Feature Selection for Stock Data Analysis

Doshi Shailesh
Java Akshay
Shanbhag Vishal

CMSC 691D, FALL 2001
University of Maryland Baltimore County

1. Executive summary

Section 2 gives the problem definition of the project and describes the goals of this project. Section 3 gives the background details about basic stock market data and time series analysis techniques and rule discovery methods. This also provides basic details of the ARIMA model that we have used in our implementation. Section 4 provides the extensive literature review and describes the various resources and references that we have used for our project. It also details some of the approaches that we had come across such as Fourier transformations and other techniques that can be used. Section 5 describes in detail the precise approach and the technical details of the project where we propose our idea. In section 6 the distribution of work among the project members is explained. In section 7 we provide the detailed description of the experimental analysis that we have performed and the implementation difficulties that we have faced during the project, we also summarize all the graphs and readings we observed during the analysis and present the results that we got. Finally we conclude the report and give the references.

2. Problem definition

Financial time series data, such as stock prices, options and index, are often highly erratic and contain complex behaviors which make their accurate prediction, even on a short-term basis, extremely difficult. Using certain statistical and data mining techniques we can predict the future trends and values of the financial data such as the stock value, volume of trading and other performance indices. The various techniques that are used are based on the fact that there is a correlation between the current trends and future prices. The use of time series analysis and non-linear models such as neural networks and genetic algorithms can successfully predict the future trends in financial data. The variation in the stock prices is highly dependent on a set of attributes that influence it. By learning the relation between these attributes and the stock price it would be possible to gain a fair approximation to the variation in the latter.

There are various types of financial data available that needs to be extracted for data mining purpose for example the parameters that would have to be used to predict the price of the stock would be the highest trading price, lowest trading price, volume and other factors such as inflation, interest rates and bullion values. This data is available in many different formats and needs to be extracted before it may be efficiently used.

The objective of this project would be to investigate the use of data mining methods along with the statistical domain knowledge to build a system that can perform automatic data collection and train on this data to predict the future trend in the stock prices based on the trends in the associated attributes.

We can formulate the problem definition as follows:

To analyze the stock data intelligently so as to extract those features, among the large number of features, those produce maximum variation in the stock price. Then try to find a relation between each of these features and the actual stock price, to help us in predicting the trend in the price of the stock based on the trend in those selected features.

3. Background

3.1 Time Series

3.1.1 Time Series Data

Time series is defined as an ordered sequence of values of a variable at equally spaced time intervals. Some examples of such data are the weather data, stock data and many more. A number of standard techniques are available to understand the behavior of such time series and to fit a model to it.

The usage of time series models is twofold:

1) Obtain an understanding of the underlying forces and structure that produced the observed data

2) Fit a model and proceed to forecasting, monitoring or even feedback and feed forward control.

Time Series Analysis is used for many applications such as:

a) Economic Forecasting
b) Sales Forecasting
c) Budgetary Analysis
d) Stock Market Analysis
e) Census Analysis

3.1.2 Time Series Properties

a) Stationarity

A stationary process has the property that the mean and variance do not change over time. Stationarity can be defined in precise mathematical terms, but for our purpose we mean a flat looking series, without trend, constant variance over time, and no periodic fluctuations

b) Seasonality

Many time series display seasonality. By seasonality, we mean periodic fluctuations. For example, retail sales tend to peak for the Christmas season and then decline after the holidays. So time series of retail sales will typically show increasing sales from September through December and declining sales in January and February.

Seasonality is quite common in economic time series. It is less common in engineering and scientific data.

If seasonality is present, it must be incorporated into the time series model. In this section, we discuss techniques for detecting seasonality. We defer modeling of seasonality until later sections.

3.1.3 Time Series Analysis

3.1.3.1 Univariate Time series

The term "univariate time series" refers to a time series that consists of single (scalar) observations recorded sequentially over equal time increments.

Although a univariate time series data set is usually given as a single column of numbers, time is in fact an implicit variable in the time series. If the data are equi-spaced, the time variable, or index, does not need to be explicitly given. The time variable may sometimes be explicitly used for plotting the series. However, it is not used in the time series model itself.

