CORRELATION BETWEEN FUNDAMENTAL VALUES IN THE FINANCIAL MARKET

by

Rui Liu

B.S., Beijing University of Posts and Telecommunications, 2003

A Thesis
Submitted in Partial Fulfillment of the Requirements for the
Master of Science Degree

Department of Computer Science
in the Graduate School
Southern Illinois University Carbondale
May 2006

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

THESIS APPROVAL

CORRELATION BETWEEN FUNDAMENTAL VALUES IN THE FINANCIAL MARKET

By
Rui Liu

A Thesis Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of Computer Science

Approved by:

Mehdi Zargham, Chair

Wen-Chi Hou

Norman F. Carver III

Graduate School
Southern Illinois University Carbondale
December 15th, 2005

AN ABSTRACT OF THE THESIS OF

Rui Liu, for the Master of Science degree in Computer Science, presented on December 15th, 2005, at Southern Illinois University Carbondale.

TITLE: CORRELATION BETWEEN FUNDAMENTAL VALUES IN THE FINANCIAL MARKET

MAJOR PROFESSOR: Dr. Mehdi Zargham

In the financial market, there are usually two methods of data analysis: the technical analysis and the fundamental analysis. In this thesis, we use the latter method and pay attention to the fundamental variables.

In the stock market, there is a large amount of data and fundamental variables associated with it. In our approach, seven variables are selected and prove that no linear correlation exists between any two of them. Then, fundamental variable pairs are constructed from these fundamental variables and their visualization figures are built. From the results of the visualization figures, a statistical method is used to find the sub-areas with high frequency. With the help of these high frequency sub-areas, we observe all the visualization figures for every variable pair. Observations show that a sub-area in which the data samples always have a good or bad return from any variable pair can not be found. However, we can find sub-areas in which the return of stocks is better than the average return of all stocks. Based on these sub-areas, a set of rules is derived using the training data sets from 1993 to 1998. These rules are tested on the data set from 1999 to 2003. Most of the rules perform well because, except

for rule 4 in year 2001, the average of returns of the rules are better than the average of returns of all stocks in S&P 500.

ACKNOWLEDGMENTS

First and foremost, I would like to express my most sincere thanks to my thesis supervisor, Professor Mehdi Zargham for his inspiring guidance and unlimited support throughout my graduate studies and research. Without his enthusiasm, encouragement and patience, this thesis would not have reached its current form.

I am also extremely grateful to Professor Wen-Chi Hou and Professor Norman Carver for serving on my thesis committees. Their advice and contributions are very precious and sincerely appreciated.

Special thanks are also extended to Ms. Georgia Norman and Ms. Monica Russell who helped me a lot even before I arrived at Southern Illinois University，Carbondale.

Last, but not least, I would like to thank my family for their love, for their always being there when I need them, and for their unreserved support.

TABLE OF CONTENTS

Chapter Page

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Stocks are among the most talked about and most popular investment opportunities available. Investors are always looking for a better way to find the right one to buy among the thousands of stocks. Two methods of data analysis have emerged to assist investors in making better investment decisions: one is technical analysis and the other is fundamental analysis. Technical analysis relies on the market data and does not consider the intrinsic value of the securities. It believes that the future price of a stock can be predicted by looking at its historical prices and other trading variables. Unlike technical analysis, fundamental analysis disregards the overall state of the market. Instead, it takes into consideration only those variables that are directly related to the company itself, such as its earnings, its dividends, and its sales.

There are many methods that can be applied to the stock market, such as decision tree (DT), fuzzy decision tree (FDT), machine learning (ML), artificial neural networks (ANN) and committee machines (CM). Decision tree is a popular method in securities analysis. Several important algorithms CART, ID3 and C4.5 are used in stock market for stock prediction. Fuzzy decision tree combines the uncertainty handling and approximate reasoning capabilities to enhance the representative power of decision tree methods. Along with the decision tree method, machine learning approaches, such as Bayesian methods, are also used as technical analysis to predict future value of securities. Artificial neural networks, such as artificial intelligence (AI) methods, are very important tools and can be applied to stock performance prediction and stock price prediction. Committee machines, which combine several so-called experts (neural networks, decision trees, or some other kind of learning algorithm); can be used to achieve better performance by coordinating these methods.

1.1 Problem Statement

Although the trend of the stock market is hard to predict, it does not mean the prices of stocks are randomly generated. Certainly, there are hidden rules associated with the stock performance in the financial market. These rules are very helpful for investors to select right stocks if we can extract them from huge amounts of securities data. Some software, such as PORSEL (PORtfolio SELection system), has already been developed to select and evaluate stocks based on the derived rules [42].

Due to the importance of rules, researchers have proposed different methods to extract them. More than thirty years ago, Benjamin Graham, the father of security analysis, had already outlined his ten investment rules for stock selection. Graham’s rules performed very well before the year 1976. With the fast development of the financial market, Graham’s rules are no longer satisfactory and their performance has decreased substantially. As the financial market gets more and more complex, people usually perform rule extraction based on recent computational methods, such as decision tree (DT), fuzzy decision tree (FDT), machine learning (ML), artificial neural networks (ANNs) and committee machines(CM).

The methods discussed above may have satisfactory results within some period when applied to the stock market, however, each of them has its own limitation. Decision tree is not efficient at dealing with continuous attributes, while many attributes in the stock market are continuous. The drawback of fuzzy decision tree is that it is hard to select the best fuzzy terms and to design them appropriately. The machine learning method works only for symbolic inputs, so continuous features must be made discrete first. Artificial neural networks are hard to train and may not reach to a good solution in the end. Committee machines are difficult in combining results from different learning methods. To avoid these limitations, in this study we will use a method to extract rules which is based on statistical approaches and data visualization.

1.2 Proposed Work

The main goal of this study is to extract rules from securities data in the financial market. Securities data representing the fundamental variables will be used in our simulation. The data ranging from 1993 to 1998 are going to be used to extract rules. First we use statistical methods to show that there are no linear correlations between any fundamental variables; then we visualize the securities data to get nonlinear correlations between fundamental variable pairs; from the visualization results, we can extract rules which can be used in the selection of stock. Lastly, the data ranging from 1999 to 2003 are used to test the performance of the rules. We will select the stocks based on the rules and see whether the stocks have a higher return than the average return of all stocks in S&P 500.

The rest of the thesis is organized as follows: In Chapter 2 we briefly review some general approaches for rules extraction and their applications in the area of securities analysis. In Chapter 3 we propose a methodology to visualize data sets into data points and provide a criterion to color these data points. In Chapter 4 we study the visualization figures. Then we derive rules based on visualization results and test performance of these rules. Overall summary, conclusions and recommended study are presented in Chapter 5.

CHAPTER 2

PERTINENT WORKS

Stocks and the stock market are attractive for investors who expect a high return for their investment. But in most cases they may hardly gain what they expect, one important reason for this is that they know little of the hidden rules which lie behind the large amount of stock data. The most famous rules for selecting stocks are attributed to Benjamin Graham [1] and are as follows:

1) An earnings-to-price yield of twice the AAA bond.
2) A P/E ratio down to four-tenths of the highest average P/E ratio the stock attained in the most recent five years.
3) A dividend yield of two-thirds the AAA bond yield.
4) A stock price down to two-thirds of tangible book value per share.
5) A stock price down to two-thirds of "net current asset value".
6) Total debt less than tangible book value.
7) Current Ratio of two or more.
8) Total debt equal or less than twice the "net current asset value".
9) Earnings growth over the most recent ten years of 7% compounded.
10) Stability of growth in earnings, defined as no more then two declines of 5% or greater in year-end earnings in the most recent ten years.

These rules performed well before the year 1976. However, the stock market has had dramatic changes since 1976, making these rules unsatisfactory in today’s stock markets.

In this section we review some general approaches for rule extraction and their applications in the area of securities analysis. We will focus our discussion on four classes of rules extraction approaches: decision trees (DT), machine learning (ML), artificial neural networks (ANN) and committee machines (CM). Several algorithms associated with each approach are reviewed and for each approach, its potential advantages and drawbacks are discussed.

2.1 Rule extraction approaches

There are two kinds of rules based on the expressive power of the extracted rule: one is Boolean rules and the other is fuzzy rules. Boolean rules are expressed using conventional symbolic logic (i.e. two-valued Boolean) in the form of “IF-THEN-ELSE-.” Combining the concepts of fuzzy logic, we can get some fuzzy rules which still use an “IF-THEN-ELSE-” structure but use a membership function to deal with what are termed “partial” truths instead of two-valued logic [2]. For example, if x is good and y is bad then z is ok; where good, bad, and ok are fuzzy sets with corresponding membership functions. There are a large number of rule extraction approaches for different applications. For instance, the decision tree approach is a principal tool in data mining, which has advantages of being highly efficient at data analysis and abstracting rules. However, decision tree lacks practicality because of the large amount of work required in tree growing and tree pruning to obtain understandable rules. Since the decision tree approach is sensitive to the noisy data which typically lies in the stock market, the fuzzy decision tree approach is proposed to handle uncertainty and can give investors intuitional information in stock prediction. In addition, the machine learning approach, which includes several programs (AQ15, CN2, RIPPER, etc.), is also used to extract rules. It is a multi-strategy approach that can deal with noisy data and large samples, as well as potentially obtain a smaller error rate of rules extracted by the decision tree approach. The most frequently used approach to derive rules is artificial neural network. It is suitable for rule generation because this approach performs particularly well in large data set due to its ability to model nonlinear relationships. It is also robust in handling data with noise or missing values due to its inherently parallel data processing technique. However, artificial neural networks are hard to train and may not reach a satisfactory solution. To adopt its advantages as well as those of other learning methods, the committee machine approach is proposed and can combine different decisions of several learning methods (like neural networks, decision trees, or some other kind of learning algorithms), but the difficulty is the criterion to optimally combine these decisions. Next we will review each approach separately and in detail.

2.1.1 Rule generation based on decision trees

Decision tree is a powerful and popular method in data mining. In decision theory, a decision tree is a graph of decisions and their possible consequences, (including resource costs and risks) used to create a plan to reach a certain goal. Rules could be generated easily based on the path in decision trees from the root to individual leaves. The leaf node could be used in the “Then” part and the other nodes of the consequences could be used as the “If” part.

The 1R algorithm leads to the simplest decision rules [3]. The input of this algorithm is a set of training examples and the output is a 1-rule (i.e., 1-level decision tree). Each rule is used to classify an object on the basis of a single attribute. The rule is obtained by choosing a certain interval in which the smallest value of the attribute is larger than those in other intervals. Several of the most widely used algorithms to construct decision trees are: CART [4], ID3 [5] and C4.5 [6]. CART [4] analysis is a tree-building technique which is ideally suited to the generation of decision rules. The limitation is that it is computationally very expensive because it requires the generation of multiple auxiliary trees. ID3 was first presented by Quinlan in 1986 [5]. Its purpose is to construct a good decision tree without much computation. The basic structure of ID3 is iterative. A subset of the training set is chosen at random and a decision tree formed from it; this tree correctly classifies all objects in the training subset. All other objects in the training set are then classified using the tree. If the tree gives the correct answer for all these objects, then it is correct for the entire training set and the process terminates. If not, a selection of the incorrectly classified objects is added to the training subset and the process continues. In this way, correct decision trees have been found after only a few iterations. An extended algorithm called C4.5 was proposed by Quinlan in [6]. Compared to ID3, it addresses many additional problems such as: handling continuous attributes, dealing with training data that have missing attribute values, improving computational efficiency, etc.

Decision trees provide us with an accurate and efficient way to generate rules. However, the limitation of this approach is that it is sensitive to noisy data.

Therefore, fuzzy decision trees, which combine uncertainty handling and approximate reasoning capabilities, are introduced to solve this problem. Fuzzy rules could be extracted from fuzzy decision trees. The authors of [7] propose a fuzzy decision tree induction method which is based on the reduction of classification ambiguity with fuzzy evidence. In [8] the author presents a complete method for building the fuzzy tree, and a number of inference procedures based on conflict resolution in rule-based systems and efficient approximate reasoning methods. In [9] a way to get discrete values from continuous attributes in the process of generating decision trees is considered and a better way to express fuzziness via fuzzy numbers is presented using probability theory. The authors of [10] use a different approach to get discrete values from continuous attributes via context-based clustering. Compared with decision tree approach, the fuzzy decision tree approach enhances the representative power of decision trees naturally with the knowledge component inherent in fuzzy logic, leading to better robustness, noise immunity, and applicability in uncertain/imprecise contexts.

2.1.2 Machine learning approaches for rule extraction

Another class of approaches used frequently in rule extraction is the machine learning (ML). Machine learning is the ability of a device to improve its performance based on its past performance. It has produced many learning algorithms but most of them work only for symbolic inputs, so continuous features have to be made discrete first.

AQ type learning programs [11, 12], which have more than 20 versions, is a large family in ML. In AQ concept description, rules for a class are built starting from an example selected for this class. Then a set of most general rules that cover this example and other examples that belong to the same class is generated. Based on the different requirements (like the precision of the rule), several criteria are applied to these rules to select the best rule. Typically, the AQ15 program [12] and several other AQ algorithms were used in a multi-strategy approach to data mining, combining ML, database, and knowledge-based technologies. In additional to AQ type programs, CN2 [13] is a more powerful induction algorithm which combines the efficiency and ability to cope with noisy data of ID3 with the if-then rule form and flexible search strategy of the AQ family. Another efficient ML algorithm is RIPPER [14], which generates rules that cover examples from a given class, adds a new feature similar to the decision tree and selects the best subset from all the training subsets. It is efficient when dealing with large samples and also obtains competitive error rates compared with rules generated by C4.5 model in decision tree approach.

2.1.3 Artificial neural network (ANN) approaches

Compared to decision tree and ML methods, the artificial neural network algorithms have many important advantages, especially when the input is continuous. However, one main limitation of neural networks is that it is hard for humans to understand the meaning of the output data. To solve this problem, many algorithms are designed to extract logical rules describing the data. The algorithms usually belong to two categories: (1) global method, which is based on analysis of the whole network for various inputs to extract rules; (2) local (decompositional) method, which analyzes fragments of the network, usually single hidden nodes, to extract rules.

The global method extracts rules by taking all possible combinations of features as inputs. It is obvious that one main problem with this approach is that the search time increases quickly with an increase in the number of inputs. The authors of [15] restricted the depth of the node-searching process and the search space to accelerate the rule generating speed. But the restrictions made the method generate very general rules. This method was improved in [16] where the authors allowed the search to be performed in various inputs. The validity interval analysis (VIA) [17] is a further extension of the global method. A validity interval which specifies the maximum activation range of each input, may be found using linear programming techniques, i.e., in polynomial time. VIA can also handle continuous-valued input features, by replacing the training values with intervals that are increased to achieve good generalization of the rules. VIA may be applied to any ANN with monotonic transfer functions, however, the rules extracted are too specific.

Local (decompositional) method (i.e., focusing on searching for and extracting rules at the level of the individual units within the ANN) has been adopted to extract Boolean rules [18], [19]. Of particular interest are the algorithms proposed in [20], [21]. In [19], [21], the authors proposed a KT algorithm which maps the output from each unit into a Boolean function. A more recent example of the approach is the SUBSET algorithm which was proposed in

[20]; in this approach the ANN was constructed in a way that the computed value of the activation function in each unit is either “near” a value of one or “near” a value of zero. The algorithms can be applied in some practical systems but the drawback of the algorithms is that the time for finding all possible subsets is exponential with the number of links to each unit. To speed up the rule generating process, the M-of-N concept was proposed in [20] which is a means of expressing rules in the form: if (M of the following N antecedents are true), then … This method forms groups of connections with similar weights rather than the explosion of the number of possible inputs. The M-of-N approach has been tested successfully in different problem domains and in [20], the authors compared the quality of rules extracted using the method with other rule extraction techniques.

In addition to extract Boolean rules from ANN, many techniques have also been proposed to extract fuzzy rules in the so-called neurofuzzy systems.
The authors of [22] developed three types of fuzzy neural works which can automatically identify the fuzzy rules and tune the membership functions by modifying connection weights of the ANN using a back-propagation algorithm. In [23] the authors incorporated knowledge initialization and rule refinement in a fuzzy reference system with a seven layer ANN. For this special structure, it had significant improvement in prediction accuracy of the model compared with a conventional three-layer neural network. The authors of [24] proposed fuzzy-MLP procedure to automatically determine the set of rule antecedents by analyzing

the weight vectors in ANN. This method also had advantages when applied to medical systems compared with conventional neural networks.

2.1.4 Committee machines approaches

Committee machines are combinations of several so-called experts (neural networks, decision trees, or some other kind of learning algorithms) to reach a better overall decision. The general methods used in the committee machines are ensemble methods and boosting methods [25].

The ensemble method requires that independent experts are trained on the same set of data and the outputs from different experts are combined to produce an overall output. One approach to combine the results is to average and the other is to let the experts “vote”, which can be associated with different weights in relation to the accuracy of the training data, to obtain desired output.

Different from the ensemble method, the boosting method requires the experts to be trained on data sets with different distributions. In [26], the authors proposed a method of boosting by filtering. Three experts are incorporated in a committee machine where the first and the second expert are trained on disjointed sets of data and the third expert is trained on the data sets that the first two experts disagree on. This method can boost the low accuracy of a weak learning algorithm. In [27], the author proposed two versions of the algorithm AdaBoost which can be used to reduce the error of any learning algorithm that generates classifiers whose performance are always bad. Experiments are also carried out to assess how well AdaBoost performs on real learning problems.

2.2 Rule generation methods in financial market

In the above, we have reviewed some rule generation approaches for general purposes. In this section we will review some references that use those approaches as well as other approaches to extract rules in the area of securities analysis.

In general, many techniques can be applied in the stock market. These techniques are used in stock prediction and stock performance evaluation. In particular, the decision tree method provides us with a good way to generate “If-Then” rules from data, and has been applied in the stock market by Ren and Zargham [28] to extract rules for stock prediction based on decision tree C4.5. The authors of [29] also applied the decision tree method to extract rules for US IPO data. However, the limitation of this method is that it cannot deal with continuous attributes efficiently. In the stock market, many attributes are continuous so we want to find other methods to deal with continuous semantics. To solve this problem, fuzzy decision tree method was applied by Zheng and Zargham in [30] to derive rules for stock selection. The fuzzy decision tree is an attractive method used to predict stock selection, one disadvantage of it may be the lack of dynamic fitting because the fuzzy sets used in fuzzy decision tree can not be changed after creation. Besides these approaches, machine learning approaches (Bayesian classification, support vector machines, reinforcement learning, etc) are also applied for stock prediction and rule extraction. The authors of [29] used several machine learning approaches to analyze US stock data. In [31] a reinforcement learning algorithm was introduced for stock price prediction. The authors of [32] proposed a fuzzy support vector machines regression method to forecast the stock composite index.

