**In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.**

**While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.**

PATTERN RECOGNITION OF MUTUAL FUNDS USING SELF-ORGANIZING MAPS

by

Danyu Xu

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Science

Information and System Science

School of Mathematics & Statistics

Carleton University

Ottawa, Ontario

May 2003

©Copyright 2003 Danyu Xu

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88787-1

Our file Notre référence

ISBN: 0-612-88787-1

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

**PATTERN RECOGNITION OF MUTUAL FUNDS
USING SELF-ORGANIZING MAPS**

Submitted by

Danyu Xu

in partial fulfillment of the requirements for the degree of

Master of Science

Information and System Science

Dr. Cyril W.L. Garner
(Director, School of Mathematics & Statistics)

Dr. Shirley E. Mills (Thesis Supervisor)

Carleton University

May 2003

ABSTRACT

The self-organizing map (SOM) is a data mining technique that projects a high-dimensional input data space onto (usually) a two-dimensional grid visualization space. The mapping roughly preserves the most important topological and metric relationships among the original data and inherently clusters the data. In this thesis, the pattern recognition and selection of mutual funds are investigated by using existing, as well as improved, SOM technology. Results show that the SOM is an efficient tool for clustering and pattern recognition of mutual funds. The SOM Toolbox is improved by implementing an algorithm to provide explicit clustering that better matches the U-matrix visualizations than existing clustering algorithms. Additional post-clustering descriptive summaries and boxplots are provided to enhance interpretation of the SOM. The proposed improved SOM also provides a better basis than traditional methods for both mutual fund portfolio management and mutual fund performance comparisons.

ACKNOWLEDGMENTS

I wish to express my sincere gratitude and appreciation to

- My supervisor Professor Shirley E. Mills, and co-supervisor Professor Edward J. Norminton for introducing me to the data mining fields, which is very interesting and exciting. Their guidance and encouragement was invaluable to me during the years of my master's studies.
- The other members of my committee for their time and interest.
- The faculty and graduate students in the School of Mathematics & Statistics who provided friendly atmosphere during my study in Carleton.
- My family for unconditional love and support throughout all my study years.

TABLE OF CONTENTS

TITLE PAGE	I
ACCEPTANCE SHEET	II
ABSTRACT	III
ACKNOWLEDGEMENTS.....	IV
TABLE OF CONTENTS.....	V
LIST OF FIGURES	VIII
LIST OF TABLES	IX
CHAPTER 1: INTRODUCTION	1
1.1 MOTIVATION.....	1
1.2 THESIS OBJECTIVE AND LIMITATION	3
1.3 ORGANIZATION OF THE THESIS	4
CHAPTER 2: DATA MINING	6
2.1 THE PROCESS OF KNOWLEDGE DISCOVERY	6
2.2 DATA MINING TECHNIQUES	8
2.2.1 CLUSTERING.....	8
2.2.2 PROJECTION.....	10
2.2.3 LINEAR REGRESSION.....	11
2.2.4 ARTIFICIAL NEURAL NETWORKS.....	11
CHAPTER 3: SELF-ORGANIZING MAPS	13
3.1 STRUCTURE	13
3.2 BASIC ALGORITHM.....	15
3.3 SOM VISUALIZATION.....	19
3.3.1 CLUSTER STRUCTURE AND SHAPE.....	19
3.3.2 COMPONENTS OF THE MAP	21
3.4 MAP QUALITY	22
3.4.1 MAPPING PRECISION	22
3.4.2 TOPOLOGY PRESERVATION.....	23