3.1.3.2 Approaches to Time Series Analysis

a) Triple Exponential Smoothing

Typically single and double exponential smoothening do not take care of seasonality information. A third method called triple exponential smoothening is used to take care of seasonality. The resulting set of equations is called the "Holt-Winters" (HW) method. The basic equations are:

$$S_t = \alpha \frac{y_t}{I_{t-L}} + (1-\alpha)(S_{t-1} + b_{t-1}) \qquad \text{OVERALL SMOOTHING}$$

$$b_t = \gamma(S_t - S_{t-1}) + (1-\gamma) b_{t-1} \qquad \text{TREND SMOOTHING}$$

$$I_t = \delta \frac{y_t}{S_t} + (1-\delta) I_{t-L} \qquad \text{SEASONAL SMOOTHING}$$

$$F_{t+m} = (S_t + m b_t) I_{t-L+m} \qquad \text{FORECAST}$$

b) Autoregressive models

A common approach for modeling univariate time series is the autoregressive (AR) model:

$$X_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + A_t$$

where X_t is the time series, A_t represent normally distributed random errors, and ϕ_1, ..., ϕ_p are the parameters of the model, with the mean of the time series equal to

An autoregressive model is simply a linear regression of the current value of the series against one or more prior values of the series. The value of ***p*** is called the order of the AR model.

c) Moving Averages models

Another common approach for modeling univariate time series models is the moving average (MA) model:

$$X_t = \bar{X} + A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \ldots - \theta_q A_{t-q}$$

where X_t is the time series, $\bar{X}$ is the mean of the series, A_{t-i} are random shocks to the series, and θ_1, ... , θ_q are the parameters of the model. The value of q is called the order of the MA model.
T hat is, a moving average model is essentially a linear regression of the current value of the series against the random shocks of one or more prior values of the series. The random shocks at each point are assumed to come from the same distribution, typically a normal distribution, with constant location and scale

3.1.4 ARIMA Model

The Box-Jenkins ARMA model is a combination of the AR and MA models described above.
The basic equations are:

$$X_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \ldots - \theta_q A_{t-q}$$

where the terms in the equation have the same meaning as given for the AR and MA model.

Some interesting properties of the model are as follows:

1. The Box-Jenkins model assumes that the time series is stationary. Box and Jenkins recommend differencing non-stationary series one or more times to achieve stationarity. Doing so produces an ARIMA model, with the "I" standing for "Integrated".

2. Some formulations transform the series by subtracting the mean of the series from each data point. This yields a series with a mean of zero. Whether you need to do this or not is dependent on the software you use to estimate the model.

3. Box-Jenkins models can be extended to include seasonal autoregressive and seasonal moving average terms. Although this complicates the notation and mathematics of the model, the underlying concepts for seasonal autoregressive and seasonal moving average terms are similar to the non-seasonal autoregressive and moving average terms.

3.2 Stock Data

3.2.1 Introduction to stock data

The stock prices of various companies recorded over a period of time form a "time series" data. The entities for the stock value analysis are divided into two categories pure technical data and fundamental data. The technical data is the only data used by technical analysts. The fundamental data includes data related to company's activities and market situation along with the technical data

3.2.2 Technical data

The daily available data for each stock is represented in the four time series with data for each day of trading (normally Monday-Friday):

yC or y : close value; price of the last performed trade during the day
yH : highest traded price during the day
yL : lowest traded price during the day
V : Volume; total number of traded stocks during the day

The most obvious choice of entity to predict is the time series yC or y. Some of the drawbacks of this approach are:

The prices y normally vary greatly and make it difficult to create a valid model for a longer period of time. y for different stocks may easily differ over several decades and therefore cannot be used as the same type of input in a model.

3.2.3 Fundamental Data:

Apart from the daily sampled data described above, there is a lot of information concerning the activities and financial situation of each company. Most of the companies, quoted at the stock market, are analyzed on a regular basis by the professional market analysts at the financial institutes. The analyses are often presented as numerical items, which are supposed to give hints on the "true" value of the company's stock.

Some of the important fundamental data include:

a) P/E ratio

A company's current P/E ratio reflects the stock price divided by current earnings per share. The higher the P/E ratio, the more an investor pays for a company's earnings and the greater the expectations for future earnings growth. If a stock trades at a P/E ratio of 10, it means that the company would take 10 years to pay for itself from its earnings.

b) EPS

These are the basic earnings per share (EPS) before extra ordinary items from the latest annual. EPS equals net earnings (or profit) before extraordinary items, less preferred share dividends, divided by average shares outstanding. EPS shows earnings available to each common share and is an important element in judging an appropriate market price of a share.

c) Beta

An important measure of a stock's (or a portfolio's) volatility in relation to the Standard & Poor's 500, which by definition has a beta of 1.0. A beta higher than this implies greater volatility than the overall market. Thus, a stock with a beta of 1.5 will move up 15 percent when the market rises 10 percent. In good times, high betas imply high returns, since a beta above 1.0 amplifies the market's movements. In bad times, of course, a beta below 1.0 is desirable.

d) Dividend

The distribution of corporate earnings to shareholders. This is also a good measure of the companies' performance.

