to the antecedent Y. For example, if the item {Media Up} is frequently followed by the item {Internet Up} in the following week, we could generate the rule Media Up $\Rightarrow$ Internet Up. This one-week delay approach can be generalized to any number of weeks. Of course, the farther we attempt to look into the future, the less reliable our predictions would become.

3.2. Results

The well-known Dow Jones Index is a weighted average of the largest companies' stocks in the NYSE. There are also Dow Jones indices for each of the major industries, also weighted averages of the largest companies in that industry. We obtained 23 such indices from [2]. These indices cover the industries described in Sec. 2, such as Aerospace, Media, and Internet. In an attempt to mine interesting rules from the data, we also included other indices such as the 30 Year Bond Index and the Philadelphia Gold & Silver Index. We gathered data for the three year period ranging from Nov. 1, 1999 to Nov. 1, 2002. Unfortunately, some of the indices are recent inventions, so data was not available for all three years for all indices. Our technique was adapted to handle the missing values. For each index, we calculated the weekly percentage change across the three year time period. We then created the "baskets" described in the last section, using a percentage change threshold P=1. This threshold was chosen because a weekly price change of less than 1% is generally considered not significant. So the minor price

fluctuations are filtered out as "noise," even though technically there is no noise in our system since the data represents actually dollars and cents. We ran a simple A Priori approach on the data with a minimum support threshold T=0.25. This threshold was chosen simply because it pruned a significant portion of the rules. The pairwise rules, rules with itemset size 2, are shown in Appendix C. We concentrate on such rules because we are trying to see how each industry correlates with another.

Note that the rules with the highest confidence involve two basic itemsets: {Software, Internet, Semiconductors, Telecommunications} and {Pharmaceuticals, Biotechnology, Health Care}. It seems natural that these sets should form the strongest, almost obvious rules. Interesting and unexpected rules include Rule 9: Construction Down $\Rightarrow$ Transportation Down, Rule 10: Biotech Down $\Rightarrow$ Internet Down, and Rule 27: Auto Down $\Rightarrow$ Internet Down. To find unexpected rules, we must look further down the list, which gives us less confidence that these rules are due to more than chance. Also, we must take the support into account. For example, Rule 16: Chemicals Down $\Rightarrow$ Transportation Down has a farily high confidence of 0.7547. But the support is only 0.2541, which indicates the two industries move down together about one-quarter of the time. It is somewhat surprising that there were no rules in which one industry went up and another went down. It seems reasonable that a drop in one industry would not cause a rise in another industry, but we still would expect to see such a rule occur just by chance. However, the stock market tends to move as a whole and the percent change threshold P eliminated noise-like fluctuations.

Using the same data set of 23 indices over the three year period, we ran the A Priori algorithm with a one-week delay as described in the last section. We again used a percent change threshold P=1.0, but we had to lower the support threshold to T=0.20 to see a significant number of rules. This indicates that is much harder to predict the stock market than to merely observe it. The rules seem to be more interesting than the concurrent rules, such as Rule 1: Financial Down $\Rightarrow$ Internet Down and Rule 6: 30 Year Bonds Down $\Rightarrow$ Internet Down. However, the confidence and support of the rules are lower than those of the rules generated for concurrent weeks. Note that once again we see several natural rules involving the technological and medical industries.

4. Conclusion

We saw in Sec. 2 that is difficult to cluster stocks into their appropriate industries, given the widely erratic behavior of certain stocks. The Internet industry in particular shows volatile behavior that is difficult to track. By filtering the outliers, we improve the clustering greatly. We could even treat the set of outliers as another cluster, assuming the outliers are Internet stocks with a high probability. Of the methods we considered, Ward's Method appeared to do the best job clustering the data. The feed-forward neural network, although a promising direction, did not perform very well. It would be interesting to see how a density-based clustering scheme, such as DBscan, performs on financial data. Such schemes automatically filter outliers, which might lead to superior results. If some domain knowledge is brought into the problem, it might be possible to pre-process the data to remove outliers. The outlier detection problem is a very difficult open problem in and of itself. But we should not accept the NYSE classification as the absolute truth. For example, hierarchical agglomerative clustering classified WebMD as an Internet rather than a Health Care stock. This indicates that WebMD

behaves more like an Internet stock than a Health Care stock, which is a more interesting result than obtaining the same classification as the NYSE.