3.5	SOM CLUSTERING	24
CHAPTER 4: PATTERN RECOGNITION OF MUTUAL FUNDS.....		26
4.1	DATA.....	26
4.2	BASIC DATA ANALYSIS.....	27
4.2.1	DATA SELECTION	27
4.2.2	DATA PRE-PROCESSING	28
4.2.2.1	<i>Statistical Summary</i>	28
4.2.2.2	<i>Missing Data</i>	29
4.2.2.3	<i>Extreme Values</i>	30
4.2.2.4	<i>Normalization of the Data</i>	30
4.2.2.5	<i>Correlation Hunting</i>	31
4.3	TRAINING SOM MAPS.....	31
4.3.1	DETERMINE TRAINING PARAMETERS	32
4.3.2	CREATING SUITABLE MAPS	33
4.4	CLUSTERING AND VISUALIZATION	34
4.4.1	IMPROVING EXPLICIT CLUSTERING	34
4.4.2	POST CLUSTERING SUMMARY AND BOXPLOT	37
4.5	RESULT AND INTERPRETATION	38
4.5.1	BOND FUNDS PATTERN RECOGNITION	38
4.5.1.1	<i>Distance Matrix Visualization</i>	40
4.5.1.2	<i>Discovered Data Structure</i>	41
4.5.1.3	<i>Comparisons of Clustering Results</i>	44
4.5.1.4	<i>Quantization Error Analysis</i>	49
4.5.2	STOCK FUNDS PATTERN RECOGNITION	50
4.5.2.1	<i>Distance Matrix Visualization</i>	51
4.5.2.2	<i>Discovered Data Structure</i>	52
4.5.2.3	<i>Comparisons of Clustering Results</i>	56
4.5.2.4	<i>Quantization Error Analysis</i>	57
4.5.3	MUTUAL FUNDS SELECTION	58
4.5.3.1	<i>Natural Resource Stocks</i>	59
4.5.3.2	<i>Financial Stocks</i>	62
4.5.3.3	<i>Aggressive Growth Stocks</i>	64
CHAPTER 5: CONCLUSION		66
5.1	RESULTS.....	66
5.2	CONTRIBUTION	67
5.3	FUTURE WORK.....	67
BIBLIOGRAPHY		68

APPENDIX A: HOW TO USE THE FUND DATA.....	73
APPENDIX B: CORRELATION MATRICES	77
APPENDIX C: SCATTERPLOT MATRICES	78
APPENDIX D: HISTOGRAMS OF VARIABLES	79
APPENDIX E: CLUSTER POST-DESCRIPTIVE SUMMARY	85
APPENDIX F: BOX PLOT OF POST-CLUSTERING SUMMARY	88
APPENDIX G: MANOVA RESULTS OF BONDS AND STOCKS.....	92
APPENDIX H: STATISTICAL SUMMARY OF EXISTING CLUSTERING OF BONDS AND STOCKS.....	96
APPENDIX I: SYMBOLS OF MUTUAL FUNDS IN CLUSTER OF BOND FUNDS	98
APPENDIX J: FLEXIBILITY CHECK OF PROPOSED METHOD	99
APPENDIX K: SOURCE CODE.....	100

LIST OF FIGURES

<i>Number</i>	<i>Page</i>
FIGURE 1. KNOWLEDGE DISCOVERY PROCESS	7
FIGURE 2. HIERARCHY CLUSTERING	9
FIGURE 3. A NEURAL NETWORK WITH TWO HIDDEN LAYERS.....	12
FIGURE 4. STRUCTURE OF SELF-ORGANIZING MAPS.	14
FIGURE 5. TWO EXAMPLES OF TOPOLOGICAL NEIGHBOURHOOD.....	15
FIGURE 6. UPDATING THE BEST MATCHING UNIT (BMU) AND ITS NEIGHBOURS	17
FIGURE 7. THE REPRESENTATION OF DISTANCE MATRIX TECHNIQUES.....	20
FIGURE 8. U-MATRIX AND ITS COMPONENT PLANES	21
FIGURE 9. VISUALIZATION OF DATA ON THE MAP	23
FIGURE 10. TWO-LEVEL APPROACH OF SOM CLUSTERING	24
FIGURE 11. PARTITIONING OF MUTUAL FUNDS.....	27
FIGURE 12. U-MATRIX AND ITS EXPLICIT CLUSTERING BASED ON THE EXISTING CLUSTERING ALGORITHMS OF SOM TOOLBOX FOCUSED ON BOND FUNDS.	35
FIGURE 13. U-MATRIX AND ITS EXPLICIT CLUSTERING BASED ON THE EXISTING CLUSTERING ALGORITHMS OF SOM TOOLBOX FOCUSED ON STOCK FUNDS.	35
FIGURE 14. THE U-MATRIX AND COMPONENT MAPS FOCUSED ON BOND FUNDS.	40
FIGURE 15. DISTANCE MATRIX TECHNIQUES OF SOM VISUALIZATION THAT FOCUSED ON BOND FUNDS.	41
FIGURE 16. TWO EXPLICIT CLUSTERINGS OF SOM BASED ON THEIR U-MATRIX VISUALIZATION (FOCUSED ON BOND FUNDS).....	42
FIGURE 17. BOXPLOT OF 1-YEAR RETURN FOR EACH CLUSTER FOCUSED ON BOND FUNDS.	43
FIGURE 18. QUANTIZATION ERROR PER NEURON OF BONDS CLUSTERS.....	49
FIGURE 19. THE U-MATRIX AND COMPONENT MAPS THAT ARE FOCUSED ON THE STOCK FUNDS.....	51
FIGURE 20. DISTANCE MATRIX TECHNIQUES OF SOM VISUALIZATION THAT FOCUSED ON STOCK FUNDS.	52
FIGURE 21. THE EXPLICIT CLUSTERS OF SOM (FOCUSED ON STOCK FUNDS).	52
FIGURE 22. BOXPLOT OF 1-YEAR RETURN OF CLUSTERS FOCUSED ON STOCK FUNDS.	53
FIGURE 23. QUANTIZATION ERROR PER NEURONS OF STOCKS CLUSTERS.	57
FIGURE 24. U-MATRIX AND COMPONENTS MAP OF 49 MUTUAL FUNDS IN NATURAL RESOURCES SECTOR.....	61
FIGURE 25. U-MATRIX AND LABELED MAP OF 49 MUTUAL FUNDS THAT ARE FOCUSED ON INVESTMENT IN NATURAL RESOURCES SECTOR.	61
FIGURE 26. SOM OF 18 MUTUAL FUNDS FOCUSED ON FINANCIAL STOCKS.....	63
FIGURE 27. SOM OF 20 MUTUAL FUNDS FOCUSED ON AGGRESSIVE-GROWTH STOCKS THAT HAVE FRONT- LOAD.....	64