3.3 Rule Discovery in Time series

3.3.1 Introduction

This is a problem of finding rules relating patterns in one time series to patterns in other time series. The method is based on discretizing the the sequence by constructing subsequences by sliding a window through the time series and then clustering these subsequences. Once the time series is discretized simple rule finding methods can be applied to obtain rules from the sequence.

3.3.2 Time series discretization by clustering

The method proceeds by discretizing each of the pair of time series among which the rules are desired. The technique slides a window of size "w" thru the existing time series to get subsequences of size "w" each of which can be considered as a point in w-dimensional space. If there are "n" point in the time series the number of subsequences obtained after windowing is "n-w+1"

These points are then clustered using any of the standard clustering algorithms (for the project we have implemented the greedy clustering algorithm explained in the paper mentioned in the reference) in order to get certain clusters, which can be considered as the basic shapes of the time series. This means that the entire time series can be represented using only these basic shapes. Thus we achieve discretization. The above process is repeated for both the time series and basic defining shapes in each of the series are obtained.

3.3.3 Rule discovery

The next and the main step of the process is trying to find interesting probabilistic rules between the clusters in the two time series. These rules are of the form:

"If a cluster A occurs in time series 1 then we can say with confidence c that B will occur in time series 2 within time T"

This can be represented as $(A \rightarrow B)\ (T)$.

The confidence of this rule is give by:

$$c((A \rightarrow B)(T)) = F((A \rightarrow B)(T))/F(A)$$

where F(A) is the frequency of cluster A in time series 1 and $F((A \rightarrow B)(T))$ is the number of occurrences of cluster A in time series 1 which are followed by an occurrence of cluster B in time series 2 in time at most T .

The entire technique is based on the selection of a number of parameters such as the window size "w", the parameter used in clustering, the time "T" in the above rule etc. The optimum values of which could be obtained only after empirical results.

4. Literature Review

There is a large volume of literature that was found dedicated to stock market analysis and prediction in time series that was extremely useful and relevant to our work in this project. The review gave us a useful insight into the working of stock markets and about the various parameters that affect the value of the stock. A lot of this material was covered extensively in [1] and provided us with the necessary background that was needed to be able to perform analysis and experiments with stock market data. The paper also described about the various approaches that are typically such as time series analysis and machine learning techniques. It also details the various types of fundamental and technical data theta affect/influence the value of the stock. We find that the main fundamental features that needed to be considered are the P/E value, EPS, dividend and the beta value for a particular stock. There are many other fundamental features that are present for every stock value that may also be a part of the feature vector. The technical features that were considered based on the details in [1] and available data that we could extract in [2] were the Volume, high, low and the closing values. A detailed definition and significance of all the features can be found at [14]

Each of the features of the stock is modeled as a time series and the stock price itself is modeled as a separate time series. [3] Details various models that are used to represent a time series such as the autoregressive models and box Jenkins approach. The model that we chose is ARIMA. From our initial experimentation, as discussed in section... we found that the ARIMA model is able to trace the stock time series to a high degree of accuracy. The approach, as discussed in [3] involves modeling the time series as two parts. One corresponds to the Auto Regressive part and the other corresponds to the moving averages part. The order of the model was decided by experimentation and was fixed at p=2, q=2. This resource also discusses schemes for multivariate ARIMA techniques.

Another interesting resource was [4] that talked about the multiple instance stock market prediction problems that deals with modeling the behavior of stocks in order to make efficient decisions about when to sell or buy a particular stock. This speaks of the importance of fundamental and financial indicators and their influence on the fluctuations on the price of the stock.

We referred to [5] for details on various similarity metrics to be used for time series data analysis. The paper provides a detailed description on similarity in time series and provides a model that takes into account the influence of outliers and different scaling functions.