In Sec 3, we showed that the standard "market basket" association analysis could be applied to financial time series. This can be used to generate rules which point out the industries that move together. The rules generated with the highest confidence and support seemed to be natural, which indicates that our system is working. The unexpected rules will necessarily have lower support and confidence. We also demonstrated that we could incorporate a one-week delay into the rule generation. However, the rules generated had lower support and confidence and also appeared less natural than the concurrent rules. This indicates that the stock market is inherently unpredictable. One possible area for future work is to determine which stocks have the greatest influence on an industry. For example, suppose we want to know whether IBM or Dell is a better predictor of the behavior of the Computer Hardware industry as a whole. We could make our set of items to be the fluctuations of the Dow Jones Computer Hardware index and the largest, say, 30 companies in the Computer Hardware category. Applying the one-week association analysis to this set would be useful to investors who are trying to predict the movement of a certain industry and would like a specific company that predicts this movement.

References

[1] A. Weigend. "Data Mining in Finance: Report from the Post-NNCM-96 Workshop on Teaching Computer Intensive Methods for Financial Modeling and Data Analysis." Proc. Fourth International Conference on Neural Networks in the Capital Markets NNCM-96, p. 399-411, 1997.

[2] "Yahoo! Financial." http://finance.yahoo.com.

[3] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. "Mining the Stock Market: Which Measure is Best?" Proc. of the KDD, p. 487-496, 2000.

[4] K. Kalpakis, D. Gada, and V. Puttagunta. "Distance Measures of ARIMA Time-Series." Technical Report TR-CS-01-14, CSEE, UMBC, 2001.

[5] C. Giles, S. Lawrence, and A. Tsoi. "Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference." Machine Learning, Vol. 44, No. 12, p. 161-183, July/August 2001.

[6] V. Rao and H. Rao. *C++ Neural Networks & Fuzzy Logic*. MIS Press: New York, 1995.

[7] G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth. "Rule Discovery from Time Series." Proc. of the KDD, p. 16-22, 1998.

Appendix A: Stock Data

These are the 91 stocks used by the clustering methods described in Sec. 2. The industrial groups are shown as reported by [2]. Data for all stocks was weekly percentage change from Nov. 1, 1999 to Nov. 1, 2001.

Table A.1

Aerospace & Defense		32.	Chevron Texaco Corp.	Internet	
1.	United Technologies Corp.	33.	Shell Transport & Trading	63.	Yahoo! Inc.
2.	The Boeing Co.	34.	Eni S p A	64.	Overture Services Inc.
3.	Lockheed Martin Corp.	35.	Conoco Phillips	65.	EarthLink Inc.
4.	Honeywell Intl. Inc.	36.	Schlumberger Ltd.	66.	United Online Inc.
5.	General Dynamics Corp.	Financial Services		67.	Priceline.com Inc.
6.	Raytheon Co.	37.	Citigroup Inc.	68.	Covad Comm. Group
7.	Northrop Grumman Corp.	38.	American Intl. Group	69.	CNET Networks Inc.
8.	L-3 Communications	39.	HSBC Holding plc	Media & Advertising	
Automotive		40.	Bank of America Corp.	70.	AOL Time Warner
9.	Toyota Motor Corp.	41.	Wells Fargo & Co.	71.	Walt Disney Co.
10.	Honda Motor Corp.	42.	Fannie Mae	72.	News Corp. Ltd.
11.	Daimler Chrysler AG	43.	JP Morgan Chase & Co.	73.	Clear Channel Comm.
12.	Nissan Motor Corp.	44.	American Express Co.	74.	Liberty Media Corp.
13.	General Motors Corp.	Computer Hardware		75.	Fox Entertainment Group
14.	Ford Motor Co.	45.	IBM	76.	Gannett Co., Inc.
15.	Harley-Davidson Inc.	46.	Cisco Systems Inc.	77.	Tribune Co.
16.	AB Volvo	47.	Dell Computer Corp.	78.	Metro-Goldwyn-Mayer
17.	Johnson Controls Inc.	48.	Hewlett-Packard	79.	Pixar
18.	PACCAR Inc.	49.	EMC Corp.	80.	IMAX Corp.
Chemicals		50.	Sun Microsystems Inc.	81.	Viacom
19.	3M Co.	51.	Pitney Bowers Inc.	82.	Sony Corp.
20.	E.I. du Pont and Co.	Health Care		Retail	
21.	Dow Chemical Co.	52.	WebMD Corp.	83.	Wal-Mart Stores Inc.
22.	Akzo Nobel N.V.	53.	Medtronic Inc.	84.	The Home Depot Inc.
23.	Bayer AG	54.	United Health Group Inc.	85.	Lowe's Companies Inc.
24.	Air Products & Chemicals	55.	HCA Inc.	86.	Walgreen Co.
25.	Praxair, Inc.	56.	Baxter International Inc.	87.	Target Corp.
26.	PPG Industries, Inc.	57.	Boston Scientific Corp.	88.	Kohl's Corp.
27.	Rohm and Haas Co.	58.	Stryker Corp.	89.	The Gap Inc.
Energy		59.	WellPoint Health	90.	Costco Wholesale Corp.
28.	Exxon Mobil Corp.	60.	Guidant Corp.	91.	Ito-Yokada Co., Ltd.
29.	BP plc	61.	Tenet Healthcare		
30.	Total Fina Elf SA	62.	Universal Health Services		
31.	Royal Dutch Petroleum				