LIST OF TABLES

<i>Number</i>	<i>Page</i>
TABLE 1. SUMMARY STATISTICS PER VARIABLE (BOND FUNDS).....	28
TABLE 2. SUMMARY STATISTICS PER VARIABLE (STOCK FUNDS).....	29
TABLE 3. COUNTS AND AVERAGES FOR BOND FUNDS USING CENTROID.....	36
TABLE 4. COUNTS AND AVERAGES FOR STOCK FUNDS USING CENTROID	36
TABLE 5. ERRORS OF DIFFERENT TRAINING LENGTH (FOR BOND FUNDS)	39
TABLE 6. CLUSTER POST-DESCRIPTIVE SUMMARY FOR BOND FUNDS	42
TABLE 7. MANOVA ANALYSIS COMPARISON OF TWO CLUSTERING METHODS FOR BOND FUNDS.....	48
TABLE 8. CLUSTER QUANTIZATION ERROR FOR BOND FUNDS	50
TABLE 9. CLUSTER SUMMARY OF STOCK FUNDS.	54
TABLE 10. MANOVA ANALYSIS COMPARISON OF TWO CLUSTERING METHODS FOR STOCK FUNDS	56
TABLE 11. CLUSTER QUANTIZATION ERROR FOR STOCK FUNDS.	58
TABLE 12. ANALYSIS OF MUTUAL FUNDS FOCUSED ON NATURAL RESOURCES SECTOR	62
TABLE 13. ANALYSIS OF MUTUAL FUNDS FOCUSED ON FINANCIAL STOCKS	63
TABLE 14. ANALYSIS OF MUTUAL FUNDS FOCUSED ON AGGRESSIVE GROWTH STOCKS.....	65

Chapter 1

INTRODUCTION

This chapter presents motivation, thesis objective and limitation, and organization of this thesis.

1.1 Motivation

Many people are interested in mutual funds investment. As investors, they want to find suitable funds among the thousands of stocks and bonds available on the market today in order to achieve their investment goals. As financial analysts or advisors, they want to compare the performance of funds, in order to improve risk and portfolio management. A tool commonly used for such purpose is financial competitor benchmarking (Bendell et al., 1998). Financial benchmarking can be conducted in many ways, the simplest way is to produce statistical summaries of the data - for example, the smallest and largest data values, the median, the standardized deviation, and so on. Another frequently used method is to use an ordinary spreadsheet program to compare financial figures.

When using a spreadsheet program, it is easy to compare a few funds at a time according to some important factors. However, if many factors - such as total return, fund size, expense ration, yield, management, etc., are to be taken into account at once, a spreadsheet program is no longer useful. In addition, common statistical methods including clustering are not very suitable for handling very high-dimensional data, nor for reducing the dimensionality of a data

set or visualizing the data. Neural networks, in the form of self-organizing maps (SOM), can provide a means for clustering and visualizing of massive amounts of data.