Another very important class of approaches used frequently in securities analysis is the neural networks (NNs). Due to its flexibility and robustness, it has become very important in making stock market predictions and is mostly implemented in forecasting stock prices and returns. In [33], neural networks were used for modeling financial time series. In [34]-[35] the authors proposed neural network approaches to predict the stock market. The authors of [36] used a neural network for rule extraction for analyzing dividend events. In addition, some researchers tend to include novel factors in the learning process for neural networks. The authors of [37] incorporated prior knowledge to improve the performance of stock market prediction. In [38], the authors integrated the rule-based technique and ANN to predict the direction of the S&P 500 stock index futures on a daily basis. In [39], the authors proposed an ANN stock selection system to select stocks that are top performers from the market and to avoid selecting under performers. A fuzzy neural network was also proposed in [40] to extract fuzzy rules.

ANN has a lot of advantages and can be easily combined with other AI approaches in stock prediction. However, it has some limitations in the learning process in the stock market because stock market data has tremendous noise and complex dimensionality. ANN often exhibits unpredictable performance on noisy data. An approach was proposed in [41] to analyze financial markets with noisy information.

As we have shown, all the approaches mentioned above can be applied in the stock market and may have satisfactory results. Different approaches should be applied for different objectives and may even be combined to achieve the best performance in securities analysis

2.3 Summary

In this Chapter, we have reviewed four main rule extraction approaches: decision tree, machine learning, artificial neural networks and committee machines. We also discussed some related work in applying these approaches in the area of securities analysis. Several algorithms associated with these approaches have been reviewed along with some of their advantages and limitations.

Unlike all the approaches mentioned above, in this study we propose a way to predict stock returns based on data visualization. This approach relies on transforming securities data sets into data points as well as plotting them in our visualization figures. By statistical analysis of the data points from all years, we can derive rules for stock selection. In the next Chapter we will discuss our method in detail.

CHAPTER 3

VISUALIZATION DATA USING MATLAB

This chapter shows how to use MATLAB to visualize the securities data. In section 3.1 we present our methods to prepare data for visualization. One of the eight fundamental variables in the securities data is used to generate two attributes about return values; other variables with annual values are directly used whereas the variables with monthly values are treated differently. We also show that there is no linear correlation between any two of our fundamental variables. In section 3.2 we propose ways to visualize the data. First we choose any two of seven variables to form a fundamental variable pair; then we build our visualization figure using one variable of the fundamental variable pair as x-axis and the other as y-axis. Based on the values of the data samples we can transform them into points in our two-dimensional visualization figure. After all the data points are obtained in our figure, we neglect those points which are far away from others, i.e., we are interested in the area where most of the points appear. To do this, we propose a way to determine the ranges of this area with high point density. In section 3.3 we associate different colors to the data points according to different point density levels in order to distinguish the data points more conveniently. In section 3.4, we take the variable pair CR and MKBK as an example and show how to get the visualization result of data set for the year 1994.

3.1 Data collection and preparation

In this research, all data samples are constructed from the S&P 500 database (The database is accessed by using the software S&P Research and Insight in Morris Library, SIUC). The fundamental variables of the securities which could be directly collected from the S&P 500 database are MonthPrice, Book Value Per Share (BKVLPS), Current Ratio (CR), Dividend Yield (DVYDF), Price to Book Ratio (MKBK), Price to Earning Ratio Monthly (PEM), % Earning Growth (EG) and Earning Stability (ES). We use the values of these fundamental variables to construct data sets which will be used in our study later.

3.1.1 Two attributes from MonthPrice: Return and Return Class

In this study, MonthPrice is used to generate two attributes about return values, which are called *Return* and *Return Class*.

Return is used as the main attribute for data samples (a data sample corresponds to a particular company). The definition of *Return* is the MonthPrice of December in that year minus the MonthPrice of December in the previous year and then divided by the MonthPrice of December in the previous year. Assume *MonthPrice (m, y)* means the MonthPrice of month *m* in year *y*, this attribute can be computed in the formula as follows:

$$\mathrm{Re}\,turn(y) = \frac{Month\,\mathrm{Pr}\,ice(December, y) - Month\,\mathrm{Pr}\,ice(December, (y-1))}{Month\,\mathrm{Pr}\,ice(December, (y-1))}$$

This attribute is a parameter for us to see whether stocks increase or decrease in value and to what extent they can be. Note that the values of *Return* are real numbers, which are not convenient for us to classify.

All the data samples in this research must have *Return* values. Clearly, if one data sample does not have either of the values of MonthPrice of December in two successive years due to the data unavailability in the database, we cannot compute its *Return* value from the formula. At this time, this sample will be filtered out at the beginning and not be considered in the following research steps.

In order to classify *Return* values in different groups, we transform them into another attribute called *Return Class.* Unlike *Return*, whose value could be any real number, *Return Class* just has five integer values which are 0, 1, 2, 3, and 4. Each of the values corresponds to a number of *Return* values in different ranges and has different meanings. Following is a table to define each value of *Return Class*.

Table 1: The definition of Return Class

Return Class	Return	Meaning of the Return Class Value
0	Return < 0	Bad
1	0 <= Return < 0.1	Ok
2	0.1 <= Return < 0.3	Good
3	0.3 <= Return < 1	Very Good
4	Return >= 1	Excellent

For example, when the *Return* value of a sample is 0.5, which is in the range of 0.3 and 1, the corresponding *Return Class* of this sample is 3 which indicate that the return of this sample is very good.

3.1.2 Data sets construction

The ultimate goal of this study is to derive rules that are based on the values of fundamental variables to predict the return values in the future. To do so, we construct data sets that include fundamental variables of a certain year and the *Return* (as well as *Return Class*) values of next year. For each year, we have a corresponding data set to represent securities data. Next we will describe how we can build these data sets.

Except for MonthPrice, we require that the all other fundamental variables are annual variables in the data set. An annual variable means that this variable of one data sample has only one value per year. Some fundamental variables such as BKVLPS, CR, DVYDF, EG and ES are annual variables, so their values will be directly used for each year. Other variables like MKBK and PEM are monthly variables. We will choose the value in December as the value of the variable in that year to change them into annual variables.

The following table is an example of a set of data samples. Each row of the data set can be seen as a data sample.

Table 2: An example of data set (1994)

Company Name	1995 R	1995 R-C	1994 BKVLPS	1994 CR	1994 DVYDF	1994 MKBK	1994 PEM	1994 EG	1994 ES
3M CO	0.24	2	8.021	1.922	3.297	3.364	17.05	5.922	5
ABBOTT	0.25	2	2.521	1.115	2.268	6.62	17.44	68.47	5
ACE LT	0.66	3	7.653	@NA	1.75	1.015	@NA	@NA	2
ADC TE	0.64	3	4.153	2.96	0	5.267	34.24	68.79	5
…	…	…	…	…	…	…	…	…	…

The data set shown above is a part of the data samples of the year 1994. The first column is the name of each company (Here we only show the first several letters for short). The second column is the *Return* of each company in year 1995 (for the purpose of stock prediction) which can be directly calculated from the variable MonthPrice of the year 1994 and the year 1995 as follows:

$$\mathrm{Re}\,turn(1995)=\frac{Month\,\mathrm{Pr}\,ice(December,1995)-Month\,\mathrm{Pr}\,ice(December,1994)}{Month\,\mathrm{Pr}\,ice(December,1994)}$$

The third column is the *Return Class* of that company according to the corresponding value of *Return* and the definition in Table 1. The fourth to the tenth columns are the values of other fundamental variables from the year 1994. The fourth column represents the values of BKVLPS (Book Value per Share) which is based on fiscal year end data and represents Common Equity Liquidation Value (CEQL) divided by Common Shares Outstanding (CSHO). The fifth column shows the values of CR (Current Ratio). CR is Current Total Assets, which represents cash and other assets that in the next 12 months are expected to be realized or used in the production of revenue, divided by Current Liabilities Total, which represents liabilities due within one year, including the current portion of long term debt. The sixth column gives the values of DVYDF (Dividend Yield) which is the Dividend Per Share by Ex-Date, defined as the cash dividends per share for which the ex-dividend dates occurred during the reporting period, divided by the company's close price for the fiscal year. The seventh column refers to the values of MKBK (Price to Book Ratio) which is defined as Market Value Monthly divided by Quarterly Common Equity Total, which represents the common shareholder's interest in the company, including common stock, capital surplus, retained earnings and treasury stock adjustments. The eighth column shows the values of PEM (Price to Earnings Ratio Monthly) which is defined as the month-end close price divided by the appropriate 12-Months Moving Earnings Per Share. As said before, the values of Beta, MKBK and PEM are actually the values in December of that variable. The ninth column is the %

Earning Growth (EG). EG for one year is the percentage change in this year's Basic Earning Per Share (EPS) versus last five year's Basic Earning Per Share. Earning Stability (ES) is shown in the tenth column. The definition of ES is the number of years for which the Basic Earning Per Share (EPS) is positive measured over the past five years. In this table, the samples without *Return* values have already been deleted. For other samples which have *Return* values but their fundamental variable value may be not available, we use symbol "@NA".

Similarly, we can construct other data sets based on fundamental variables. All data sets we have in this study are from the year 1993 to the year 2003. The six data sets between years 1993 and 1998 are used as training data sets, while the five data sets after 1998 are used as test data.

3.1.3 Pre-selection of fundamental variables

Except MonthPrice, the other seven fundamental variables (BKVLPS, CR, DVYDF, MKBK, PEM, EG and ES) are the objectives on which our research will focus. Before we go further, it is necessary to confirm that there is no linear correlation between any two of them. Otherwise, we can choose one as the representative from the two fundamental variables.

A correlation coefficient is a number between -1 and 1 which measures the degree to which two variables are linearly related. If there is perfect linear relationship with positive slope between the two variables, we have a correlation coefficient of 1; if there is positive correlation, whenever one variable has a high (low) value, the other does also. If there is a perfect linear relationship with negative slope between the two variables, we have a correlation coefficient of -1; if there is negative correlation, whenever one variable has a high (low) value, the other has a low (high) value. A correlation coefficient of zero means that there is no linear relationship between the variables. The following table shows the values of the correlation coefficient between any two of fundamental variables from years 1993 to 1998:

Table 3: Correlation coefficients of the fundamental variables (1993-1998)

	BKVLPS	CR	DVYDF	MKBK	PEM	EG	ES
BKVLPS	1	-0.285	0.395	-0.045	-0.039	-0.044	0.123
CR	-0.285	1	-0.284	0.035	0.002	0.014	-0.195
DVYDF	0.395	-0.284	1	-0.023	-0.027	-0.034	0.168
MKBK	-0.045	0.035	-0.023	1	0.038	0.009	0.004
PEM	-0.039	0.002	-0.027	0.038	1	0.089	-0.061
EG	-0.044	0.014	-0.034	0.009	0.089	1	0.078
ES	0.123	-0.195	0.168	0.004	-0.061	0.078	1

From the above the table, we can see most values of correlation coefficient are very small. The largest value (between BKVLPS and DVYDF) is 0.395 which is still not enough to estimate the linear correlation. The conclusion is that we can assume that no linear correlation exists between any two of our fundamental variables in the data set.

3.2 Data sample visualization and variable range determination

3.2.1 Fundamental Variable Pairs

Since some information of the data samples cannot be shown if we study a single fundamental variable only, a better way to find the relationship of data samples is using those variables as *variable pairs*. From previous discussion we know that variable MonthPrice is used to calculate *Return* and *Return Class* and cannot be used for variable pairs. Therefore, the other seven fundamental variables could be randomly chosen as pairs. The number of pairs, clearly, is twenty one (C_7^2). We associate each pair with a unique *pair key* (a number to identify which variables are included in this pair).

3.2.2 Visualization of data samples

After we obtain fundamental variable pairs, we would like to visualize them. Recall that each pair has exactly two fundamental variables; we will let one of them be the x axis and the other the y axis. Therefore, every data sample has a value on the x axis and a value on the y axis which form a point on the two-dimensional visualization figure. Note, if a data sample does not have either of these two values in a certain pair, this sample is not considered in the visualization figure or in later research. Next, we will show how to visualize a point based on values of a fundamental variable pair.

For example, we choose fundamental variables CR and MKBK as a pair. For company “3M CO,” the value of CR is 1.922 and the value of MKBK is 3.364 in year 1994 as shown in Table 2. We treat CR as the x axis and MKBK as the y axis. The two values can form a point on our two-dimensional visualization figure as follows:

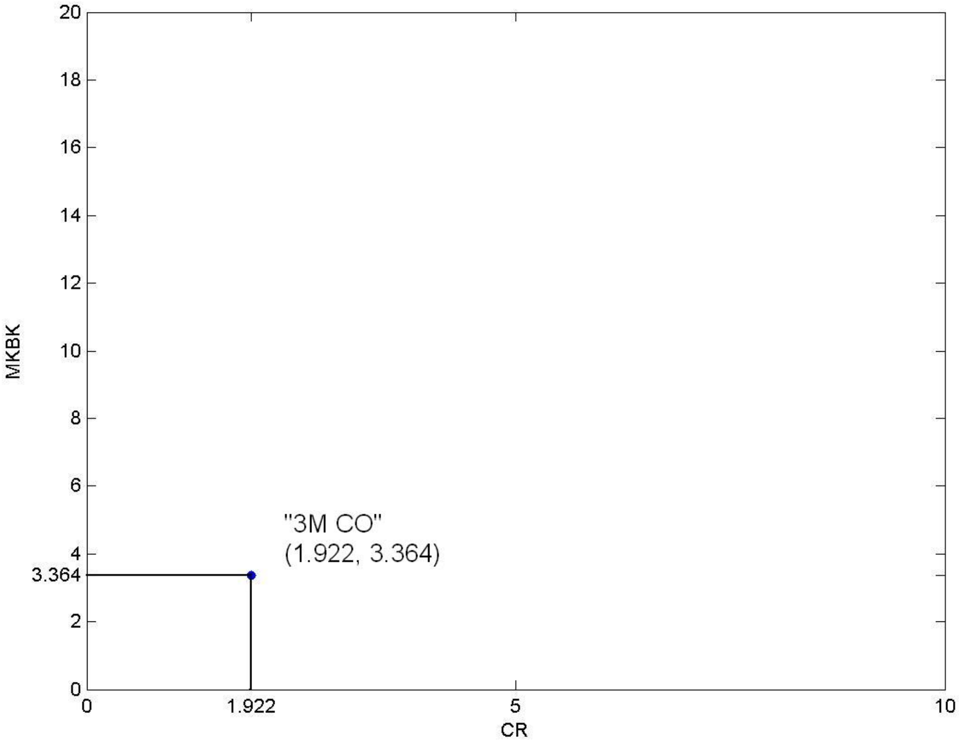


Figure 1: A data point example of the fundamental variable pair CR and MKBK.

For company “ACE LT,” the value of CR is not available in the year 1994. So we do not consider this sample for the any variable pair including CR.

Note that it does not matter which variable in the variable pairs we should choose to appear on the x axis and which one should be on the y axis because it will not affect the distribution of the data points.

3.2.3 Setting ranges of variables in each variable pair

After visualization of all the data samples from our data sets, the points are distributed in our two-dimensional visualization figures. Some points distributed far away may cause a large area in which the density of the points is very small (large area has only small number of points), we neglect those points because they are not typical points in the figure. In our study, we pay more attention to the area in which most of the points appear. So this area with high point density is our objective to study and we would like to set the range of it.

For instance, we take CR and MKBK as fundamental variable pairs and visualize the values of all the data samples (1993-2003). The figure is shown as follows:

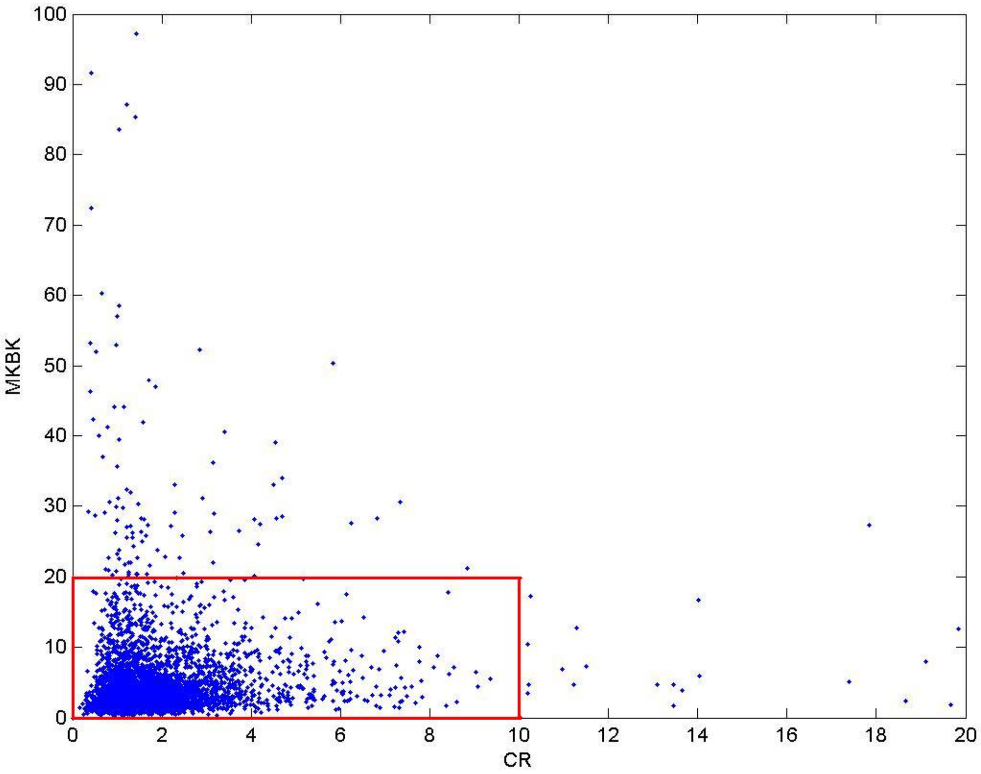


Figure 2: Visualization of data samples for the variable pair CR and MKBK

Every point in the above figure stands for one company in the data sets. It is clear that most points are distributed in the rectangle. So we set the range of CR in this variable pair as [0, 10] and the range of MKBK is [0, 20]. Note that the ranges of variables are not fixed. They are subjective to different researchers. The number of the points that are out of this range is small compared to those in this range, so we will not consider them in our later approaches.