Another approach discussed by [10] is to reducing the dimensionality in time series is by use of Fourier Transformations and using the Discrete Fourier transformations to select those values of the flourier coefficient that represent the maximum information [6] discusses the details of using Fourier techniques and locate matching subsequences in a collection of time series. A fast way of performing the Fourier transformation is using the FFT algorithm. This algorithm is discussed in detail in [7] The use of Fourier transformations and various similarity measures were also detailed in [9]. This paper also spoke of using the sum of the squared distances as a measure of the distance between two sequences. We also adapted this approach on the features after they were reduced from the initial set.

5. Technical Approach

The ARIMA (Auto Regressive Integrated Moving Averages) model is frequently used for modeling univariate time series data. It essentially studies the time series and tries to capture its structure and behavior. The output of the ARIMA model is a set of parameters of both the Auto Regressive and Moving Averages model such that any point in the time series can be computed using these parameters and a number of previous points in the time series. Thus the ARIMA model is a good and compact representation of the time series.

If a pair of time series is “similar” i.e. if both of them are showing similar trends in terms of seasonality and other properties then the ARIMA model is bound to capture the two time series in a similar fashion. This seems to suggest that the values of the parameters for the models generated for the two time series would be very much equal to each other. If we were two treat the two models as points in Euclidean space characterized by the parameters of the model, then the distance between the two points would be very small. Similarly for a pair of time series that are completely different from one another, the ARIMA model would generate completely different parameters and the distance between the two models in this case would be large. This concept can be used to perform some kind of similarity-based operation such as clustering on a number of time series in order to group the similar ones together.

Thus the approach that we propose in this project is as follows:
Given a set of time series data, in order to find the interrelationship among pairs of time series an intelligent way would be to build an ARIMA model for each of the time series and then perform some kind of “metal level” mining on these models such as clustering to group all the similar time series together.

For the first part of our project we will be using this approach to perform a feature selection on the stock data. We will be initially building an ARIMA model, one for the stock price and for each of the features associated with the stock price. On top of this model we will be performing Euclidean distance-based computation in order to find all those feature vector time series that are very close and hence similar to the stock price time series.

In the second part of our project we plan to find how exactly each of the features in the subset, generated in the first part, would influence the stock price. For this purpose we plan to work with the actual time series data in order to mine interesting rules between the feature vector and the stock price.

6. Distribution of work among the project members

This project was a result of combined effort by all the team members who contributed equally in every phase of the project. The first phase involved data collection. Initially the data that we wanted to work on was the fundamental data. The code for collecting the data from the Internet was written by Shailesh Doshi and Akshay Java. The problem that we faced was that the fundamental data had to be downloaded in real time and the data does not change over a short time frame. Vishal Shanbhag was responsible for modifying the code in view of the changed approach and downloading the stock values for a set of companies. The next phase was building a model using ARIMA in semstat, this was done by Shailesh Doshi and Vishal Shanbhag. Akshay Java was responsible for the coding that was required using MATLAB for performing the correlation analysis and doing similarity based measures. The Association rule learning modules were programmed by Vishal while the clustering parts were sone by Akshay. The experimenation and evaluation was performed by Akshay and Vishal.

7. Experimental Results and Discussions

7.1 Data Collection and Preprocessing

7.1.1 Technical Difficulties

For our project we had initially decided to pick up one particular company and get the technical as well as the fundamental data for it, over a long enough period of time, so as to perform our analysis. Although we managed to get the technical data easily, the fundamental data was not available online. Therefore for our experiments we decided to collect the stock price data for a number of companies to get multiple time series. We randomly picked up 1 company data as the as the "stock price" time series and each of the remaining company data as the "feature" time series.

7.1.2 Data Extraction and Cleaning

Our main source of stock data was "http://finance.yahoo.com". We used a Perl script to extract historical stock prices of about 20 companies from the time period: April 2000 to November 2001. The size of each time series was 398 data points resulting in a total of 7960 data points.

The crude data that we extracted from Yahoo needed cleaning. First of all the data that we got was not univariate, it had a total of 6 fields for each data point in the time series. These included the date, the previous day closing price, today's high, low, closing price and volume. We had to extract just the closing price column from this data. This and most of our Data Preprocessing was done using MATLAB. Second, the data that we extracted was not in the correct order. We had to reverse the time series so that it represented data collected over increasing period of time. For all our analysis we used normalized data. Normalization was done using MATLAB inbuilt functions.

7.2 Data Modeling and Analysis.

We had proposed a novel idea of comparing two time series by comparing their corresponding time series models. The correctness of this approach was an interesting question. For this purpose we decided to perform simple correlation analysis to act as a benchmark to test our model.