A self-organizing map is a form of artificial neural network that is based on ‘unsupervised’ learning (Kohonen, 1997). It maps a high-dimensional input space onto a two-dimensional array of neurons, providing a means of visualizing an abstraction of the original data, and combining two important techniques: clustering and visualization. Over 5300 research papers have been published since it was invented in 1981 (Kohonen, 2000). SOM has been used in such diverse fields as image object classification (Becanovic, 2000), cloud classification (Ambroise et al., 2000), internet traffic classification (Raivio et al., 2000), knowledge extracting from text documents (Visa et al., 2000), and breast cancer diagnosis (Chen et al., 2000). Several textbooks have been published containing the theory and applications developed using SOM (Kohonen, 1997; 2001; Deboeck and Kohonen, 1998) and various software packages have been developed facilitating the application of the SOM algorithm.

A recent book by Deboeck and Kohonen, (1998) contains a dozen examples of SOM application. Their Chapter 3, “Picking Mutual Funds with Self-organizing Maps”, shows how SOM can be used in mutual funds field. It provides a significant improvement over the information that is traditionally published on mutual funds and demonstrates clear differences and patterns among the best-rated mutual funds using data published by Morningstar (<http://www.morningstar.com>). However, this was a very restricted application based only on the seemingly similar funds under the same category that were selected on the basis of an expert's experience and multiple criteria, e.g. the return achieved over one, three and five years, the risk of volatility, the size of the fund and the expense ratio, etc. It contained only a small number of mutual funds that were visualized and clustered on a ‘big map’ (also called ‘big

SOM’; a more detailed explanation of this concept and ‘small SOM’ is in Section 3.1 and Section 4.3.2). The application of SOM to the existing wide range of available mutual funds and an abstract picture of data structures or ‘small SOM’ for massive datasets was not performed nor was there an in-depth examination for different categories of mutual funds. This thesis addresses all of these concerns.

1.2 Thesis Objective and Limitation

If observations contain large amounts of time series data where patterns are identified, pattern recognition can serve as a predictive model to predict future events. If observations are complex and high dimensional clustering may serve as an intermediate step for further data analysis or modeling (e.g. there may exist no obvious statistical model based on the entire dataset, but there may exist some clusters for which statistical models for partial prediction can be established.) The goal of this thesis is to explore two kinds of SOMs – a ‘small SOM’ and a ‘big SOM’ as mentioned previously - to do mutual fund pattern recognition and mutual fund selection based on the knowledge discovery process and the existing software package ‘SOM Toolbox’ (Vesanto et al, 2000) and where possible to extend capabilities. The objectives of this thesis are:

- Carry out a review of existing ‘big SOM’ techniques using a much larger and more complex mutual funds dataset to verify the suitability of SOM as a decision support tool along the lines of Deboeck’s presentation in “Picking Mutual Funds with Self-organizing Maps”.

- Create small maps, using a much larger and more complex mutual funds dataset than Deboeck's, to find interesting groups for further data analysis based on pattern recognition.
- Improve explicit clustering by implementing an algorithm that produces clusters that more closely match the U-matrix visualization.
- Due to the limitation of the data source we have and the scope of this thesis, no attempt is made to build models based on the discovered structure. However, pattern recognition using SOM appears to be a useful tool for understanding possible cluster structure in a massive and complex mutual fund dataset and can provide valuable information for mutual fund investment strategies.

1.3 Organization of the Thesis

This thesis is organized into five chapters: Chapter 2 introduces the knowledge discovery process and some commonly used data mining techniques. In Chapter 3, the basic SOM algorithm and its properties are presented; the chapter is mainly based on the "Self-organizing Maps" book, written by Kohonen (1997). In Chapter 4, we use the knowledge discovery process, applying the SOM algorithm to mutual funds pattern recognition with primary focus on obtaining an abstract map to improve detection of data structure and cluster visualization. An algorithm for improving explicit clustering along with a post-clustering summary and corresponding boxplot is presented. Discovered data structures are then examined and interpreted. A detailed comparison of the improved SOM clustering algorithm with the existing algorithm is provided. We also apply SOM for mutual funds selection, based on funds

that seem similar under the same category, to support the portfolio management and performance comparison. Chapter 5 summarizes some important results of this thesis, along with the contribution it makes and recommendations for future research.

Chapter 2

DATA MINING

"Data mining is the process of discovering meaningful correlations, patterns and trends by sifting through large amounts of data, using pattern recognition technologies as well as statistical and mathematical techniques" (Moxon, 1996). It has also been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data" (Frawley et al, 1992). It may be viewed as the process of analyzing massive amounts of data from different perspectives, finding hidden patterns, and summarizing results into useful information.