Similarly, we can determine the range of other variable pairs. Since there are 21 variable pairs, we would like to associate each of them with a unique integer variable pair key for convenience in our study. The following table gives the range for each variable pair and their corresponding variable pair key.

Table 4: The definition of variable pair keys and their ranges

Variable Pair Key	Variable Pair and Range
1	BKVLPS, [0, 50]; CR, [0, 10]
2	BKVLPS, [0, 50]; DVYDF, [0, 10]
3	CR, [0, 10]; DVYDF, [0, 10];
4	BKVLPS, [0, 50]; MKBK, [0, 20]
5	CR, [0, 10]; MKBK, [0, 20]
6	DVYDF, [0, 10]; MKBK, [0, 20]
7	BKVLPS, [0, 50]; PEM, [0, 100]
8	CR, [0, 10]; PEM, [0, 100]
9	DVYDF, [0, 10]; PEM, [0, 100]
10	MKBK, [0, 10]; PEM, [0, 100]
11	BKVLPS, [0, 50]; EG[-300, 400]
12	CR, [0, 10]; EG[-300, 400]
13	DVYDF, [0, 10]; EG[-300, 400]
14	MKBK, [0, 10]; EG[-300, 400]
15	PEM, [0, 100]; EG[-300, 400]
16	BKVLPS, [0, 50]; ES[0, 5]
17	CR, [0, 10]; ES[0, 5]
18	DVYDF, [0, 10]; ES[0, 5]
19	MKBK, [0, 10]; ES[0, 5]
20	PEM, [0, 100]; ES[0, 5]
21	EG[-300, 400]; ES[0, 5]

3.3 Data point coloring according to density

Since the points shown in the range of each area are with the same color, it is very difficult for humans to distinguish density of the points accurately. To solve this problem, we divide each area into sub-areas and classify them into a different density level based on number of points in them. Then according to the different density levels of the sub-areas, we give different colors to the points in these sub-areas.

3.3.1 Sub-area generation

According to the range of each fundamental variable pair, we can find a corresponding area. We divide each area average into 100 sub-areas and associate each sub-area with a unique integer number from 1 to 100 and we call each number *sub-area ID*. Take the fundamental variable pair 11 (CR and MKBK) as an example. The sub-areas of this pair are shown in Figure 3. The number in each sub-area is the corresponding *sub-area ID*. As we can see, the *ID* of leftmost downward sub-area is 1 and the *ID* number increases in the same row from left to right (also from downward to upward). Note that each sub-area has its ranges, for example, sub-area 2 is within [1, 2] for variable CR and is within [0, 2] for variable MKBK.

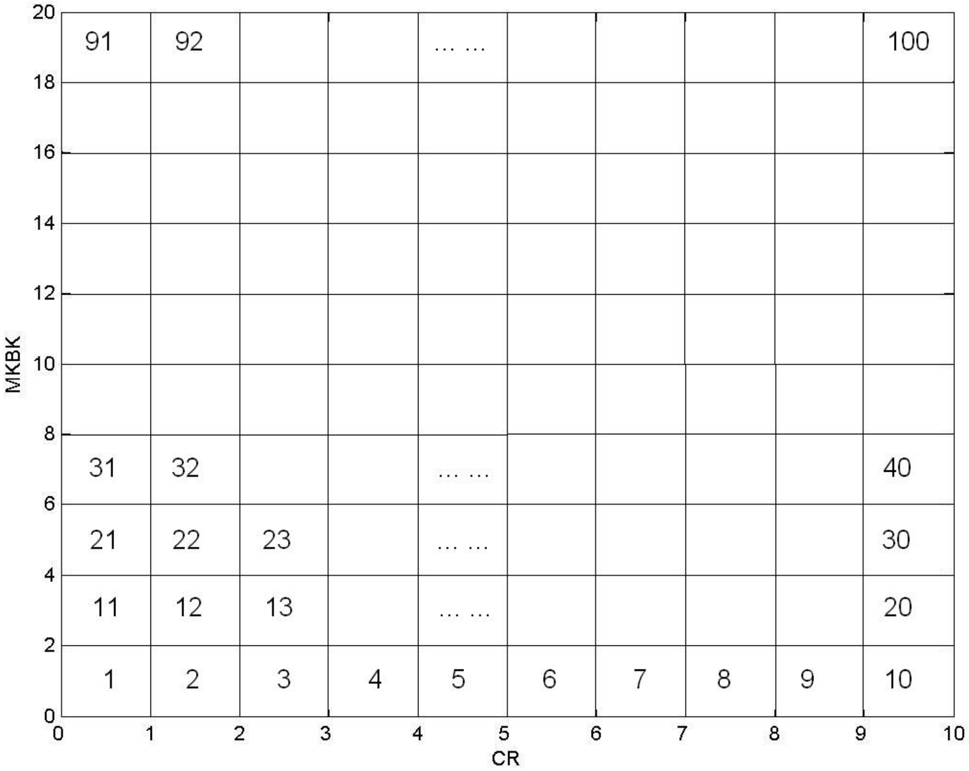


Figure 3: The sub-areas of variable pair CR and MKBK

3.3.2 Density levels calculation

The density levels for different fundamental variable pairs are obviously different. The following are the steps used to get the density levels for the particular fundamental variable pair based on a certain data set.

Step 1. From the data set, we choose the *Return Class* and two fundamental variables in the particular pair to form a useful data set S. The other columns of the data set which will not be used are ignored at this time.

Step 2. We divide this useful data set S into five subsets S_0, S_1, S_2, S_3 and S_4 according to the values of *Return Class*. In subsets S_0, the values of *Return Class* of all the data samples are 0. Similarly, the Return Class values in subsets S_1, S_2, S_3 and S_4 are 1, 2, 3 and 4 respectively.

Step 3. In subset S_0, we visualize each data sample whose values are within the variable range and count the number of data points in every sub-area. Assume M_0 is the maximum number of data points per sub-area.

Step 4. For subsets S_1, S_2, S_3 and S_4, we repeat the work in Step 3 and get the maximum number of data points per sub-area, namely M_1, M_2, M_3 and M_4.

Step 5. To unify the density level of all kinds of *Return Class*, assume M is the maximum number of data points per sub-area for the whole data set S. The formula to calculate M is $M = \max\{M_0, M_1, M_2, M_3, M_4\}$. In this study, there are five density levels where each of them is with length $len = M/5$.

Step 6. Suppose the number of points in a sub-area *n* ($n \in [1,...,100]$) is *P(n)*, the density level of one sub-area *n* ($n \in [1,...,100]$) is *L(n)*, we have: if $0 \le P(n) < len$, *L(n)* = 1; if $len \le P(n) < 2len$, *L(n)* = 2; if $2len \le P(n) < 3len$, *L(n)* = 3; if $3len \le P(n) < 4len$, *L(n)* = 4 and if $P(n) \ge 4len$, *L(n)* = 5.

3.3.3 Setting different color for each density level

We assign different colors to the data points in sub-area with different density levels. In this study, there are five levels associated with five colors which are Red, Yellow, Blue, Cyan and Green. When the density level is equal to 5, the color of the points should be red. Similarly, the color of points is yellow for level 4, blue for level 3, cyan for level 2 and green for level 1.

3.4 An example of visualization

In this section, there is an example to show how to visualize the data samples and color the data points.

Let's take the data set from the year 1994 (shown in Table 2) as an example. At this time, we would like to pay attention to the variable pair CR and MKBK whose variable pair key is 5. Looking at Table 4, we get the range of CR, which is [0, 10], and the range of MKBK, which is [0, 20]. In this example we also choose CR as the x-axis and MKBK as the y-axis.

First, we choose the useful information from the whole data set to build the useful data set S. The following table shows the first several lines of set S:

Table 5: Part of small data set S for variables CR and MKBK (1994)

Row Num	1995 R-C	1994 CR	1994 MKBK
1	2	1.922	3.297
2	2	1.115	2.268
3	3	@NA	1.75
4	3	2.96	0
…	…	…	…

According to the *Return Class* values, every data sample could be put in its corresponding subset. For instance, in Table 5, the first and the second samples will be put in S_2 and the fourth one should be in S_3 (we do not consider the third sample because its value of CR is not available).

Next, we use MATLAB to plot the data samples from different subsets in different figures as following:

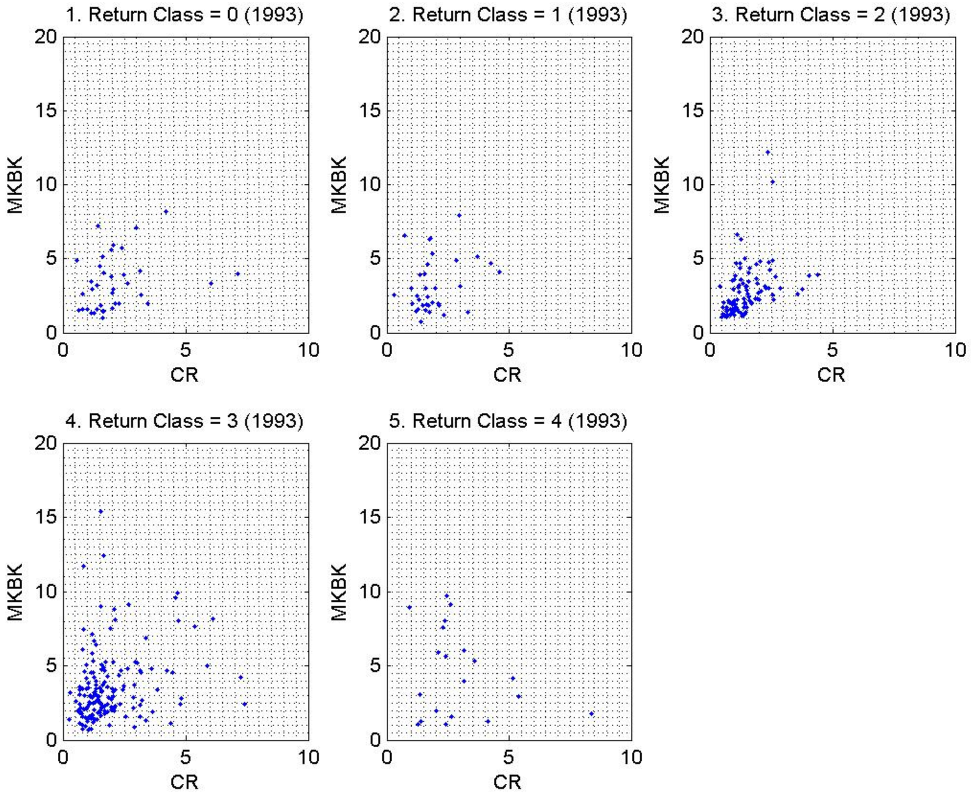


Figure 4: Non-colorful visualization for the variable pair CR and MKBK (1994)

Each subfigure in Figure 4 shows the data samples with particular value of *Return Class*. For example, the first sample of Table 5 will be displayed as a data point in sub-area 12 in the third subfigure in Figure 4 (As shown in Figure 3, sub-area 12 contains the data samples whose CR in [1, 2] and MKBK in [4, 6]). Also, it is easy to count the number of points of each sub-area using MATLAB. Another advantage of using MATLAB is that it avoids the problem of miscounting if two points are overlapping. The maximum number of points per sub-area from all the 500 sub-areas (each subfigure contains 100 sub-areas), M , is 36. This number is calculated using MATLAB.

According to the maximum number of points per sub-area M , which is equal to 36, the length of each density interval is 36/5=7.2. The following table is an example of the density levels' definition, showing relationship of different density levels, the range of the number of data points in the sub-area and the color of the data points in that area.

Table 6: An example of density levels definition (M = 36)

Density Level	The Range of Point Numbers	The Color of the Points
5	(28.8, 36]	Red
4	(21.6, 28.8]	Yellow
3	(14.4, 21.6]	Blue
2	(7.2, 14.4]	Cyan
1	[0, 7.2]	Green

Based on Table 6, we can redraw Figure 4 into its colorful version, Figure 5 which is shown below as:

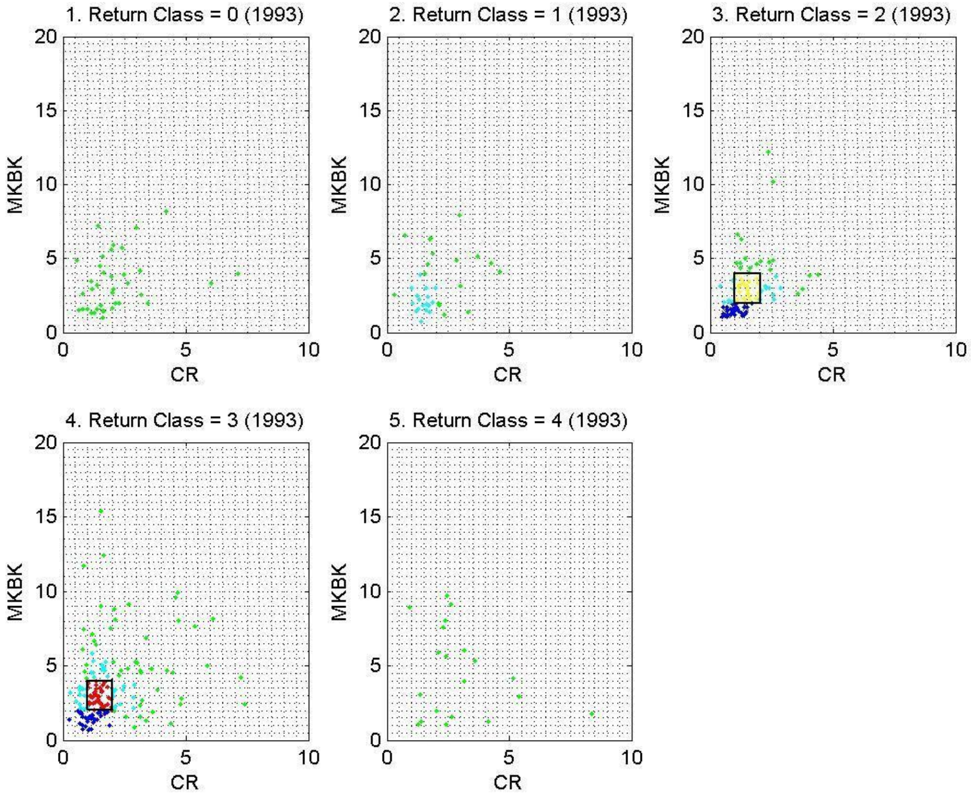


Figure 5: Colorful visualization of data set for CR and MKBK (1994)

In Figure 5, we can get more information about the distribution of data points. Take the sub-area 12 (as shown in black rectangular in Figure 5) an example; it is very clear that this area has more data points when the Return Class is 3 than when the Return Class is 2 because the red points denote higher density level than yellow points. However, it is difficult for us to see that in Figure 4.

This example shows the visualization of one variable pair (CR and MKBK) for the data set 1994. Similarly, we could visualize the other twenty variable pairs. That is, we could generate 21 figures for every data set if there are enough variable values. Totally, from the 6 data sets (1993-1998) we generated more than one hundred figures like Figure 5.

CHAPTER 4

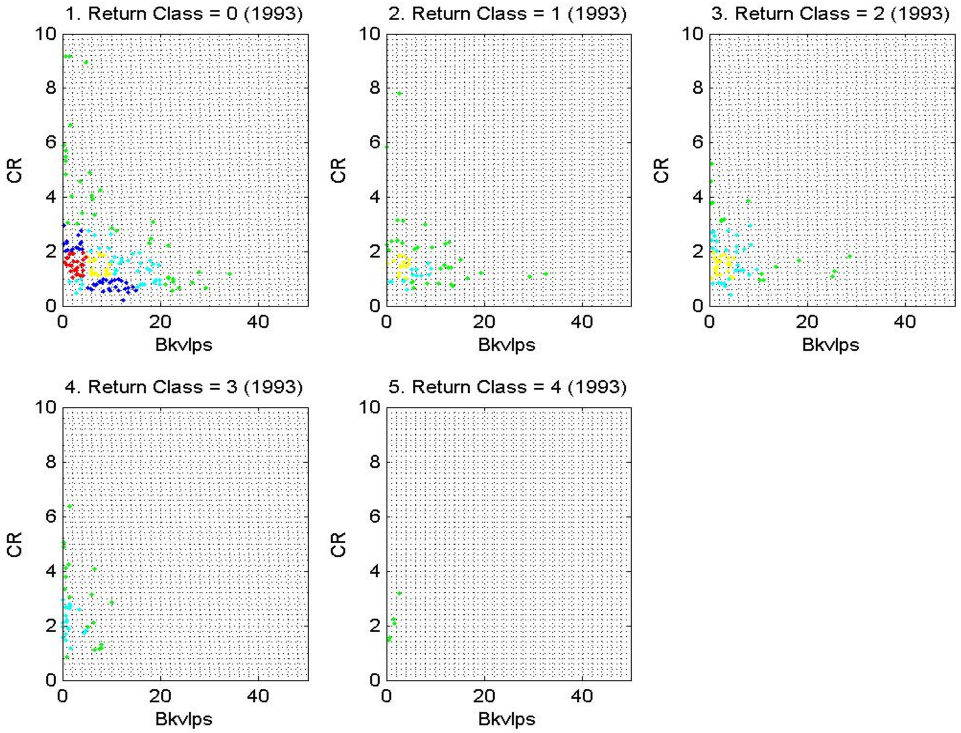
RULE EXTRACTION AND VERIFICATION

From the discussion in Chapter 3, we get figures which represent the visualization results of the fundamental data. These figures are a more informational and direct form for us to analyze than large amounts of data values. However, since there are a large number of the figures which we have generated, if we want to find some rules, one efficient way is to use the statistic methods.