Below are the time-series plots of all 91 stocks by industry. All plots cover the three year period from Nov. 1, 1999 to Nov. 1, 2001. The same scale was used for all plots. Note the Internet stocks are much more volatile than the other 9 categories.

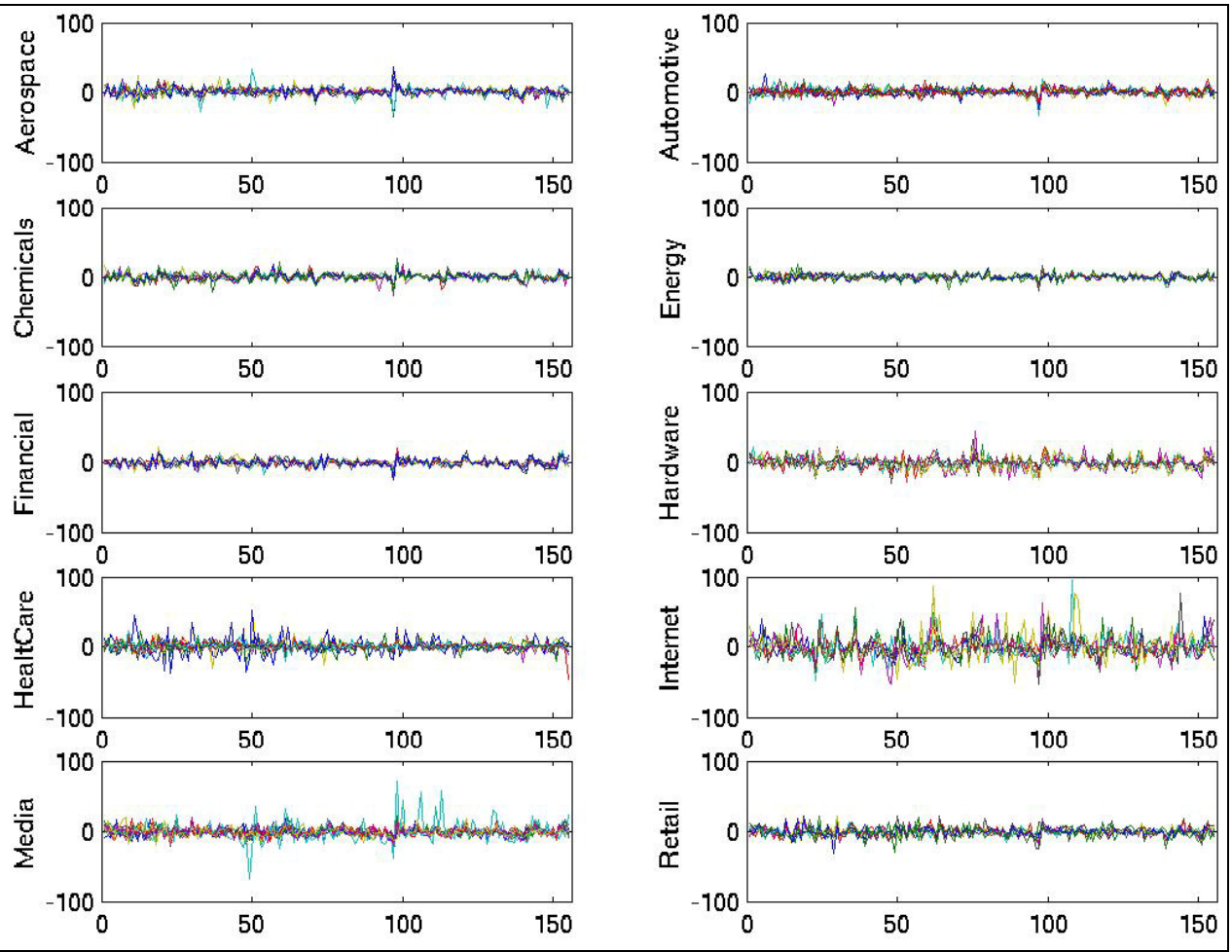

Figure A.1. Time-series plots of all 91 stocks by industry.

Appendix B: Hierarchical Agglomerative Clustering Results

Below are the confusion matrices and statistics under the complete link, average link, and Ward's Method schemes. All 91 stocks were used for the three-year period.

Table B.1: Complete Link

Cluster#	Ae	Au	Ch	En	Fi	Ha	He	In	Me	Re	Purity	Entropy
1	5	0	7	0	0	0	0	0	2	0	0.5	1.4316
2	0	2	0	0	3	1	1	0	0	5	0.4167	2.0546
3	0	4	0	0	0	0	2	0	1	0	0.5714	1.3788
4	0	3	2	9	4	0	0	0	0	0	0.5	1.7652
5	0	0	0	0	0	0	5	0	0	0	1	0
6	0	1	0	0	1	1	0	0	2	0	0.4	1.9219
7	0	0	0	0	0	0	0	0	0	2	1	0
8	0	0	0	0	0	0	0	0	6	0	1	0
9	0	0	0	0	0	2	0	0	0	0	1	0
10	2	0	0	0	0	0	0	0	0	0	1	0

Outliers (18 total)
4 47 48 49 52 57 60 63 64 65 66 67 68 69 70 80 89 91

Overall Entropy = 1.3114
Overall Purity = 0.6027

Table B.2: Average Link

Cluster#	Ae	Au	Ch	En	Fi	Ha	He	In	Me	Re	Purity	Entropy