7.2.1 Time series analysis

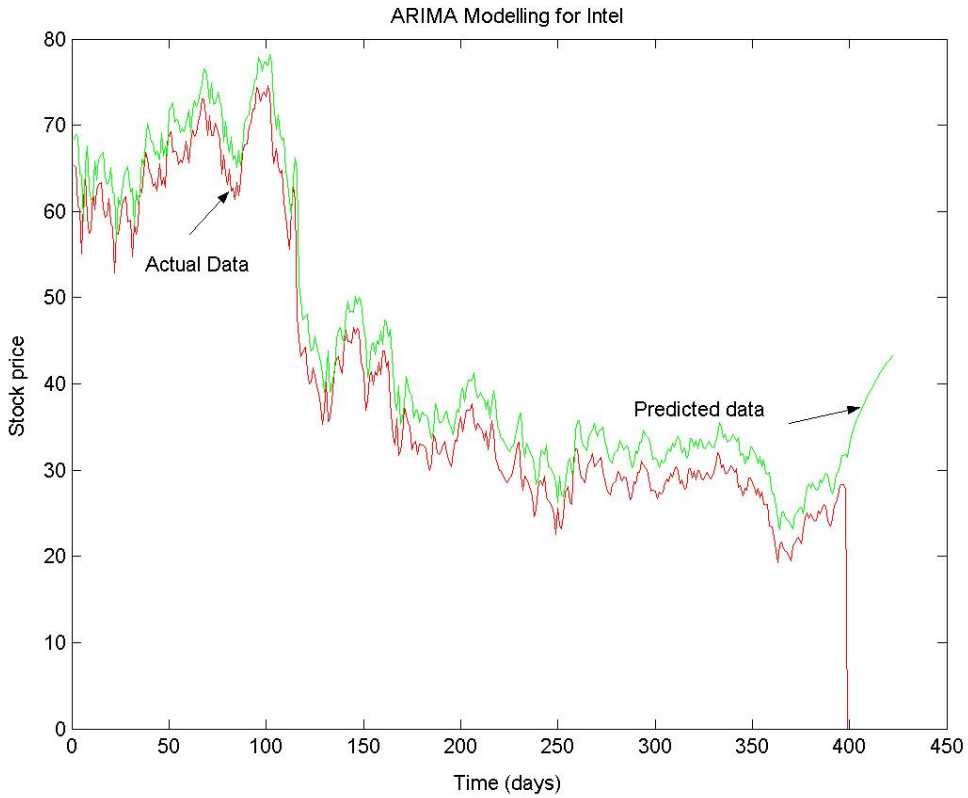
This was done using the Standard Statistical software, SAS. SAS provides a facility to load data into it and then gives standard procedures to compute the ARIMA model. One of the parameters to the procedures is the values p and q, the orders of the Autoregressive and Moving Averages model respectively. Now our initial task was the selection of these parameters. We wanted to select the model that will give the best performance. Usually second order ARIMA model is preferred because it usually gives a very good modeling of the time series. A model that performs well is usually the one with lowest AIC value.

We ran the procedure on 10 different time series each time varying the values of "p" and "q" and studied the performance of the models on each of these. The following table gives the AIC values for the different cases:

Table 1

Sr. No.	Company	AIC Values			
		P=2, q=2	P =1, q =1	P=2, q=1	P=1, q=1
1	erts	1679.858	1678.684	1681.55	1692.023
2	hgsi	2200.928	2199.625	2202.078	2202.868
3	ibm	1987.472	1985.533	1985.731	1984.551
4	msft	1685.608	1684.195	1684.479	1682.196
5	novl	660.09	653.59	657.21	652.52
6	orcl	1367.001	1365.006	1370.531	1371.649
7	sun	1537.276	1536.26	1535.471	1535.045
8	yhoo	2174.8	2166.392	2171.774	2170.556
9	cnet	1416.272	1412.51	1411.042	1411.065
10	palm	1706.406	1705.629	1705.79	1703.814

From the table we observe that the AIC values don't change considerable for different values of p and q and since the AIC values for the second order model is fairly low we chose to use p=2 and q=2 for the remaining part of our analysis. Following is the plot of how closely the second order ARIMA model approximates the data points:



Graph 1

For the next part of our analysis we modeled each of the company data as an ARIMA model and collected the model parameters thus obtained. So for every company we have a representative vector of four parameters thereby reducing the sample space of 7960 points to a set of only 80 points on which we will be performing further analysis. The following table shows the values of the parameters that we got as part of the modeling:

Table 2

Sr. No.	Company	AR1	AR2	MA1	MA2
1	dell	1.266	0.26604	0.28812	0.05454
2	intel	1.4092	0.40942	0.42202	0.06659
3	appl	0.88915	0.11085	0.00511	0.06796
4	ge	0.72618	0.26965	0.21574	0.12397
5	abgx	0.74906	0.23535	0.26709	0.13683
6	bgen	0.30476	0.59662	0.61668	0.09793
7	vtss	1.3879	0.38787	0.44173	0.02308
8	amzn	1.3064	0.30645	0.26735	0.21493
9	cien	0.21759	0.76182	0.82901	0.00589
10	ebay	1.2167	0.21938	0.36088	0.03567
11	erts	1.2605	0.26059	0.38361	0.16742
12	hgsi	1.3353	0.34243	0.33507	0.12357
13	ibm	1.1088	0.11934	0.20756	0.03759
14	msft	1.5008	0.50204	0.46818	0.04966
15	novl	0.2035	0.7965	0.79292	0.0564
16	orcl	0.94806	0.05194	0.05009	0.15431
17	sun	1.5115	0.51153	0.57258	0.04202
18	yhoo	0.47747	0.52253	0.52461	0.15808
19	cnet	1.7773	0.77734	0.79654	0.01836
20	palm	1.6772	0.67718	0.66109	0.05887

The third and most important part of this stage was to do meta-level mining on these models that we had built upon. The strategy that we used was to consider these parameters as points in 4 dimensional space and to use Euclidean distance as a measure of similarity or proximity between a pair of points. We computed all possible inter-point distances using this measure. Based on these distances we decided to find all those pairs of time series that are "closely" spaced and also all those pairs, which are "far" away from each other. For this purpose we fixed a couple of threshold values "t1" and "t2". All those pair-wise distances below the threshold t1 signify that the respective time series are "close" to each other and those above the threshold t2 signify that the respective time series are "far" away from each other. We performed the experiments on the data by varying the value of t1 and t2 and computing the closest and farthest pairs. Following table shows the results of our analysis.

7.2.2 Correlation analysis

As part of this step we computed the cross correlation coefficients for all the pairs of time series that we had. We used MATLAB to help us do this and came up with a 20x20 matrix consisting of the pair wise correlation coefficients. A pair of companies with high correlation coefficient signifies that the two companies are strongly correlated and similarly if the correlation coefficient is low then the pair is not correlated.

7.2.3 Testing our model

We decided to compare the results obtained in section 6.2.1 with the results of correlation analysis performed in section 6.2.2. For all those pairs of time series, which were “close” and “far” to each other, we checked the corresponding correlation coefficients. Our results of modeling would be correct if the correlation coefficients are inversely proportional to the inter point distances. This means that if for a pair of close time series the cross correlation coefficients are high then the results are good. A similar proposition can be made for those pairs of companies those are "far" away from each other. We got some positive results as shown in the tables below:

Table 3.1

Table showing some of the entries for closely spaced companies, when t1 = 0.25 and t2 = 1.0

Company i	Company j	Euclidean Distance	Correlation Coefficient
1	2	0.24318	0.50802
1	7	0.23296	0.54597
1	8	0.17153	0.85451
1	10	0.10132	0.53029
1	11	0.14806	0.57169
1	12	0.13268	0.13064
1	13	0.23026	0.44968
2	7	0.056592	0.84137
2	11	0.23648	0.68815
2	12	0.14408	0.39705
2	14	0.13921	0.11827
2	17	0.21015	0.78378
3	16	0.12814	0.78109
4	5	0.067098	0.80616
6	18	0.21775	0.1274
7	11	0.23799	0.52952

Table 3.2

Table showing some of the entries for companies that are far away, when t1 = 0.25 and t2 = 1.0

Company i	Company j	Euclidean Distance	Correlation Coefficient
1	6	1.6092	0.12982
1	9	1.2806	0.23929
1	15	1.2904	0.75006
2	6	1.1375	0.18324
2	9	1.309	0.75058
2	15	1.3196	0.63529
3	9	1.248	0.78103
3	15	1.2494	0.59471
3	19	1.3645	0.76818
3	20	1.1714	0.76063
4	19	1.3081	0.76818
4	20	1.1283	0.76063
5	19	1.2828	0.69605
5	20	1.1036	0.88769
6	7	1.1193	0.31802
6	8	1.106	0.001686
6	10	1.0214	0.11884
6	11	1.0419	0.16459
6	12	1.0985	0.26877