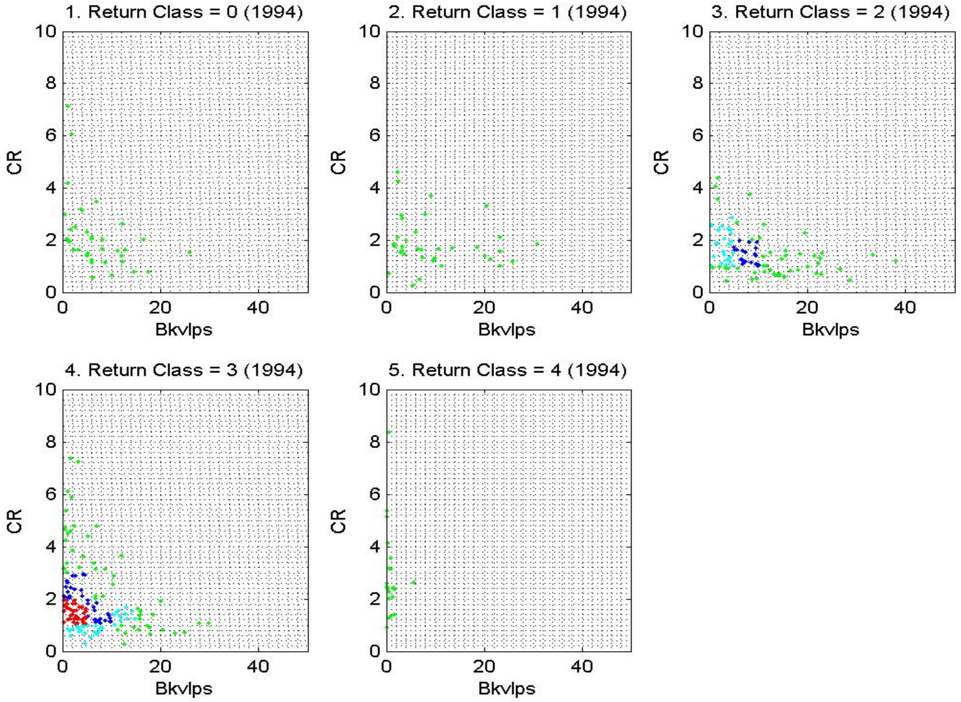
In section 4.1, we separate variable pairs to analyze correlations between fundamental variables. For each variable pair, we have a corresponding visualization result which has all the visualized figures for it from the years 1993 to 1998. In section 4.2, we propose a way to calculate the sub-area frequencies of each variable pair. First we find the density levels of each sub-area corresponding to different *Return Class* in a particular year; then for each *Return Class* and each high density level, we allow *Sub-area Set* to include all the sub-areas corresponding to them; *statistic set* can be obtained by taking union of all experience years Sub-area Sets with the same *Return Class* and the same density level. From each statistic set, we can calculate the *frequency* of each sub-area. At last, statistic results for a specific variable pair can be derived from each statistical set to provide us information about the potential rules. In section 4.3, we show how to derive rules from observing the visualization results and we show the performance of these rules derived in section 4.4.

4.1 Visualization result of each variable pair

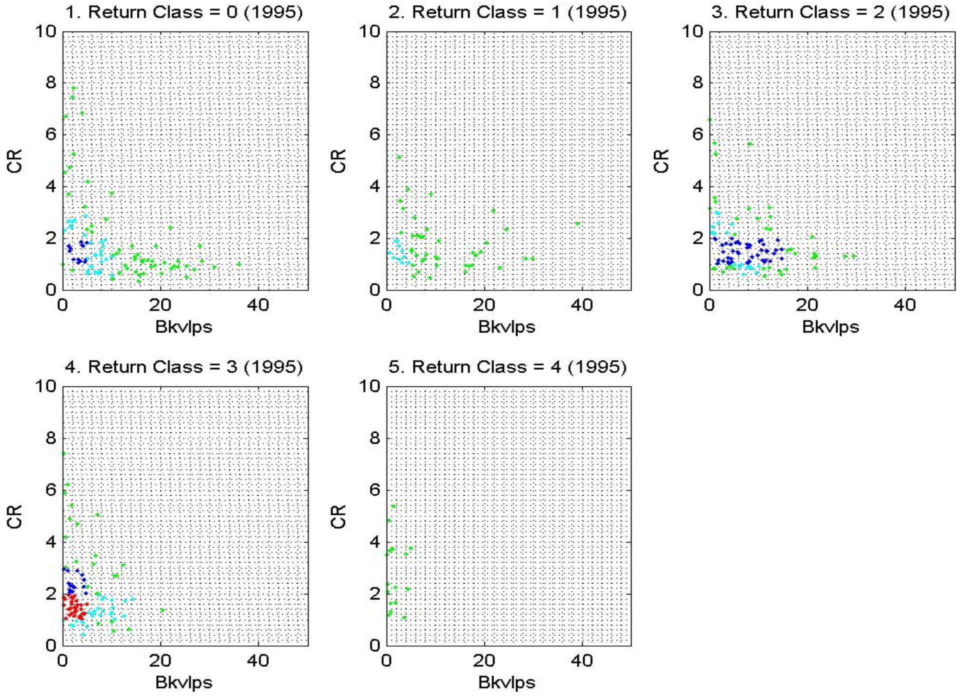
In Chapter 3, we get figures which show the distribution of the fundamental values according to the Return Class of each company. To find the correlation between the fundamental variables, we separate these figures by variable pairs (each variable pair has its own six corresponding visualization figures from year 1993 to year 1998). The following is an example for variable pair 1 (BKVLPS and CR). In Figure 6, six subfigures are shown and each of them stands for one particular year. For example, subfigure (a) is the visualization result of variable pair 1 (BKVLPS and CR) in 1993. Similarly, subfigures (b) to (f) show the visualization figures from year 1994 to year 1998 respectively.



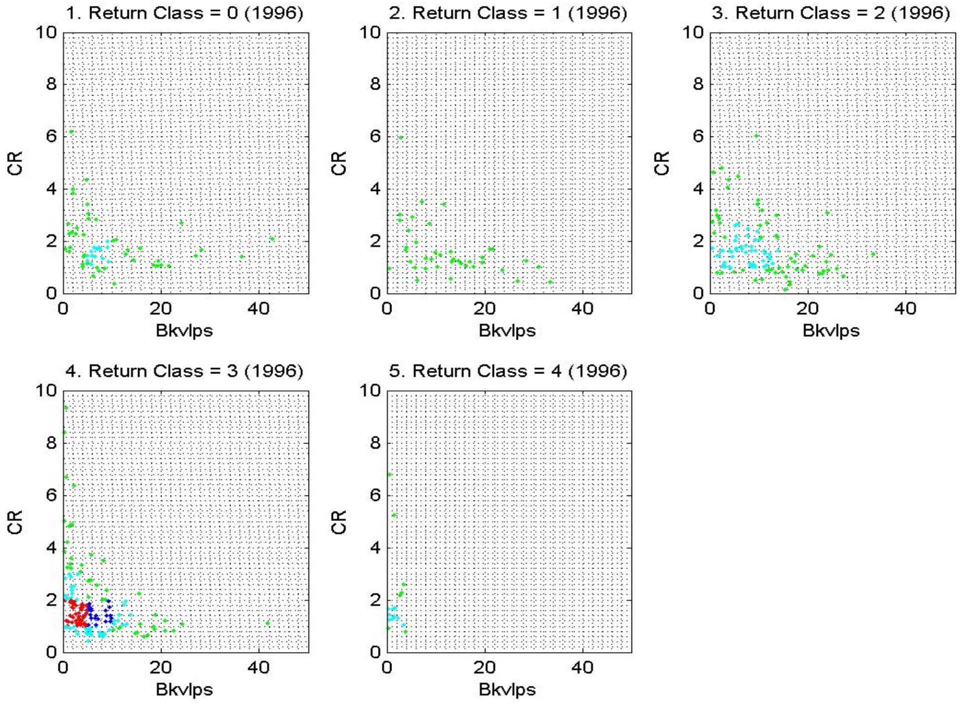
(a) Visualization figure of variable pair 1 (BKVLPS and CR) in 1993



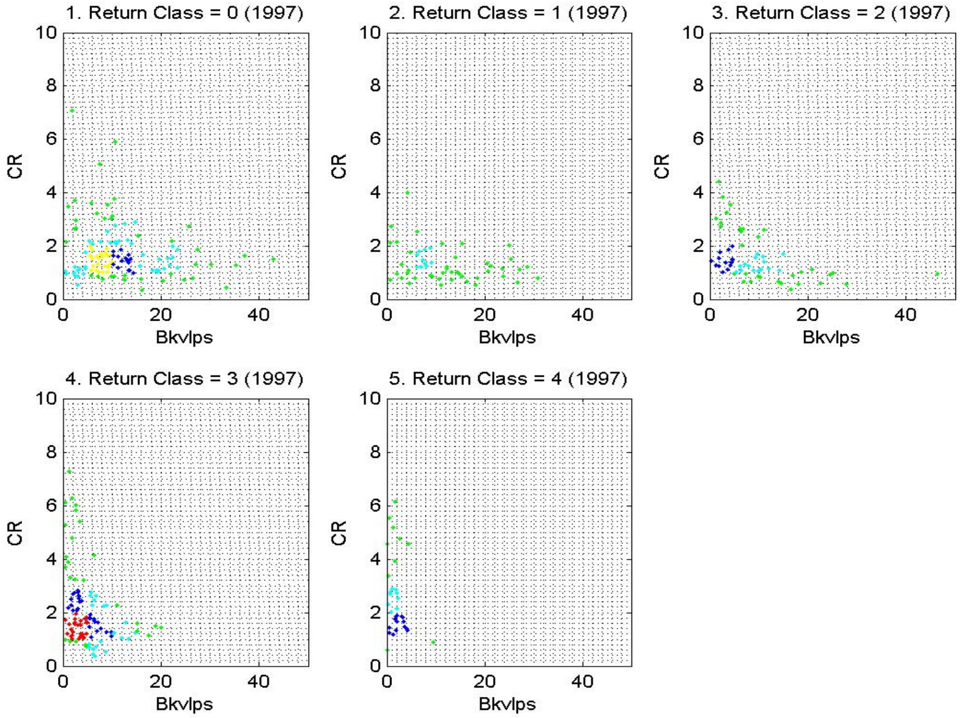
(b) Visualization figure of variable pair 1 (BKVLPS and CR) in 1994



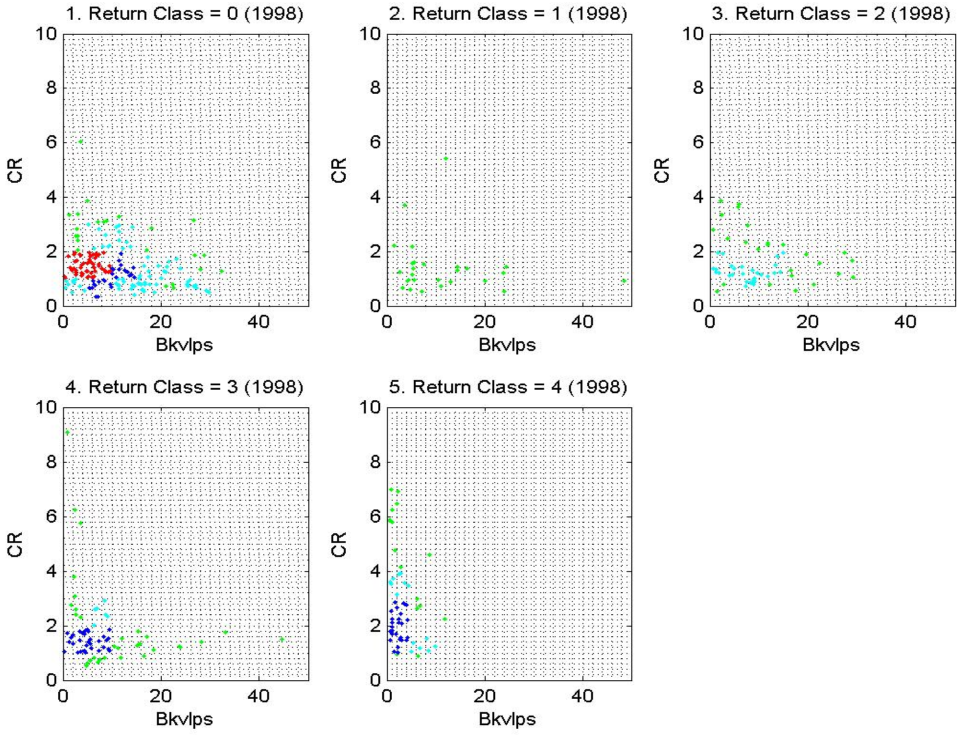
(c) Visualization figure of variable pair 1 (BKVLPS and CR) in 1995



(d) Visualization figure of variable pair 1 (BKVLPS and CR) in 1996



(e) Visualization figure of variable pair 1 (BKVLPS and CR) in 1997



(f) Visualization figure of variable pair 1 (BKVLPS and CR) in 1998

Figure 6: Visualization result of variable pair 1 (BKVLPS and CR)

For each variable pair, we have a visualization result similar to Figure 6. Some results of the other variable pairs are shown in Appendix A.

4.2 Calculating the sub-area frequencies of each variable pair

For each variable pair, there are six visualization figures and each of them has 100 sub-areas. First, we want to find the sub-area which can provide us more information. One efficient way to do this is to calculate the sub-area frequencies of each variable pair. The sub-areas which have high frequency are the ones we want to pay more attention to because they denote the areas in which these two fundamental variables have more correlations (not linear).

From Chapter 3 we can see that there are 100 sub-areas in our visualization figure of a certain variable pair. These sub-areas have their values of density levels for each *Return Class* in a certain year. Take variable pair BKVLPS and CR as an example, the density levels of each sub-area corresponding to different *Return Class* in the year 1993 is shown in the table as follows:

Table 7: Density levels of each sub-area (1993)

Sub-area ID	Return Class = 0	Return Class = 1	Return Class = 2	Return Class = 3	Return Class = 4
1	2	2	2	1	1
2	3	1	1	1	1
3	3	1	1	1	1
…	…	…	…	…	…
100	1	1	1	1	1

In the above table, column one represents the ID of each sub-area (from 1 to 100). Column two shows the density levels of the sub-areas in which the *Return Class* is of data points all equal to 0. Similarly, column three, four, five and six show the density level of sub-areas in which the *Return Class* of data points are equal to 1 , 2 , 3 and 4 respectively. Each row of Table 7 represents the density levels of one particular sub-area according to different *Return Class*. For example, for sub-area 1, the density level is 1 when the value of *Return Class* is 3 or 4, and its density level is 2 when the value of *Return Class* is 0, 1 or 2.

After we have the density levels of each sub-area according to a different *Return Class,* we want to know the information which the sub-area corresponds to high density levels (3, 4 and 5) for each *Return Class*. To do so, for each

Return Class and each high density level, we let *Sub-area Set* to include all the sub-areas which correspond to them. There are a total of fifteen *Sub-area Sets* for each variable pair (five values of Return Class multiplied by three density levels). For instance, the following table shows the *Sub-area Sets* for each *Return Class* and each high density level of the variable pair BKVLPS and CR in the year 1993.

Table 8: Sub-area sets of the variable pair BKVLPS and CR (1993)

Return Class	Density Level	Sub-area Set
4	5	Φ
4	4	Φ
4	3	Φ
3	5	Φ
3	4	Φ
3	3	Φ
2	5	Φ
2	4	{11}
2	3	Φ
1	5	Φ
1	4	{11}
1	3	Φ
0	5	{11}
0	4	{12}
0	3	{2, 3, 21}

In Table 8, the first column shows all *Return Class* from 0 to 4. The second column represents the high density levels from 3 to 5 (we do not consider low density levels here because they are not of our interest). The third column shows the *Sub-area Set* which corresponds to a certain *Return Class* under a certain density level where Φ means an empty-set (no sub-areas included). Take the ninth row as an example; sub-area 11 has a density level 4 when the

Return Class is 2.

Similar to Table 8 which shows sub-area sets for each *Return Class* and each high density level of the variable pair BKVLPS and CR in the year 1993, we can have tables for each variable pair for all the data sets. For each variable pair, we want to find how many times (the *frequency*) each sub-area corresponds to a specific *Return Class* and specific density level of all the data sets. To do so, we unify the sub-area sets with the same *Return Class* and the same density level. We call this new union set a *statistic set*. Note that we just take the union of sub-area sets, so each *statistic set* should have 15 rows similar to Table 8.

For each *statistic set*, we can calculate the *frequency* $f(ID)$ of each sub-area ID in each set (how many times that sub-area ID appears in that set). Also, we can get the number of different sub-area ID Num_ID (how many different sub-area IDs a particular statistic set has) and the total number of *sub-areas* Num_Total (total number of IDs the *statistic set* has). For example, if *statistic set* is {2, 2, 12} for a particular *Return Class* and density level, $f(2) = 2$ and $f(12) = 1$ (sub-area 2 appears twice and sub-area 12 appear once in this statistic set); $Num\_ID = 2$ (this statistic set has two different IDs) and $Num\_Total = 3$(The number of all IDs are 3).

The following table is an example of the statistic result of the variable pair BKVLPS and CR (1993-1998):

Table 9: Statistic result of variable pair BKVLPS and CR

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	5	0	0										
1	4	4	0	0										
1	4	3	2	3	11	2	21	1						
1	3	5	1	4	11	4								
1	3	4	0	0										
1	3	3	3	8	21	3	12	4	11	1				
1	2	5	0	0										
1	2	4	1	1	11	1								
1	2	3	3	5	11	2	12	2	13	1				
1	1	5	0	0										
1	1	4	1	1	11	1								
1	1	3	0	0										
1	0	5	2	3	11	2	12	1						
1	0	4	1	2	12	2								
1	0	3	5	7	21	1	2	2	3	1	11	1	13	2

In the table 9, we use data sets from 1993 to 1998. Every row shows a statistic result for a particular *Return Class* and density level. The first column is the variable pair key which denotes a unique variable pair corresponding to it. In this example, the variable pair key is 1 which refers to the variable pair BKVLPS and CR. The second column shows the *Return Class* from 0 to 4 and the third column shows the high density level from 3 to 5. The fourth column represents Num_ID. The fifth column denotes Num_Total. The rest of the columns represent the sub-area IDs and their corresponding *frequencies.* The even columns (6th, 8th, … , etc.) show the *sub-area IDs*. The odd columns (7th, 9th, … , etc.) are the *frequencies* of the sub-areas in the left column. Take the fourth row (1, 4, 3, 2, 3, 11, 2, 21, 1) for example; 1 means the variable pair is BKVLPS and CR; 4 means the Return Class is 4, and 3 means the density level is 3; 2 means there are two kinds of sub-areas; the second 3 means the total number of sub-areas is 3; 11 means one of the sub-areas is 11 and the second 2 means its frequency is 2; 21

means the other sub-area is 21 and the second 1 means its frequency is 1.

4.3 Observation results from visualization figures

In last section, we got the sub-areas with high frequencies in our statistic results. Compared to sub-areas with low frequencies, these sub-areas are more likely to provide more general information. Considering these high frequency sub-areas, we review the visualization result for each variable pair and attempt to find hidden rules.