1	0	0	0	0	0	0	0	0	2	0	1	0
2	0	0	0	0	0	0	0	0	2	0	1	0
3	4	4	9	8	2	0	1	0	2	0	0.3	2.4892
4	0	1	0	0	0	0	2	0	0	0	0.6667	0.9183
5	0	0	0	0	6	0	0	0	0	1	0.8571	0.5917
6	0	0	0	0	0	0	0	0	0	3	1	0
7	0	2	0	0	0	0	0	0	0	0	1	0
8	0	2	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	2	1	0
10	0	0	0	0	0	0	5	0	0	0	1	0

Overall Entropy = 1.4064
Overall Purity = 0.6034

Outliers (33 total)
2 4 6 8 12 36 45 46 47 48 49 50 51 52 57 60
63 64 65 66 67 68 69 70 74 75 78 79 80 82 89 90 91

Table B.3: Ward's Method

Cluster#	Ae	Au	Ch	En	Fi	Ha	He	In	Me	Re	Purity	Entropy
1	1	9	2	0	0	1	0	0	2	0	0.6	1.7383
2	0	0	0	0	0	0	0	0	0	6	1	0
3	0	0	0	0	0	1	0	0	8	0	0.8889	0.5033
4	0	0	0	0	0	2	0	0	1	0	0.6667	0.9183
5	0	0	0	0	8	0	0	0	0	0	1	0
6	1	0	7	0	0	0	0	0	0	0	0.8570	0.5436
7	0	0	0	9	0	0	0	0	0	0	1	0
8	6	1	0	0	0	0	5	0	1	1	0.4286	1.8703
9	0	0	0	0	0	0	5	0	0	0	1	0
10	0	0	0	0	0	3	0	0	0	0	1	0

Overall Entropy = 0.7986
Overall Purity = 0.7875

Outliers (11 total)
52 63 64 65 66 67 68 69 80 89 91

Appendix C: Association Rules

Table A.1 The list below shows all pairwise concurrent rules with support greater than T=0.25. The percentage threshold used was P=1.0 %. The values are sorted on confidence.