From the above table we can make out that in most of the cases our model is performing in accordance with the correlation analysis. We repeated the above experiments with t1 = 0.35 & t2 = 0.9 and also with t1 = 0.45 and t2 = 0.8. We annotate these cases as “Case 1” “Case 2” and “Case 3” respectively. We more or less got similar results with one observation that as you shrink the interval more and more the number of entries in the table increases.

We then computed the error in our model counting all those instances where our model misclassified a pair of time series as being "close" or "far" to each other when compared with the Correlation analysis. In order to do this we had to set certain threshold, which signifies how bad a misclassification is. Let “p1” be the threshold such that if the correlation coefficient, for a pair of time series classified as “close”, is less than p1 then the pair has been misclassified. Similarly we can define “p2”. We performed experiments with 3 sets of thresholds. These are the results that we found out:

E1 = Error in classifying a “closely” located pair.
E2 = Error in classifying a “far” away pair.
C = Combined misclassification error.

Table 4.1

p1 = 0.5, p2 =0.5

Case	E1	E2	C
1	48%	60%	54%
2	45%	57%	51%
3	43%	58%	50%

Table 4.2

p1 = 0.35, p2 = 0.65

Case	E1	E2	C
1	16%	49%	41%
2	27%	44%	36%
3	31%	48%	39%

Table 4.3

p1 = 0.3, p2 = 0.7

Case	E1	E2	C
1	31%	41%	36%
2	24%	37%	30%
3	29%	39%	34%

From the table 4.3 we observe that the best success rate that we could get was about 70% whereas the average success rate that we got was about 60% as compared to the correlation analysis.

7.3 Rule Discovery

7.3.1 Feature Selection

From the experimental results obtained so far we can define our model as follows:
Given a set of time series, as input to it, the model finds with high degree of confidence all those pairs of time series that are highly correlated to each other.

We can now use this model to try and solve the very problem that was stated in section 2, i.e. given stock price data for a particular company we can find all those feature vectors that strongly influence it.

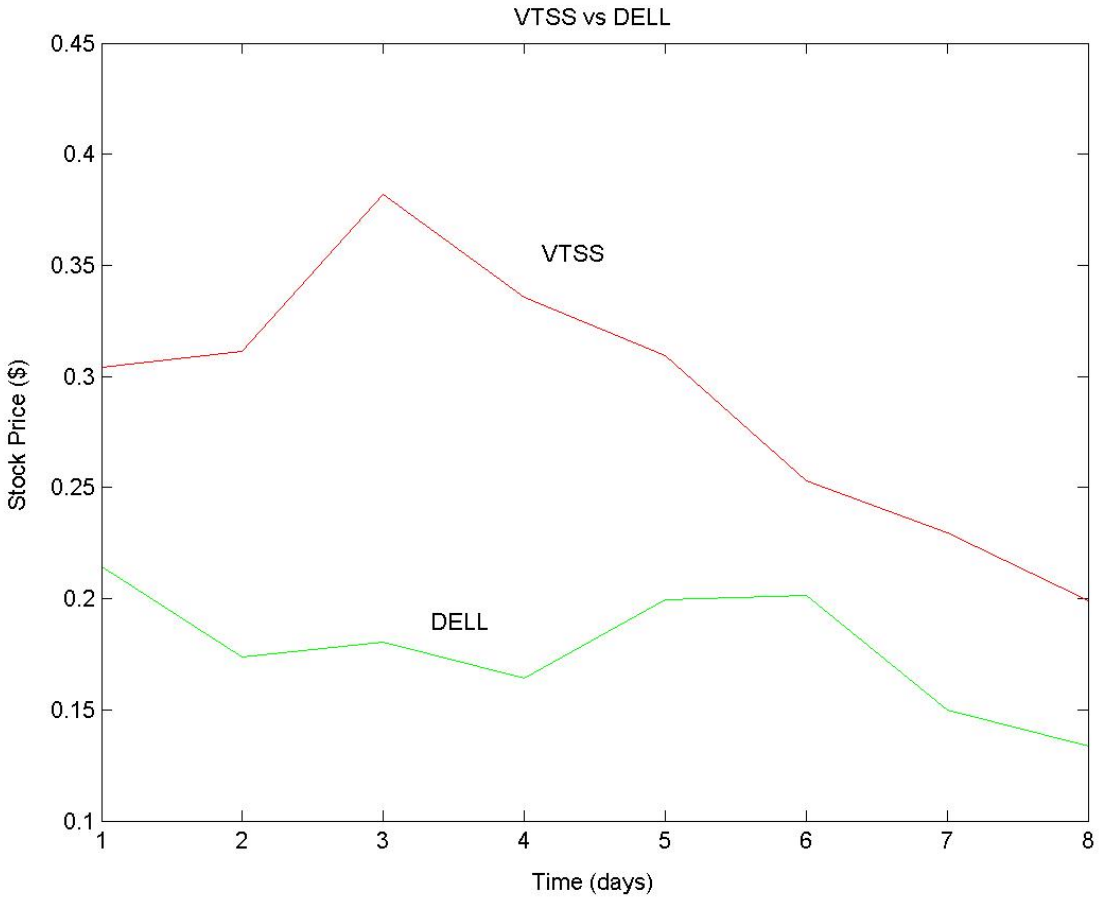
For our test case we had stock prices of 20 different companies. If we treat any one of them as being the actual stock price data of a hypothetical company and the remaining time series as being the feature vectors associated with this stock price, then we can find all those features that are strongly correlated to the stock price. For e.g. if company 1 is chosen as the stock price then from table 3.1 we can say that features influencing it are the companies 2,7, 8,10,12,13.