Initially, we want to find some variable pairs who have different sub-areas for good and bad return stocks. For example, as Figure 7 shows, lots of stocks in which Return Classes are equal to 0 distribute in sub-area 12, while many high return stocks cluster in sub-area 11.

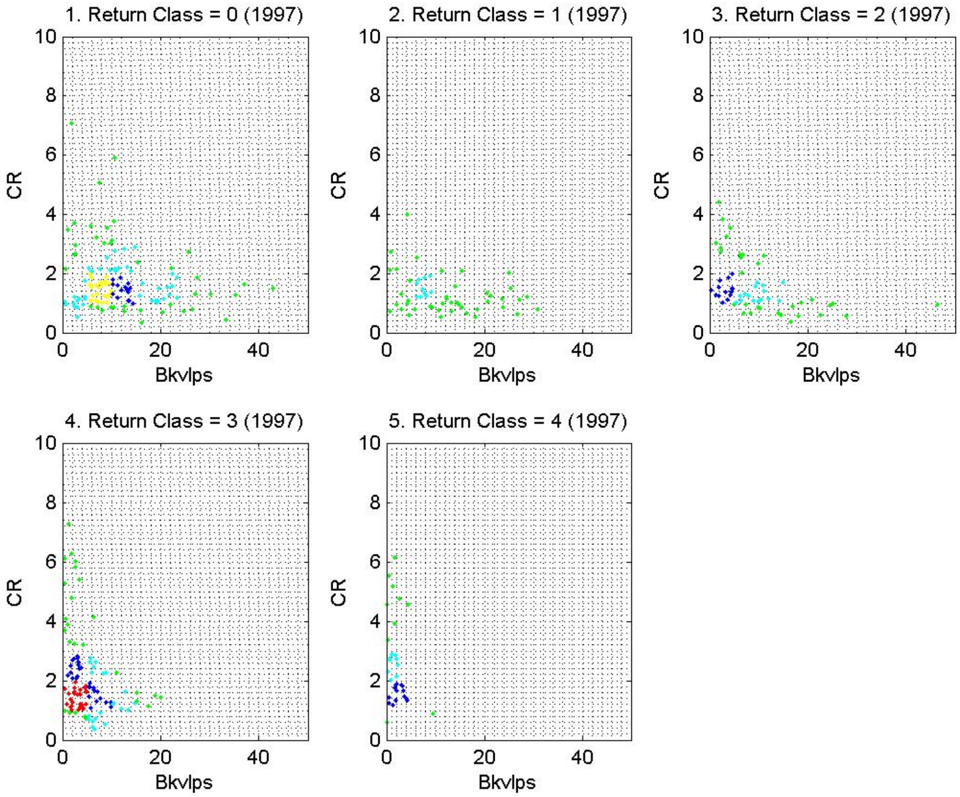


Figure 7: Colorful visualization of data set for BKVLPS and CR (1997)

From this observation, we would like to generate two rules to predict the stock performance. One rule to get more profit is “If BKVLPS is within [0, 5] and CR is within [1, 2], then the return is good.” Another rule to avoid risk is “If BKVLPS is within [5, 10] and CR is within [1, 2], then the return is bad.” However, this phenomenon does not occur in all of the other years for the same variable pair BKVLPS and CR (See Figure 6). There are similar results for the other variable pairs. The results from this kind of observation can be seen as an “accident” but cannot be developed as a general rule for stock prediction.

Although we can not find a sub-area in which the data samples always have a good or bad return in any variable pair, we notice that time is a very important factor that could affect the return of stocks. Again, take the variable pair BKVLPS and CR as an example; most of the data points are plotted in the first subfigure (Return Class = 0) in year 1993, while many data points are plotted in the fourth subfigure (Return Class = 3) in year 1994. Most stocks had a high return because of the great improvement in the economy in 1995. This situation inspires us to compare the individual stock's performance of each sub-area with the performance of all the stocks. We calculate the average return of all stocks and the average return of stocks in each sub-area. Then we put the result in the form of a table which is convenient for us to compare. The following table is an example for variable pair 1 (BKVLPS and CR):

Table 10: Table of average return values for the variable pair BKVLPS and CR

	1993	1994	1995	1996	1997	1998	Sum
All Stocks 1	0.008	0.428	0.250	0.369	0.254	0.350	
All Stocks 2	434	442	451	464	469	475	
1_1	0.048	0.650	0.289	0.648	0.430	0.116	
1_2	20	21	16	16	17	18	
1_3	1	1	1	1	1	-1	5
2_1	-0.105	0.326	0.081	0.385	0.291	0.100	
2_2	20	19	23	21	22	26	
2_3	-1	-1	-1	1	1	-1	2
…	…	…	…	…	…	…	…
11_1	0.084	0.487	0.363	0.482	0.596	0.924	
11_2	77	72	76	75	67	66	
11_3	1	1	1	1	1	1	6
…	…	…	…	…	…	…	…

In the above table, the second row shows the average return of all the stocks from the year 1993 to the year 1998. The third row shows the number of all available samples for each year. The following rows contain information about the sub-areas. There are three rows for each sub-area: the first row shows the average return of the stocks in that sub-area; the second row shows the number of stocks in the sub-area; the third row shows whether the return of the stocks in the sub-area is higher than or equal to the average of all stocks (1 means yes, -1 means no, 0 means no data sample in that sub-area). The last column shows the total number of ones in the third row of each sub-area. That is the number of years in which the average return of the corresponding sub-area is not lower than the average return of all the stocks. Let’s take sub-area 1 of the variable pair BKVLPS and CR as an example; in the year 1993, there are 20 stocks in this sub-area and their average return is 0.048 which is higher than the average return of all the stocks (0.008). And sub-area 1 did well in 5 years (from 1993 to 1997) except for the year 1998 (the average return is 0.116).

Table 10 is for the variable pair BKVLPS and CR. We can get similar tables for the other twenty variable pairs. After studying all of these tables, we notice that some sub-areas work well in all 6 years. Based on these special sub-areas, we can derive some rules in “IF-THEN” form. For example, from Table 9 we can see that sub-area 11 did better than the average return of all the stocks for each year, its corresponding rule is as follows:

1. If BKVLPS is within [0, 5] and CR is within [1, 2], then the return is better than the average.

Here, “the return is better than the average” means the average return of the stocks within both of the ranges in IF condition is higher than or equal to the average return of all the stocks in most years. For other variable pairs, we may find their sub-areas like sub-area 11 in variable pair 1 (BKVLPS and CR) and we derive similar “If-Then” rules from them.

In this study, we generate a total of four rules which are as follows:

1. If BKVLPS is within [0, 5] and CR is within [1, 2], then the return is better than the average.

2. If BKVLPS is within [0, 5] and DVYDF is within [0, 1], then the return is better than the average.

3. If BKVLPS is within [0, 5] and MKBK is within [4, 6], then the return is better than the average.

4. If DVYDF is within [0, 1] and MKBK is within [4, 6], then the return is better than the average.

4.4 Rules performance

In the previous section, we derived rules from data sets between 1993 and 1998. In this section, we show the performance of these rules using data sets from 1999 to 2003. Note that here fundamental variables are from data sets between 1999 and 2003 and the return values are from 2000 to 2004.

According to certain rules, we chose stocks within the ranges of the corresponding fundamental variables; then we can see the average of *Return* values for these stocks in the next year. The following table shows the performance of rule 1:

Table 11: The performance of rule 1 in data sets from 1999 to 2003

Year	Number of Selected Stocks	Average Return of Next year	Average Return of all stocks in S&P 500 of Next year
1999	56	0.246	0.192
2000	36	0.048	0.008
2001	46	-0.136	-0.169
2002	41	0.610	0.536
2003	36	0.166	0.162

From Table 11 we can see that rule 1 has good performance, the average of the returns of rule 1 is always better than the average of the returns of all stocks in S&P 500. The companies which selected base on rule 1 in the year 1999 are listed in Appendix B.

Similarly, we can show the performance of the other rules. The performances of these other rules are shown in the table 12:

Table 12: The performance of all rules

	1999	2000	2001	2002	2003
All Stocks	0.192 (482)	0.008 (490)	-0.169 (495)	0.536 (497)	0.162 (498)
Rule 1	0.246 (56)	0.048 (36)	-0.136 (46)	0.610 (41)	0.166 (36)
Rule 2	0.287 (100)	0.039 (77)	-0.145 (67)	0.673 (61)	0.178 (59)
Rule 3	0.213 (16)	0.153 (11)	-0.052 (17)	0.671 (16)	0.179 (19)
Rule 4	0.724 (25)	0.217 (32)	-0.218 (43)	0.638 (87)	0.281 (53)

In the above table, the second row shows the average of the returns of all the stocks in S&P 500. The third through sixth rows show the average of the returns of the stocks selected by each rule. The numbers of the corresponding stocks are given in the brackets. From the above table, the average of the returns of most of the rules is better than the average of the returns of all stocks in the S&P 500 except the rule 4 in year 2001. We can say that most of the rules perform well. Furthermore, the performance of some of the rules (rule 2 in the year 1999 and the year 2002) is sometimes very good because the average returns from those rules are almost 50% larger than the average of returns of all the stocks in S&P 500.

.

CHAPTER 5

CONCLUSIONS AND RECOMMENDED FUTURE STUDY

5.1 Overall Summary

The main goal of this research is to derive a set of rules which can be used to select stocks with better than average returns. From these rules, we can see the correlations (not linear) between fundamental variables and how they are related with the return of the stocks. In other words, we can find the ranges of these variables within which the corresponding stocks have better than average returns. To derive these kinds of rules, first we visualize the securities data. Then based on the visualization figures from the data, we use statistical methods to find the sub-areas with high frequency. Those sub-areas are where we pay the most attention to derive our rules. After observation of these sub-areas, we find that we cannot find a sub-area in which the data samples always have a good or bad return in any variable pair. However, we can find sub-areas in which the return of the stock is better than the average return of all the stocks. Then we can derive rules based on these observations with which the stock has better returns.

Visualization of the data is implemented in three phases: data collection and preparation; data sample visualization and variable range determination; and data point density coloring. Lastly, an example of visualization is provided. In data collection and preparation, we use the fundamental data to construct data sets for every year (from 1993 to 1998). Every data set includes the name of companies and their corresponding values of *Return*, *Return Class* and other fundamental variables in a specific year. In the next phase, we choose any two fundamental variables to be variable pairs and build their visualization figures. Each data sample in the data sets can be transformed to a data point in our two-dimensional visualization figure. To neglect those points which are far away from the others, we set ranges for each variable pair. The next phase is data point coloring according density; we define sub-areas for each variable pair and set their density levels. According to the density level of each sub-area, we associate the corresponding color to the data points in it. At last, an example of data sets for the variable pair CR and MKBK in the year 1994 is used to show the details of visualization.

From the results of our visualization, we can directly see the sub-areas where the most points appear for a certain variable pair in a particular year. However, a statistical method is a better way to get the distribution of the data points. From all experience with data sets (the year before 1999), we build *statistic sets* for each variable pair corresponding to each *Return Class* and each high density level. Then we can calculate the *frequency* of each sub-area from which we can derive our rules. Note that the rules we derive are not based on statistical methods. Statistical methods only provide us with high frequency sub-areas, which we pay more attention to (as well as considering other sub-areas) to derive our rules. Rules are derived based on the observations that within the sub-areas; the stock has better performance than average returns of the stock. The data sets after the year 1998 are used to test the performance of the rules we generated. The average returns of most rules are better than the average return of all stocks in S&P 500.

5.2 Conclusions

From the result of the visualization and observations of sub-areas provided by the statistical methods, we derive four rules based on the experience data sets from year 1993 to year 1998:

1. If BKVLPS is within [0, 5] and CR is within [1, 2], then the return is better than the average.

2. If BKVLPS is within [0, 5] and DVYDF is within [0, 1], then the return is better than the average.

3. If BKVLPS is within [0, 5] and MKBK is within [4, 6], then the return is better than the average.

4. If DVYDF is within [0, 1] and MKBK is within [4, 6], then the return is better than the average.

We use data sets from years 1999 to 2003 to test the performance of the above rules. Most of these rules can pick up stocks with better returns. Furthermore, the performance of some rules (rule 2 in the year 1999 and the year 2002) are sometimes very good because the average returns from those rules are almost 50% larger than the average of returns of all stocks in S&P 500.

Our methods of data visualization and statistical analysis of the results give us a way to process continuous stock data and to drive rules to make a decision for stock selection. This method has the capability to reveal hidden patterns in the fundamental variable data of the stock market and these patterns are helpful for the selection of stock.

5.3 Recommended future study

In this study, we use seven fundamental variables which are Book Value Per Share (BKVLPS), Current Ratio (CR), Dividend Yield (DVYDF), Price to Book Ratio (MKBK), Price to Earning Ratio Monthly (PEM), % Earning Growth (EG) and Earning Stability (ES). However, there are many other fundamental variables available in the stock market which can be incorporated to derive rules as well.

This study selects the variable pairs and determines their ranges which have a relationship with the return of stocks. Based on those variable pairs and their ranges, other data mining methods such as decision tree can be used to improve the performance of securities analysis.

Finally, in this study any two fundamental variables are chosen to become pairs and our results are based on these variable pairs. Similarly, we can derive rules where multiple variables are chosen in one group. For these kinds of groups, the corresponding multi-dimensional visualization of data becomes more difficult to realize but the methods of statistic sub-areas frequencies are still useful.

REFERENCES

[1] P. Blustein, “Ben Graham's last will and testament”, *Forbes*, August 1, pp. 43-45, 1977.

[2] R. Andrews, J. Diederich and A. Tickle, “A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks,” *Knowledge-Based Systems*, 8(6), pp. 373-389, 1995.

[3] R. C. Holte, “Very simple classification rules perform well on most commonly used datasets,” *Mach. Learn*, Vol. 11, pp. 63–91, 1993.

[4] L. Breiman, J.H. Friedman, R. A. Olsen, and C. J. Stone, *Classification and Regression Tree*, Belmont, CA: Wadsworth, 1984.

[5] J. R. Quinlan, “Induction of decision trees”, *Machine Learning*, Vol. 1, pp. 81-106, 1986.

[6] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann, 1993.

[7] Y. Yuan and M. J. Shaw, “Induction of fuzzy decision trees,” *Fuzzy Sets Syst.*, vol. 69, pp. 125–139, 1995.

[8] C. Z. Janikow, “Fuzzy decision trees: Issues and methods,” *IEEE Trans. Syst., Man, Cybern. Part B*, vol. 28, no. 1, pp. 1–14, Jan. 1998.

[9] W. X. Zhao and J. Hong, “On the handling of fuzziness for continuous valued attributes in decision tree generation,” *Fuzzy Sets Syst.*, vol. 99, pp. 283–290, 1998.

[10] W. Pedrycz and A. Sosnowski, “Designing decision trees with the use of fuzzy granulation,” *IEEE Trans. Syst., Man, Cybern., Part A*, vol. 30, pp. 151–159, Mar. 2000.

[11] R. S. Michalski and K. A. Kaufman, “Data mining and knowledge discovery: A review of issues and a multistrategy approach,” *Machine Learning and Data Mining: Methods and Applications*, R. S. Michalski, I. Bratko, and M. Kubat, Eds. New York: Wiley,1997.

[12] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, “The multi-purpose incremental learning system AQ15 and its testing application to three medical domains,” *Proceedings of the National Conference on Artificial Intelligence, AAAI*, Philadelphia, pp. 1041-1045, 1986.

[13] P. Clark and T. Niblett, “The CN2 induction algorithm,” *Mach.Learn.*, vol. 3, pp. 261–283, 1988.

[14] W. W. Cohen, “Fast effective rule induction,” *Proc. 12th Int. Conf. Machine Learning*, pp. 115–123, 1995.

[15] K. Saito and R. Nakano, “Medical diagnostic expert system based on PDP model,” *Proc. IEEE Int. Conf. Neural Networks*, vol. 1, pp. 255-262, 1988.

[16] S. Gallant, *Neural Network Learning and Expert Systems*, Cambridge, MA: MIT Press, 1993.

[17] S. Thrun, “Extracting rules from artificial neural networks with distributed representations,” in *Advances in Neural Information Processing Systems* 7, G. Tesauro, D. Touretzky and T. Leen Eds. Cambridge, MA: MIT Press, 1995.

[18] L. Bochereau and P. Bourgine, “Extraction of semantic features and logical rules from a mult-layer neural network,” *Int. Joint Conf. Neural Networks*, vol. 2, pp. 579-582, 1990.

[19] L. M. Fu, “Rule learning by searching on adapted nets,” *Proc. of the ninth national Conf. Artificial Intelligence*, pp. 590-595, 1991.

[20] G. Towell and J. Shavlik, “The extraction of refined rules from knowledge based neural networks,” *Machine Learning*, vol. 131, pp. 71-101, 1993.

[21] L. M. Fu, “Rule generation from neural networks,” *IEEE Trans. Sys. Man and Cyber.*, vol. 28, no. 8, pp. 1114-1124, 1994.

[22] S. Horikawa, T. Furuhashi and Y. Uchikawa, “On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm,” *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 801-806, Sep. 1992.

[23] H. Okada, R. Masuoka and A. Kawamura, “Knowledge based neural network – using fuzzy logic to initialize a multilayered neural network and interpret postlearning results,” *Fujitsu Scientific and Technical Journal*, vol. 29, no. 3, pp. 217-226, 1993.

[24] S. Mitra, “Fuzzy MLP based expert system for medical diagnosis,” *Fuzzy Sets and Systems*, vol. 65, no. 2-3, pp. 285-296, Aug. 1994.

[25] T. Lofstrom and U. Johansson, “Predicting the benefit of rule extraction: A novel component in data mining,” *HUMAN IT*, 7-3, pp. 78-108, 2005.

[26] R. E. Schapire, “The strength of weak learnability,” *Machine Learning*, vol. 5, pp. 197-227, 1990.

[27] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” *Proc. 13th Int. Conf. Machine Learning*, pp. 148-156, 1996.

[28] N. Ren, “Rule Extraction for Security Analysis Based on Decision Tree Classification Model”, *Master’s Thesis*, Southern Illinois University Carbondale, Major Professor: M. Zargham, May 2004.

[29] R. Mitsdorffer, J. Diederich and C. Tan, “Rule extraction from technology IPOs in the US stock market,” *Proc. 9th Int. Conf. Neural Information Processing*, vol. 5, pp. 2328-2334, 2002.

[30] W. Zheng, “Fuzzy decision tree based rule extraction in security analysis,” *Master’s Thesis*, Southern Illinois University Carbondale, Major Professor: M. Zargham, July 2005.