Rule	Support	Confidence
1. Software Down => Internet Down	0.4000	0.9254
2. Pharmaceuticals Down => HealthCare Down	0.2645	0.8723
3. HealthCare Down => Biotech Down	0.2774	0.8431
4. Software Down => Semiconductors Down	0.3613	0.8358
5. Semiconductors Down => Internet Down	0.3871	0.8219
6. Software Up => Semiconductors Up	0.2581	0.8163
7. Software Up => Internet Up	0.2581	0.8163
8. HealthCare Down => Pharmaceuticals Down	0.2645	0.8039
9. Construction Down => Transportation Down	0.2774	0.7963
10. Biotech Down => Internet Down	0.3032	0.7833
11. Chemicals Down => Industrial Down	0.2645	0.7736
12. Financial Down => Industrial Down	0.2581	0.7692
13. Semiconductors Down => Software Down	0.3613	0.7671
14. Telecom Down => Internet Down	0.3484	0.7606
15. Insurance Down => Transportation Down	0.2645	0.7593
16. Chemicals Down => Transportation Down	0.2581	0.7547
17. Industrial Up => Transportation Up	0.2516	0.7500
18. Financial Down => Internet Down	0.2516	0.7500
19. Internet Down => Software Down	0.4000	0.7381
20. Transportation Up => Industrial Up	0.2516	0.7222
21. Industrial Down => Semiconductors Down	0.2645	0.7193
22. Industrial Down => Internet Down	0.2645	0.7193
23. Industrial Down => Chemicals Down	0.2645	0.7193
24. Industrial Down => Transportation Down	0.2645	0.7193
25. Biotech Down => HealthCare Down	0.2774	0.7167
26. Internet Down => Semiconductors Down	0.3871	0.7143
27. Auto Down => Internet Down	0.2645	0.7069
28. Telecom Down => Semiconductors Down	0.3226	0.7042
29. Industrial Down => Financial Down	0.2581	0.7018
30. Software Down => Telecom Down	0.3032	0.7015
31. Biotech Down => Software Down	0.2710	0.7000
32. Biotech Down => Semiconductors Down	0.2710	0.7000
33. Semiconductors Up => Internet Up	0.2710	0.7000
34. Internet Up => Semiconductors Up	0.2710	0.7000
35. Auto Down => Semiconductors Down	0.2581	0.6897
36. Semiconductors Down => Telecom Down	0.3226	0.6849
37. Transportation Down => Internet Down	0.2839	0.6769
38. Auto Down => Telecom Down	0.2516	0.6724
39. Semiconductors Up => Software Up	0.2581	0.6667
40. Internet Up => Software Up	0.2581	0.6667
41. Telecom Down => Software Down	0.3032	0.6620

42. Transportation Down => Construction Down	0.2774	0.6615
43. Biotech Down => Telecom Down	0.2516	0.6500
44. Transportation Down => Semiconductors Down	0.2710	0.6462
45. Internet Down => Telecom Down	0.3484	0.6429
46. Transportation Down => Industrial Down	0.2645	0.6308
47. Transportation Down => Insurance Down	0.2645	0.6308
48. Software Down => Biotech Down	0.2710	0.6269
49. Transportation Down => Chemicals Down	0.2581	0.6154
50. Transportation Down => Telecom Down	0.2516	0.6000
51. Semiconductors Down => Biotech Down	0.2710	0.5753
52. Semiconductors Down => Transportation Down	0.2710	0.5753
53. Semiconductors Down => Industrial Down	0.2645	0.5616
54. Internet Down => Biotech Down	0.3032	0.5595
55. Telecom Down => Biotech Down	0.2516	0.5493
56. Telecom Down => Auto Down	0.2516	0.5493
57. Telecom Down => Transportation Down	0.2516	0.5493
58. Semiconductors Down => Auto Down	0.2581	0.5479
59. Internet Down => Transportation Down	0.2839	0.5238
60. Internet Down => Industrial Down	0.2645	0.4881
61. Internet Down => Auto Down	0.2645	0.4881
62. Internet Down => Financial Down	0.2516	0.4643

Table A.2 The list below shows all pairwise 1-week-delay rules with support greater than T=0.20. The percentage threshold used was P=1.0 %. The values are sorted on confidence.