Thus we have achieved feature selection. Now we can say that in order to find interesting rules between the feature vectors and the stock price we only need to consider those features that are selected by the above model.

7.3.2 Extracting rules from the subsets

For our experiments we chose to find rules between company 1(Dell) and company 7(VTSS). Our objective was to find the influence of VTSS on DELL. We implemented the code for this, based on the description given in [15]. We fixed a window size of 8 and the cluster radius of 0.5 units and a time frame of 15 days to get a number of interesting rules. One of them can be shown as follows:

Graph 2.



The above rule could be interpreted as follows:
"If the VTSS stock price follows the pattern shown above in a period of 8 days then we can say with 100% confidence that in a span of 15 days, the Dell stock price will follow the pattern as shown"

8. Conclusions

The project has sought to find a novel approach in time series analysis by using meta-level data mining techniques. The idea that we wish to emphasize here is that we can reduce the computational expense involved in large time series analysis by using techniques such as ARIMA and using the parameters of these models in order to do feature selection. The experiments with this approach have shown some encouraging results. We found that even by using a simple measure such as Euclidean distance between the parameters of the ARIMA models of each of the features we were to get a good estimate of its correlation. We compared the performance of our model with simple correlation analysis and found almost 70% accuracy.

References

[1] *Predicting the stock market* Thomas Hellstrom and Kenneth Holmstrom

[2] WSRN http://www.wsrn.com

[3] *Engineering statistics handbook* http://www.itl.nist.gov/div898/handbook/index.htm

[4] http://www.ai.mit.edu/people/oded/thesis/node8.html

[5] *Finding Similar Time Series* (1996) Gautam Das, Dimitrios Gunopulos, Heikki Mannila

[6] *Fast Subsequence Matching in Time-Series Databases (1994)* Christos Faloutsos M. Ranganathan Yannis Manolopoulos

[7] *An introduction to Fourier theory* http://aurora.phys.utk.edu/~forrest/papers/fourier/index.html

[8] *Fast Similarity search in presence of noise scaling and translation in time series databases* R agarwal, K lin, H S Sawhney K Shim

[9] *Efficient Similarity Search in Time Series Databases* Rakesh Agarwal, Christos Falutsos, and Arun Swami, FODO conference, Evanston, Illinois,

[10] *A survey of recent methods for efficient retrieval of similar time series* Magnus Lie Hetland Norwegian University of science and technology

[11] *Mining association rules between sets of Items in large databases* Rakesh Agarwal Tomasz Inmielinski Arun Swami IBM Almaden research center.

[12] *Nasdaq* http://www.nasdaq.com

[13] *Yahoo finance* http://chart.yahoo.com

[14] *Money Central.com* http://moneycentral.msn.com/investor/glossary/

[15] *Rule Discovery in Time Series* Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, Padhraic Smyth.