[31] J. W. Lee, “Stock price prediction using reinforcement learning,” *Proc. IEEE Int. Symp. Industrial Electronics*, vol. 1, pp. 690-695, 2001.

[32] Y. K. Bao, Z. T. Liu, L. Guo and W. Wang, “Forecasting Stock Composite Index by Fuzzy Support Vector Machines Regression,” *Proc. Int. Conf. Machine Learning and Cybernetics*, Vol 6, pp. 3535-3540, Aug. 2005

[33] R. Sharda and R. B. Patil, “A connectionist approach to time series prediction: an empirical test,” in R. R. Trippi, E. Turban, (Eds.), *Neural Networks in Finance and Investing*, pp. 451-464, 1994.

[34] H. Ahmadi, “Testability of the arbitrage pricing theory by neural networks,” *Proc. Int. Conf. Neural Networks*, pp. 385–393, 1990.

[35] J. H. Choi, M. K. Lee and M. W. Rhee, “Trading S& P 500 stock index futures using a neural network,” *Proc. Annual Int. Conf. Artificial Intelligence Applications on Wall Street*, pp. 63–72, 1995.

[36] M. Dong and X. S. Zhou, “Analyzing dividend events with neural network rule extraction,” *Proc. Int. Joint Conf. Neural Networks*, vol. 4, pp. 2854-2859, Jul. 2003.

[37] K. Kohara, T. Ishikawa, Y. Fukuhara and Y. Nakamura, “Stock price prediction using prior knowledge and neural networks,” *Int. J. Intell. Syst. Accounting Finance Manage*, vol. 6, no. 1, pp. 11–22, 1997.

[38] R. Tsaih, Y. Hsu and C. C. Lai, “Forecasting S& P 500 stock index futures with a hybrid AI system, “ *Decision Support Syst.*, vol. 23, no. 2, pp. 161-174, 1998.

[39] T.-S. Quah and B. Srinivasan, “Improving returns on stock investment through neural network selection, “ *Expert Syst. Appl.*, vol. 17 pp. 295–301, 1999.

[40] T. Nishina, M. Hagiwara and M. Nakagawa, “Fuzzy inference neural networks which automatically partition a pattern space and extract fuzzy if-then rules,” *Proc. IEEE Conf. Computational Intelligence*, vol. 2, pp. 1314-1319, Jun. 1994.

[41] M. Magdon-Ismail, A. Nicholson and Y. S. Abu-Mostafa, “Financial markets: very noisy information processing,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2184-2195, Nov. 1998.

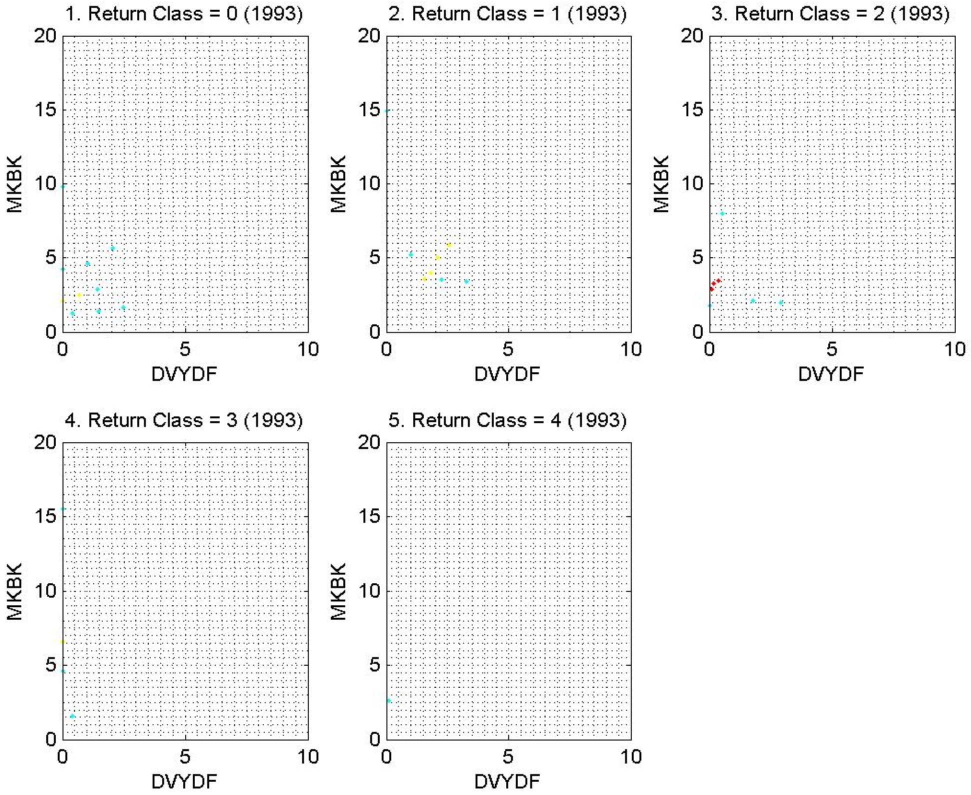
[42] M. R. Zargham and M.R. Sayeh, “A web-based information system for stock election and evaluation,” International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems, pp. 81, 1999.

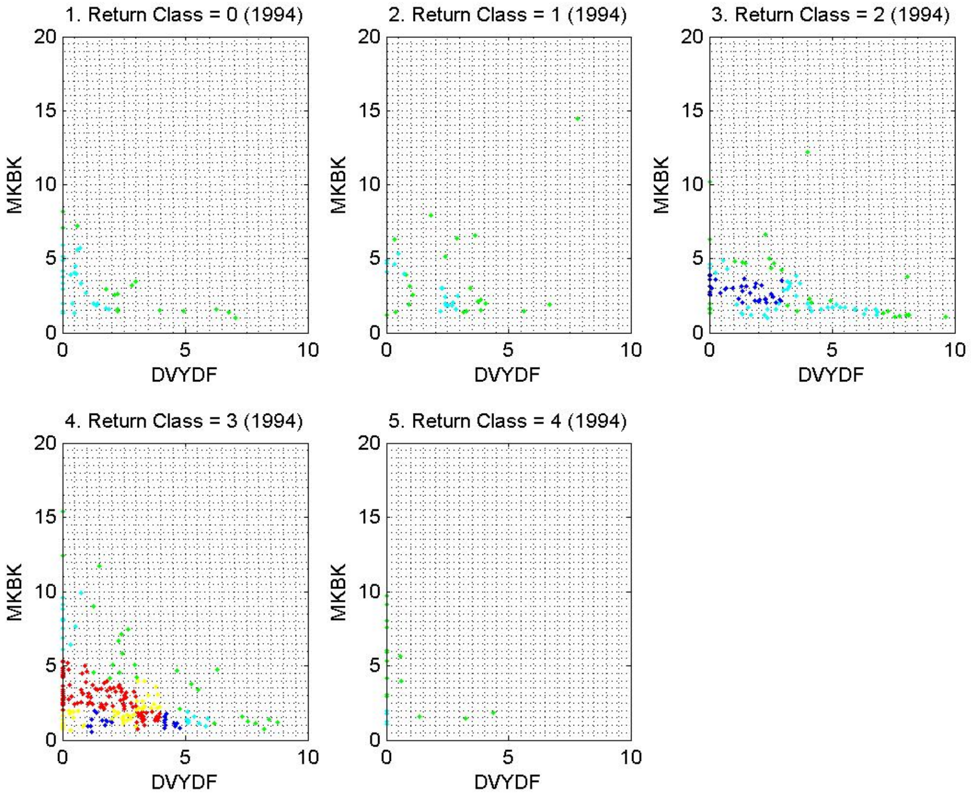
APPENDICES

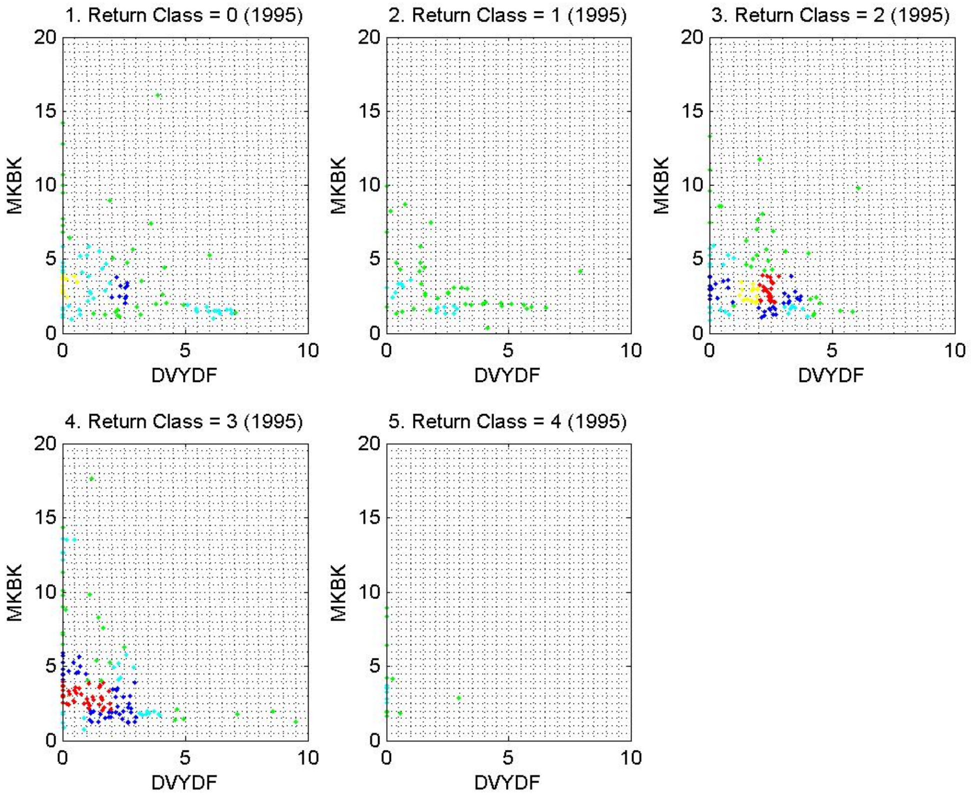
APPENDIX A

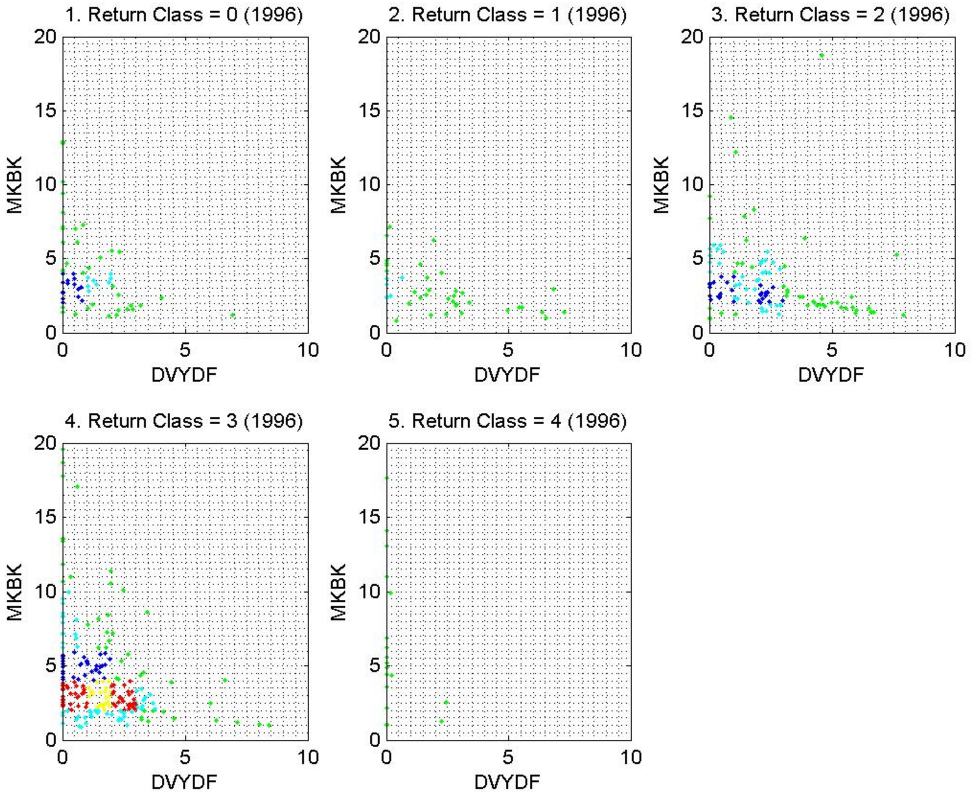
Visualization results for some variable pairs

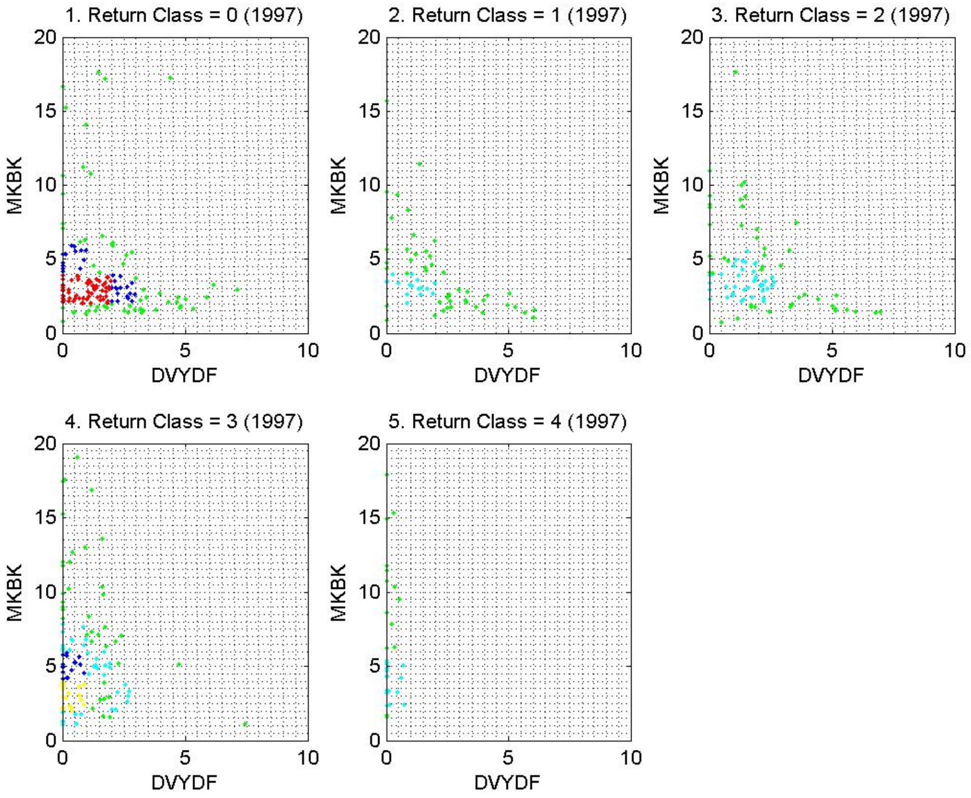
1. Variable pair 10 (DYVDF and MKBK)

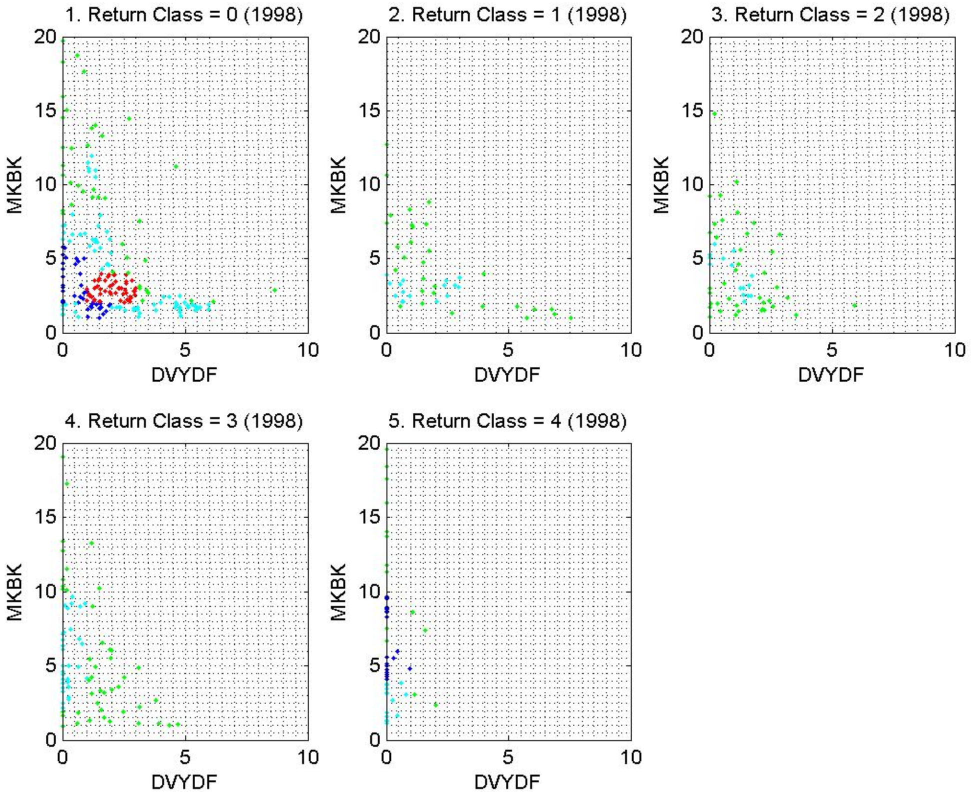



1. Return Class = 0 (1994)
2. Return Class = 1 (1994)
3. Return Class = 2 (1994)
4. Return Class = 3 (1994)
5. Return Class = 4 (1994)
MKBK
DVYDF
0
5
10
15
20