Rule	Support	Confidence
1. Financial Down => Internet Down	0.2143	0.6346
2. Software Down => Internet Down	0.2727	0.6269
3. Auto Down => Internet Down	0.2338	0.6207
4. Semiconductors Down => Internet Down	0.2922	0.6164
5. Telecom Down => Internet Down	0.2792	0.6056
6. 30 Yr Bonds Down => Internet Down	0.2143	0.6000
7. Industrial Down => Internet Down	0.2208	0.5965
8. Construction Down => Biotech Up	0.2013	0.5741
9. Construction Down => Internet Down	0.2013	0.5741
10. Transportation Down => Internet Down	0.2403	0.5692
11. Biotech Up => Internet Down	0.2143	0.5690
12. Software Down => Telecom Down	0.2468	0.5672
13. Biotech Down => Semiconductors Down	0.2208	0.5667
14. Biotech Down => Internet Down	0.2208	0.5667
15. 30 Yr Bonds Down => Telecom Down	0.2013	0.5636
16. Internet Down => Telecom Down	0.3052	0.5595
17. Semiconductors Up => Semiconductors Down	0.2143	0.5500
18. Gold&Silver Down => Internet Down	0.2338	0.5455
19. Auto Down => Semiconductors Down	0.2013	0.5345

20. Semiconductors Down => Telecom Down	0.2532	0.5342
21. Biotech Down => Telecom Down	0.2078	0.5333
22. Aerospace Up => Internet Down	0.2078	0.5246
23. Gold&Silver Up => Internet Down	0.2143	0.5238
24. Software Down => Semiconductors Down	0.2273	0.5224
25. Telecom Down => Software Down	0.2403	0.5211
26. Telecom Down => Semiconductors Down	0.2403	0.5211
27. Aerospace Up => Semiconductors Up	0.2013	0.5082
28. Transportation Down => Telecom Down	0.2143	0.5077
29. Internet Down => Software Down	0.2727	0.5000
30. Gold&Silver Up => Gold&Silver Down	0.2013	0.4921
31. Internet Down => Semiconductors Down	0.2662	0.4881
32. Software Down => Gold&Silver Up	0.2078	0.4776
33. Software Down => Chemicals Up	0.2078	0.4776
34. Transportation Down => Semiconductors Down	0.2013	0.4769
35. Transportation Down => Gold&Silver Up	0.2013	0.4769
36. Gold&Silver Down => Gold&Silver Up	0.2013	0.4697
37. Semiconductors Down => Aerospace Up	0.2208	0.4658
38. Semiconductors Down => Gold&Silver Up	0.2208	0.4658
39. Semiconductors Down => Insurance Up	0.2208	0.4658
40. Internet Down => Biotech Down	0.2468	0.4524
41. Semiconductors Down => Software Down	0.2143	0.4521
42. Semiconductors Down => Biotech Up	0.2143	0.4521
43. Semiconductors Down => Semiconductors Up	0.2143	0.4521
44. Semiconductors Down => Chemicals Up	0.2143	0.4521
45. Telecom Down => Aerospace Up	0.2078	0.4507
46. Telecom Down => Gold&Silver Up	0.2078	0.4507
47. Internet Down => Auto Down	0.2403	0.4405
48. Internet Down => Gold&Silver Up	0.2403	0.4405
49. Internet Down => Gold&Silver Down	0.2403	0.4405
50. Semiconductors Down => Auto Down	0.2078	0.4384
51. Internet Down => Semiconductors Up	0.2338	0.4286
52. Internet Down => Insurance Up	0.2338	0.4286
53. Internet Down => Chemicals Up	0.2338	0.4286
54. Internet Down => Transportation Down	0.2338	0.4286
55. Internet Down => Construction Down	0.2273	0.4167
56. Internet Down => HealthCare Down	0.2273	0.4167
57. Internet Down => Aerospace Up	0.2208	0.4048
58. Internet Down => Financial Down	0.2143	0.3929
59. Internet Down => Chemicals Down	0.2143	0.3929
60. Internet Down => Industrial Down	0.2078	0.3810
61. Internet Down => 30 Yr Bonds Down	0.2078	0.3810