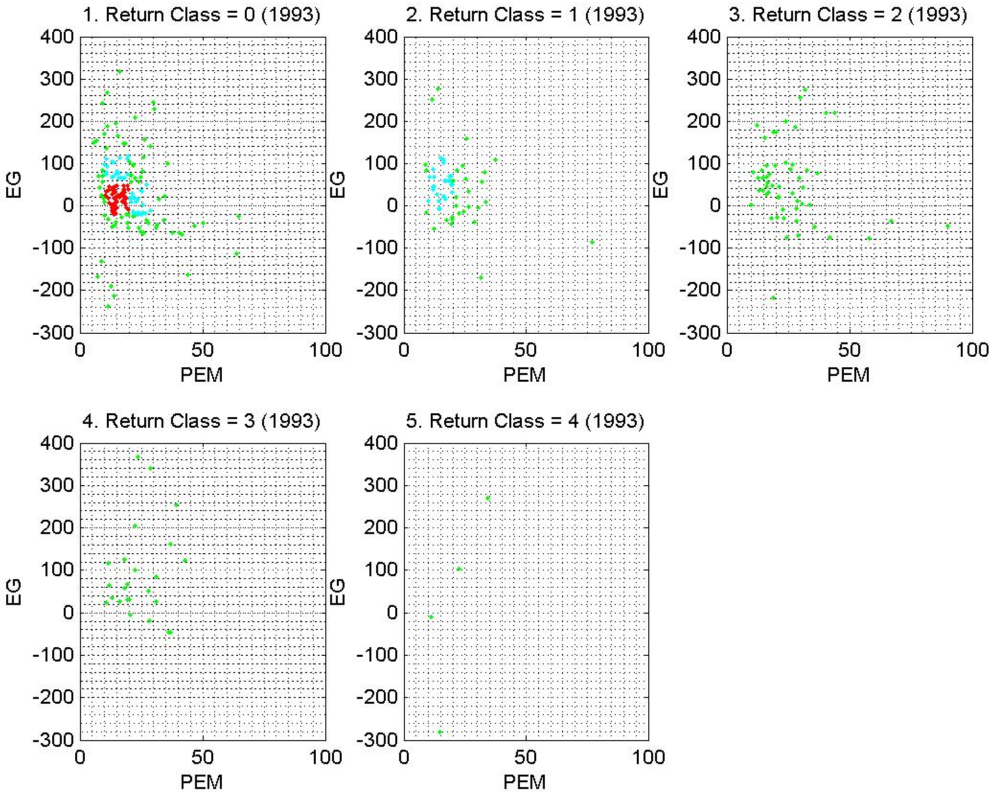

1. Return Class = 0 (1995)
2. Return Class = 1 (1995)
3. Return Class = 2 (1995)
4. Return Class = 3 (1995)
5. Return Class = 4 (1995)
MKBK
DVYDF
0
5
10
15
20


1. Return Class = 0 (1996)
2. Return Class = 1 (1996)
3. Return Class = 2 (1996)
4. Return Class = 3 (1996)
5. Return Class = 4 (1996)
MKBK
DVYDF
0
5
10
15
20


1. Return Class = 0 (1997)
2. Return Class = 1 (1997)
3. Return Class = 2 (1997)
4. Return Class = 3 (1997)
5. Return Class = 4 (1997)
MKBK
DVYDF
0
5
10
15
20



2. Variable pair 21 (PEM and EG)



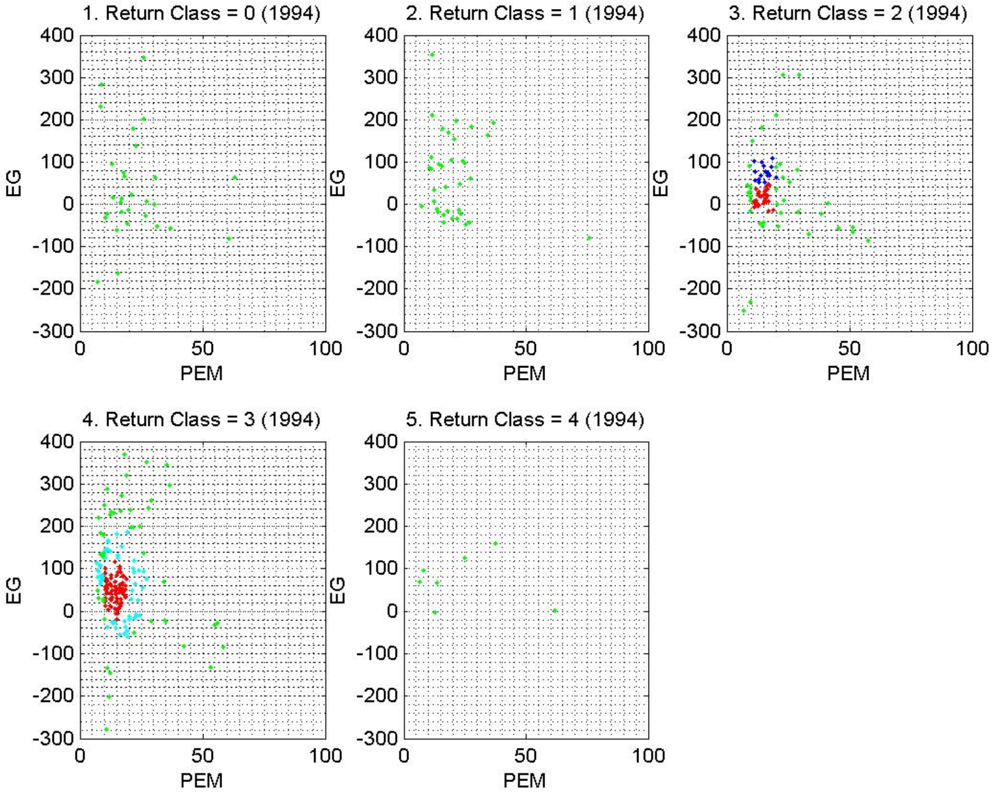
1. Return Class = 0 (1993)

2. Return Class = 1 (1993)

3. Return Class = 2 (1993)

4. Return Class = 3 (1993)

5. Return Class = 4 (1993)



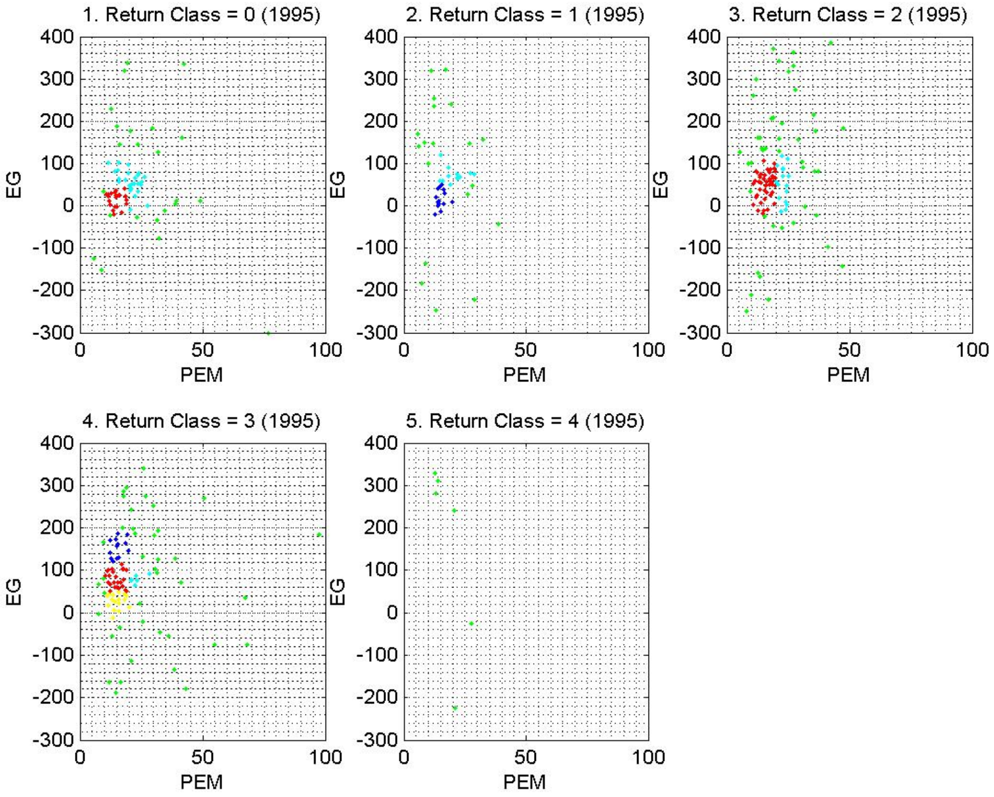
1. Return Class = 0 (1994)

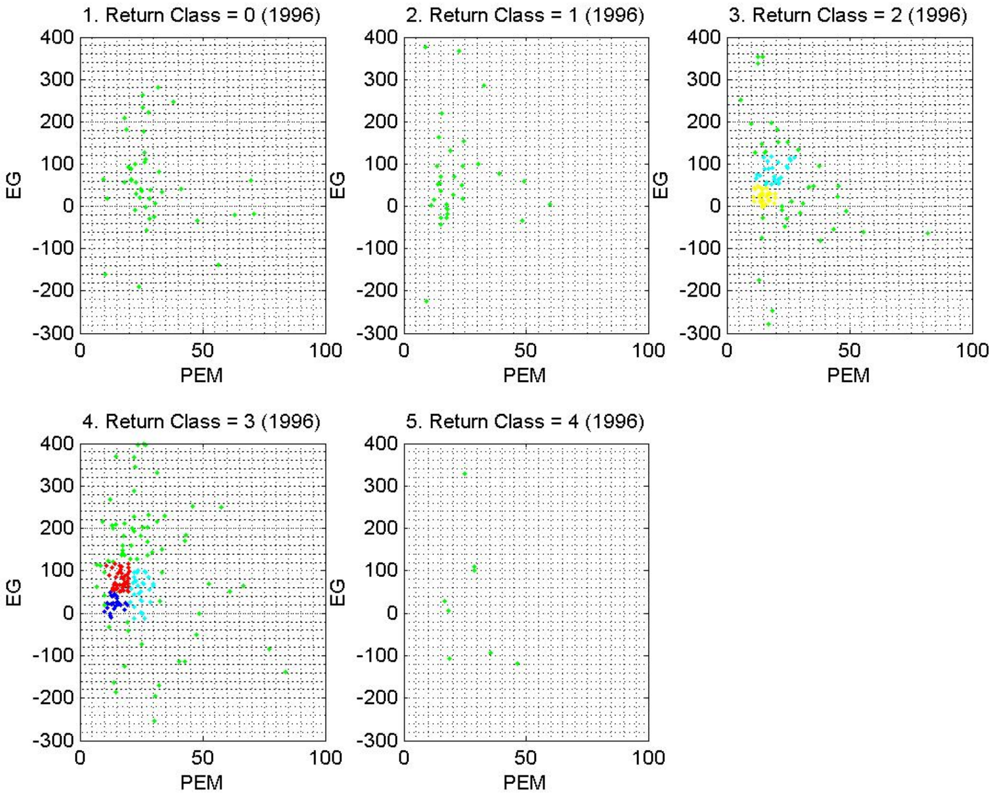
2. Return Class = 1 (1994)

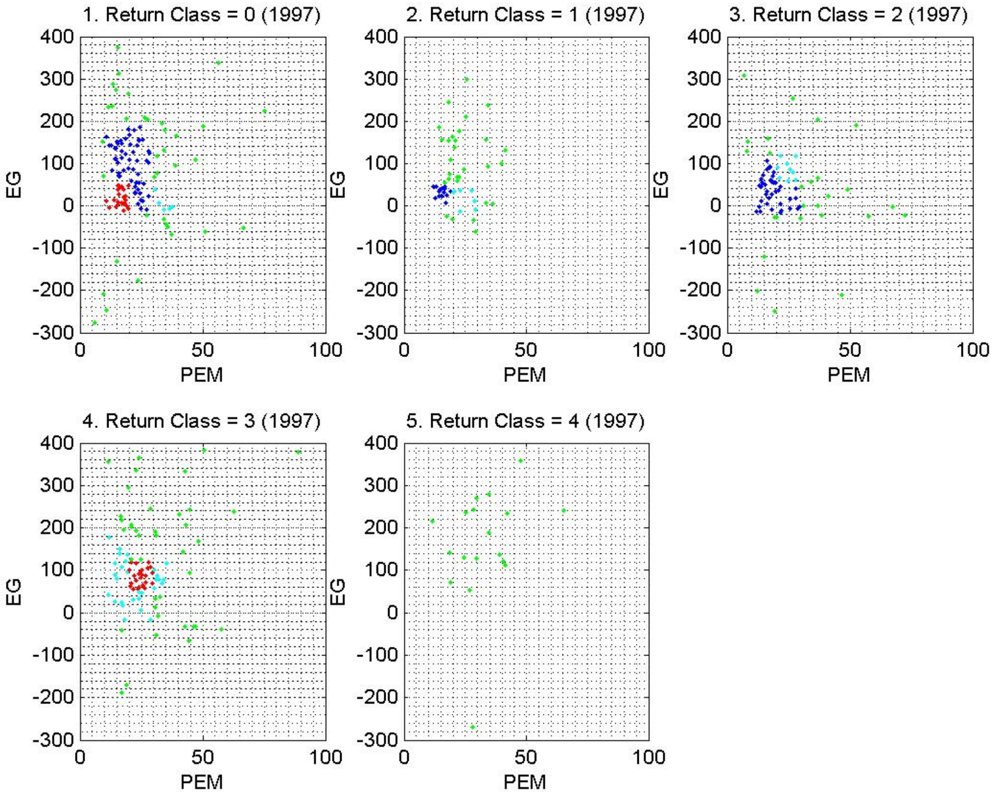
3. Return Class = 2 (1994)

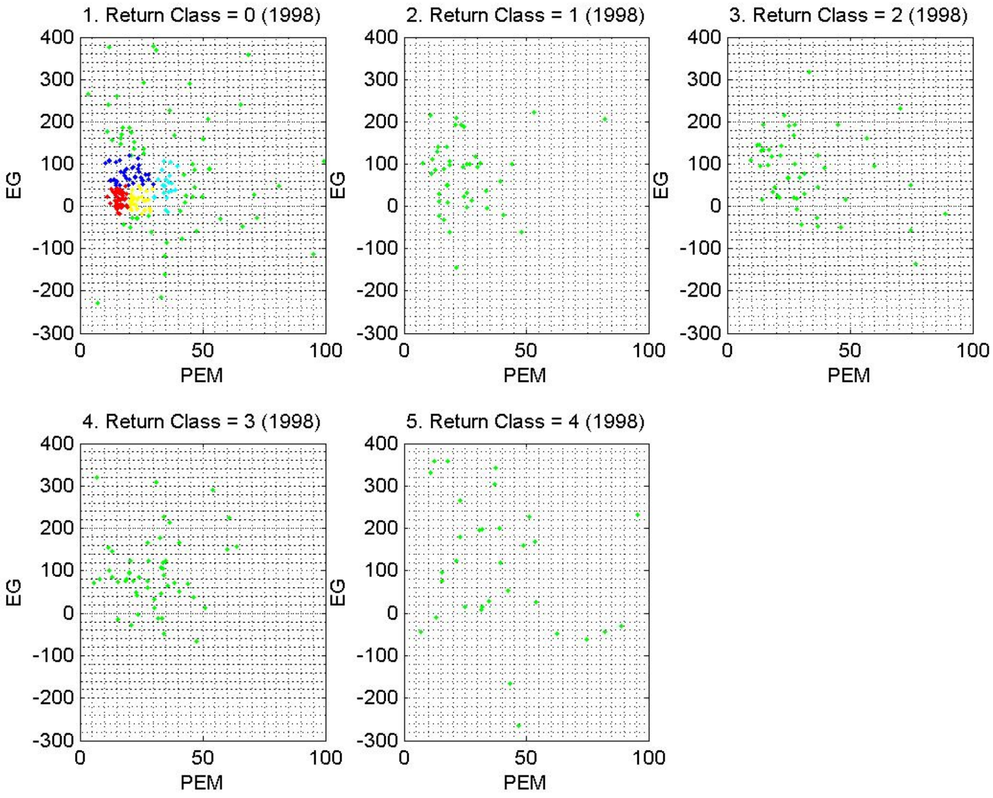
4. Return Class = 3 (1994)

5. Return Class = 4 (1994)

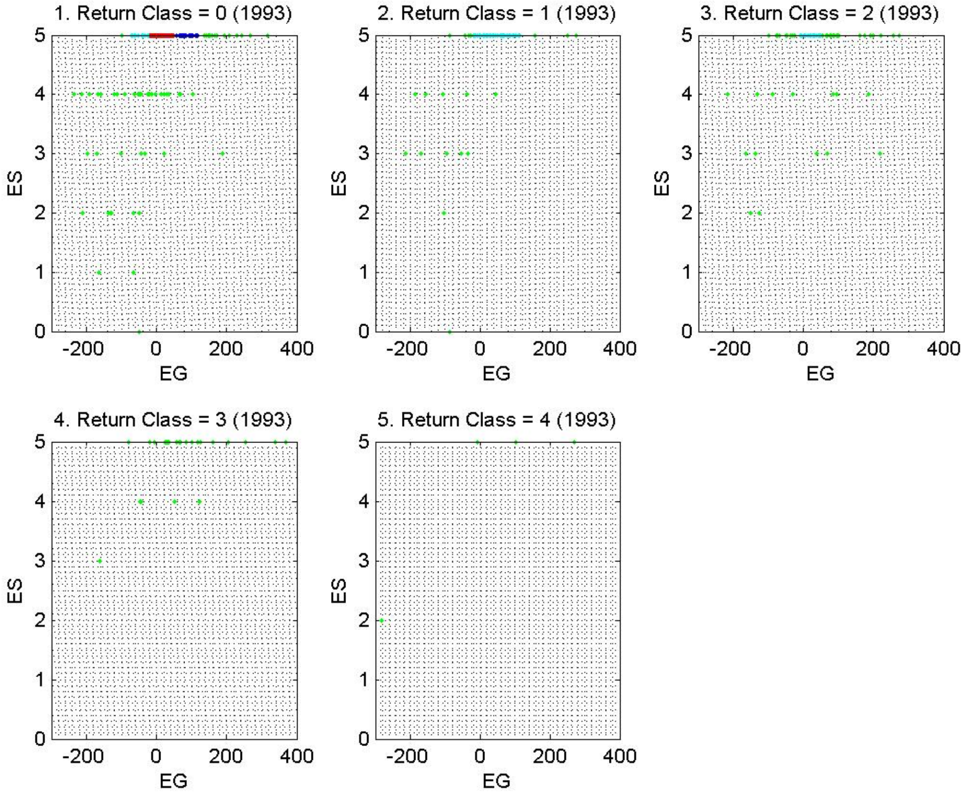

1. Return Class = 0 (1995)
2. Return Class = 1 (1995)
3. Return Class = 2 (1995)
4. Return Class = 3 (1995)
5. Return Class = 4 (1995)
EG
PEM
400
300
200
100
0
-100
-200
-300
0
50
100

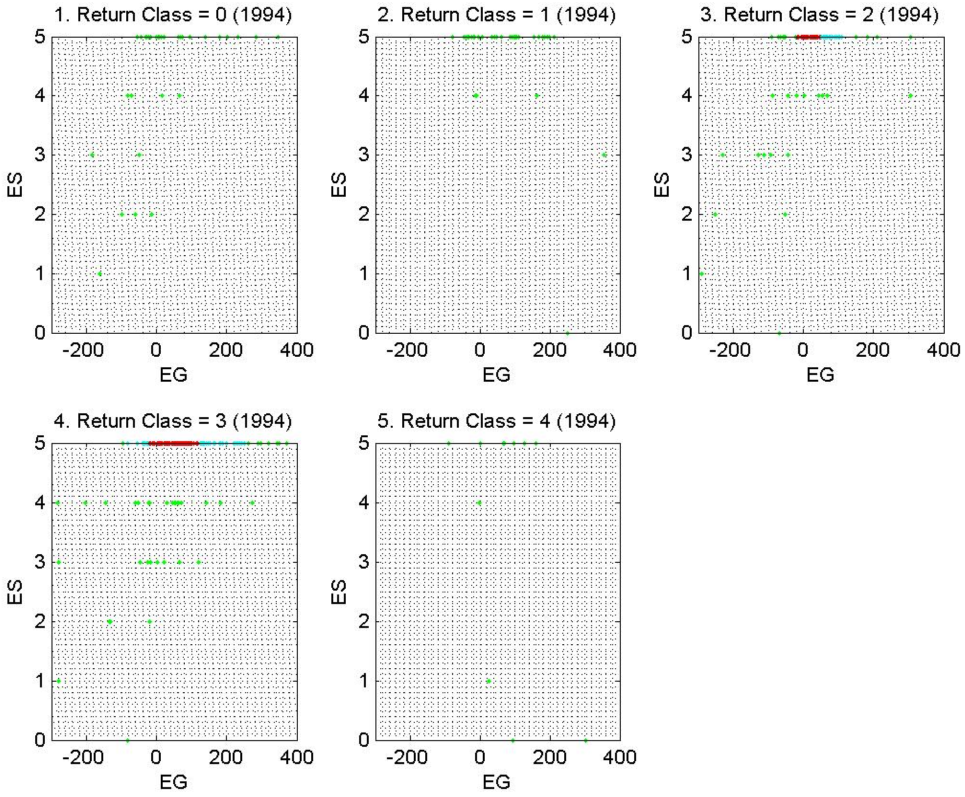

1. Return Class = 0 (1996)
2. Return Class = 1 (1996)
3. Return Class = 2 (1996)
4. Return Class = 3 (1996)
5. Return Class = 4 (1996)
EG
PEM
400
300
200
100
0
-100
-200
-300
0
50
100

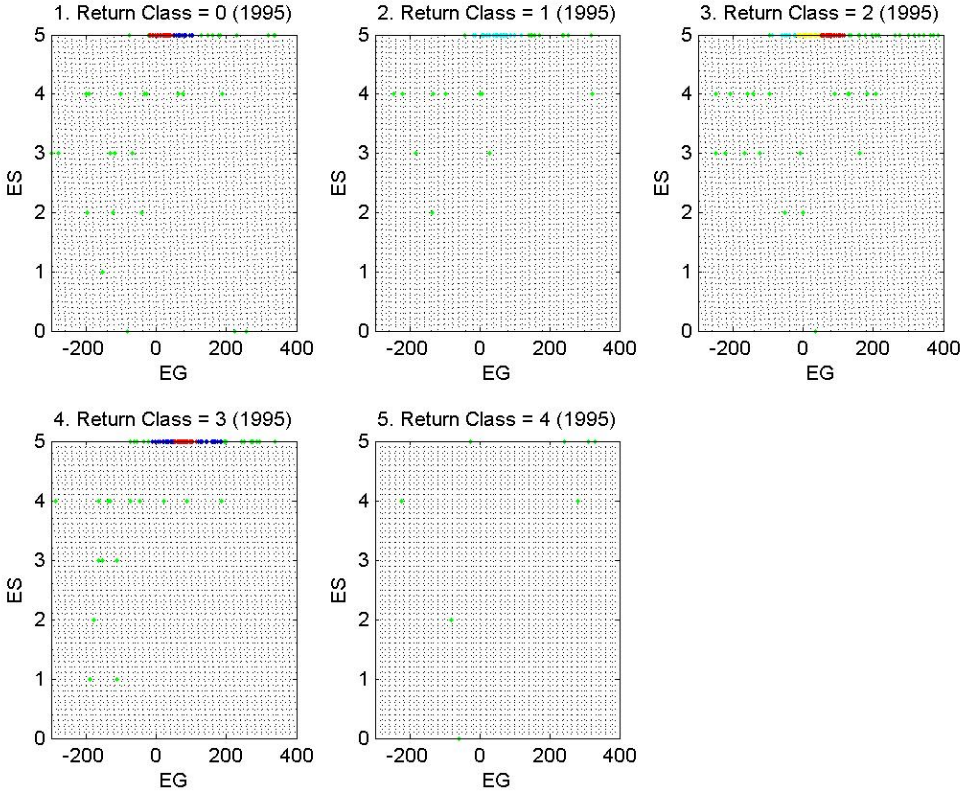

1. Return Class = 0 (1997)
2. Return Class = 1 (1997)
3. Return Class = 2 (1997)
4. Return Class = 3 (1997)
5. Return Class = 4 (1997)
EG
PEM
400
300
200
100
0
-100
-200
-300
0
50
100

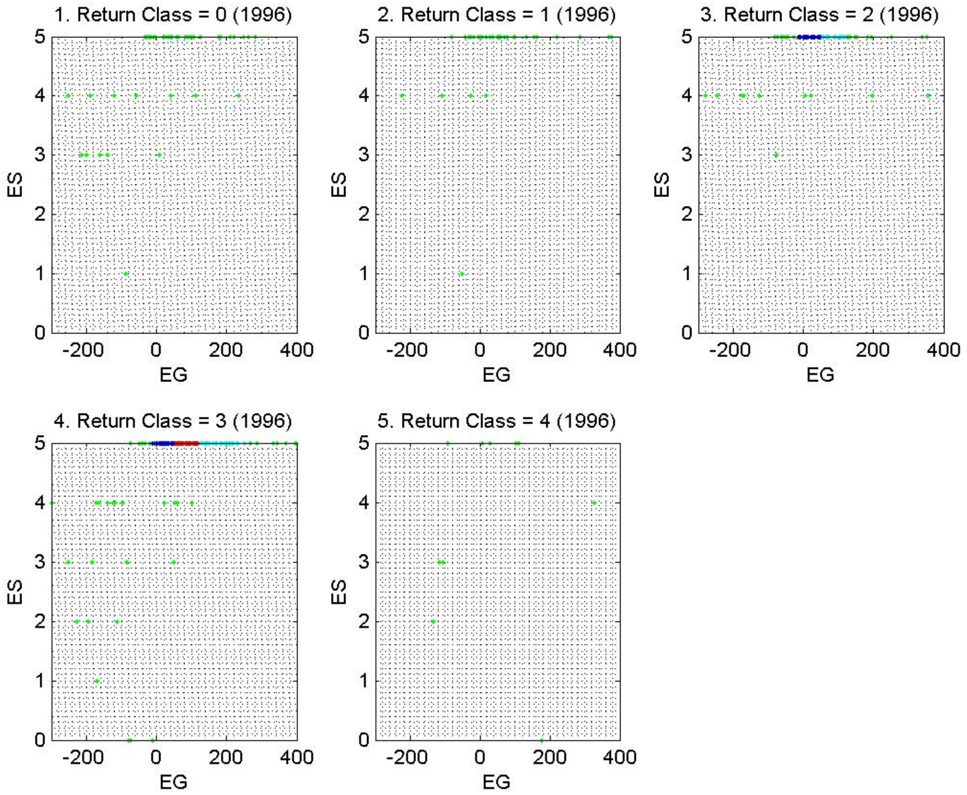

1. Return Class = 0 (1998)
2. Return Class = 1 (1998)
3. Return Class = 2 (1998)
4. Return Class = 3 (1998)
5. Return Class = 4 (1998)
EG
PEM
400
300
200
100
0
-100
-200
-300
0
50
100

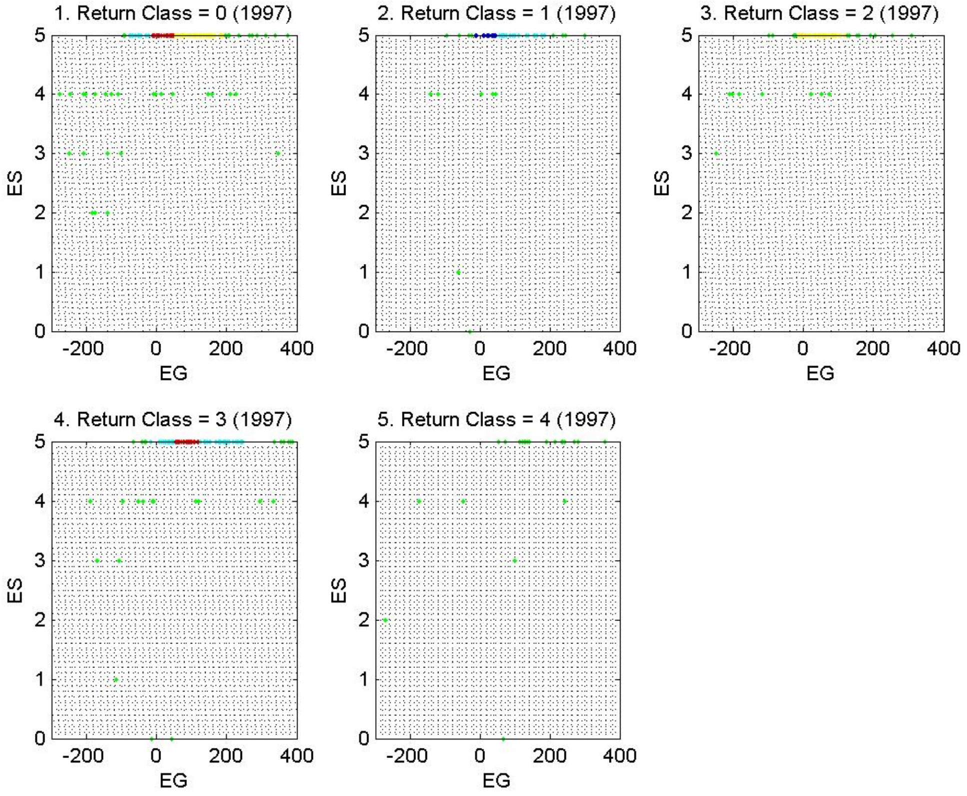
3. Variable pair 28 (EG and ES)

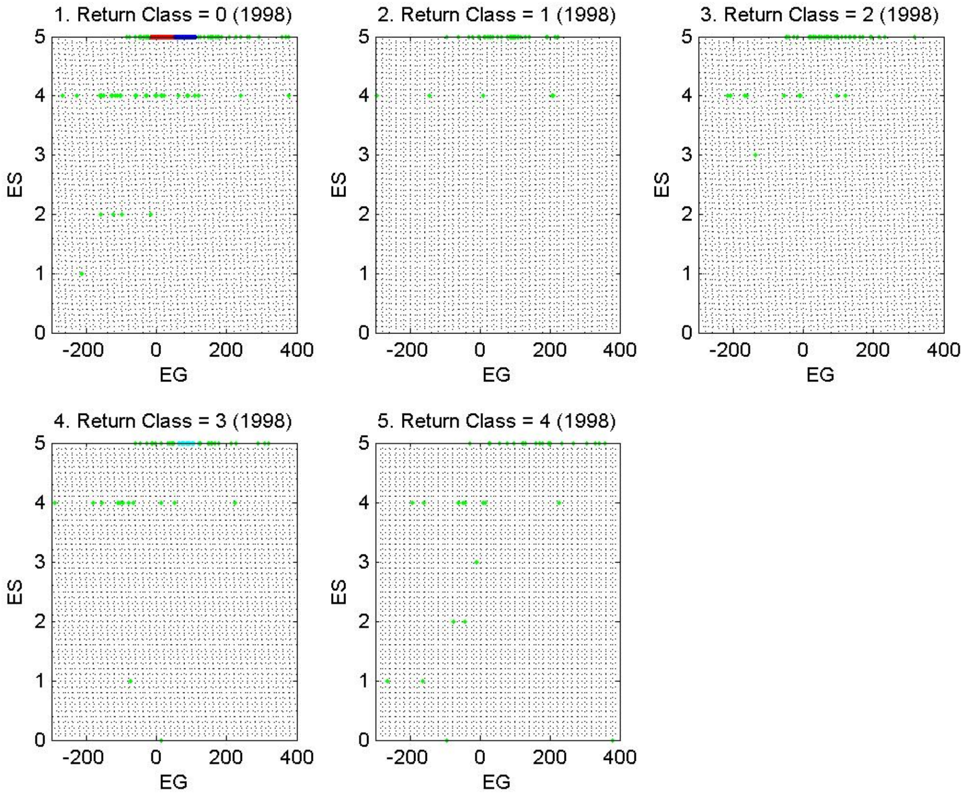



1. Return Class = 0 (1994)
2. Return Class = 1 (1994)
3. Return Class = 2 (1994)
4. Return Class = 3 (1994)
5. Return Class = 4 (1994)
ES
EG
0
1
2
3
4
5
-200
0
200
400


1. Return Class = 0 (1995)
2. Return Class = 1 (1995)
3. Return Class = 2 (1995)
4. Return Class = 3 (1995)
5. Return Class = 4 (1995)
ES
EG
0
1
2
3
4
5
-200
0
200
400


1. Return Class = 0 (1996)
2. Return Class = 1 (1996)
3. Return Class = 2 (1996)
4. Return Class = 3 (1996)
5. Return Class = 4 (1996)
ES
EG
0
1
2
3
4
5
-200
0
200
400


1. Return Class = 0 (1997)
2. Return Class = 1 (1997)
3. Return Class = 2 (1997)
4. Return Class = 3 (1997)
5. Return Class = 4 (1997)
ES
EG
0
1
2
3
4
5
-200
0
200
400



1. Return Class = 0 (1998)

2. Return Class = 1 (1998)

3. Return Class = 2 (1998)

4. Return Class = 3 (1998)

5. Return Class = 4 (1998)

APPENDIX B

The company list according to rule 1 in year 1999

Company Name	MonthPrice (Dec 1999)	MonthPrice (Dec 2000)	Return 2000
ABBOTT LABORATORIES	36.313	48.438	0.3339
ALLERGAN INC	49.75	96.813	0.9460
AMERISOURCEBERGEN CORP	15.188	50.5	2.3250
APOLLO GROUP INC -CL A	8.917	21.861	1.4516
APPLERA CORP APPLIED BIOSYS	60.156	94.063	0.5637
AUTODESK INC	8.438	6.734	-0.2019
BEST BUY CO INC	33.5	19.708	-0.4117
BOSTON SCIENTIFIC CORP	10.938	6.844	-0.3743
BRISTOL-MYERS SQUIBB CO	64.188	73.938	0.1519
CALPINE CORP	16	45.063	1.8164
CISCO SYSTEMS INC	53.563	38.25	-0.2859
COLGATE-PALMOLIVE CO	65	64.55	-0.0069
COMPUWARE CORP	37.25	6.25	-0.8322
CORNING INC	42.979	52.813	0.2288
DELL INC	51	17.438	-0.6580
E TRADE FINANCIAL CORP	26.125	7.375	-0.7177
ECOLAB INC	19.563	21.594	0.1038
EQUIFAX INC	23.563	28.688	0.2175
FAMILY DOLLAR STORES	16.312	21.438	0.3142
FREEPRT MCMOR COP&GLD - CL B	21.125	8.563	-0.5947
GAP INC	46	25.5	-0.4457
GILLETTE CO	41.188	36.125	-0.1229
GUIDANT CORP	47	53.938	0.1476
HARLEY-DAVIDSON INC	32.031	39.75	0.2410
HEINZ (H J) CO	39.813	47.438	0.1915
HERSHEY CO	23.719	32.188	0.3571
HILTON HOTELS CORP	9.563	10.5	0.0980
JABIL CIRCUIT INC	36.5	25.375	-0.3048
KING PHARMACEUTICALS INC	28.031	38.766	0.3830
LILLY (ELI) & CO	66.5	93.063	0.3994
LIMITED BRANDS INC	21.656	17.063	-0.2121
LUCENT TECHNOLOGIES INC	75	13.5	-0.82
MATTEL INC	13.125	14.44	0.1002
MCCORMICK & COMPANY INC	14.875	18.031	0.2122
MCGRAW-HILL COMPANIES	30.813	29.313	-0.0487
MILLIPORE CORP	38.625	63	0.6311
ORACLE CORP	28.016	29.063	0.0374

PARAMETRIC TECHNOLOGY CORP	27.063	13.438	-0.5035
PAYCHEX INC	26.667	48.625	0.8234
PEPSICO INC	35.25	49.563	0.4060
PFIZER INC	32.438	46	0.4181
PROCTER & GAMBLE CO	54.781	39.219	-0.2841
RADIOSHACK CORP	49.188	42.813	-0.1296
SCHERING-PLOUGH	42.375	56.75	0.3392
STAPLES INC	13.833	7.875	-0.4307
STARBUCKS CORP	12.125	22.125	0.8247
STRYKER CORP	17.406	25.295	0.4532
SUN MICROSYSTEMS INC	38.719	27.875	-0.2800
SYSCO CORP	19.781	30	0.5166
TIME WARNER INC	75.875	34.8	-0.5413
TJX COMPANIES INC	10.219	13.875	0.3578
UNIVISION COMMUNICATIONS INC	51.094	40.938	-0.1988
WALGREEN CO	29.25	41.813	0.4295
WATERS CORP	26.5	83.5	2.1509
WYETH	39.25	63.55	0.6191
XTO ENERGY INC	1.813	8.325	3.5918

VITA

Graduate School
Southern Illinois University

Rui Liu Date of Birth: August 18, 1982

81 Jianxin Street, Zhumadian, Henan, China 463000

Beijing University of Posts and Telecommunications
Bachelor of Science, Computer Science and Technology, July 2003

Southern Illinois University Carbondale
Bachelor of Science, Forestry, May 1987

Thesis Title:
CORRELATION BETWEEN FUNDAMENTAL VALUES IN THE FINANCIAL MARKET

Major Professor:
Dr. Mehdi Zargham
Professor and Chair, Department of Computer Science,
Southern Illinois University