The techniques used to gain access to the knowledge contained in a stored dataset are referred to as data mining techniques. All the steps necessary to extract knowledge from a large dataset are known as the process of knowledge discovery.

2.1 The Process of Knowledge Discovery

Extracting potentially useful information or knowledge from a database consists of several steps (Fayyad et al, 1996) as shown in Figure 1.

Figure 1. Knowledge Discovery Process

The basic steps of the knowledge discovery process involve:

1. Understanding the knowledge of the underlying domain and the goal of the research.
2. Pre-processing data, including the selection of the data sources, the removal of noise or outliers, the treatment of missing data, the transformation and reduction of data, etc.
3. ‘Mining’ the data to extract patterns or models hidden in the dataset.
4. Interpreting (post-processing) discovered knowledge, especially the interpretation in terms of description and prediction.
5. Putting discovered knowledge to practical use.

2.2 Data Mining Techniques

Data mining can be used for description and prediction purposes.

- Description focuses on finding interpretable patterns that describe the data. Commonly used techniques here are clustering and projections.
- Prediction involves using some known variables to predict other interesting variables. Commonly used techniques include regression and artificial neural networks.

2.2.1 Clustering

Clustering is often one of the first steps in data mining analysis. According to a standard textbook on pattern recognition (Ripley, 1996) “Clustering algorithms are methods to divide a set of n observations into g groups so that members of the same group are more alike than members of different groups”. Since the main purpose of clustering is to find clusters or groups where members of the same group are more alike than members of other groups, one has to have a means of measuring the “likeness”. The most common measure of likeness (similarity or dissimilarity) between two members is the distance between them.

Clustering methods can be divided into two basic types: hierarchical and non-hierarchical clustering methods. Within each, there exist different approaches and algorithms.

Figure 2. Hierarchy Clustering

Hierarchical clustering can be sub-divided into agglomerative and divisive algorithms, corresponding to bottom-up and top-down strategies for building a hierarchical tree as illustrated in Figure 2. The agglomerative hierarchy is formed bottom-up. Starting with each observation as a cluster, then small clusters are combined into bigger clusters at each level of the hierarchy until at the top all the observations objects are in one cluster. Divisive hierarchical clustering is the reverse process of agglomeration.

Non-hierarchical clustering decomposes the dataset into disjoint clusters. These algorithms assign clusters to dense regions in the data space, i.e. a cluster is formed where there are similar items.

K-means is one form of non-hierarchical clustering. K-means produces k clusters from a set of n data values, such that the squared-error is minimized. The k-means algorithm is as follows:

- (1). Select k objects as initial centers;
- (2). Assign each datum to the closest center;
- (3). Recalculate the centers of each cluster;
- (4). Repeat step 2 and 3 until centers do not change;

In k-means clustering choosing of the number of clusters k is critical. Changing k may result in many different cluster formations.

2.2.2 Projection

Projection methods are used to reduce dimensionality of the data, in the hope of making interpretation and analysis easier. Projection techniques can be linear or non-linear.

Linear techniques form linear combinations of components of the original data to project the data onto a new co-ordinate system of lower dimensionality. Principal Component Analysis (PCA) is one such method (Johnson and Wichern, 1998). The PCA technique finds the directions of greatest variability in the data. These directions are called ‘principal components’. The first principal component describes the direction of the largest variation in the data. The second principal component, orthogonal to the first, describes the direction of the second largest variation in the data, etc.

If a dataset is high dimensional and its distribution highly nonsymmetrical, visualizing its distributional structure using linear projection techniques may not prove satisfactory. Several **non-linear projection techniques** exist for projecting non-linear, high-dimensional structures onto a low-dimensional display. Multi-dimensional scaling (MDS) is one such method. It

places the original, high dimensional data points on a lower dimensional display in such a way that the relative rank ordering of “similarity” between data in the input space is preserved as closely as possible (i.e. the new distance between the two least similar observations is largest, and vice versa, the new distance between the two most similar observations is smallest).

2.2.3 Linear Regression

Multiple linear regression is used to model the relationship between a dependent variable and several independent variables. The regression equation has the following form:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik} + \varepsilon_i, i = 1, \dots, n,$$

where Y is the dependent variable, $X_1, \dots, X_k$ are the independent variables, and i is the index of the n sample observations. The disturbance term ε models random influences that we are unable to capture with the model, such as measurement errors. The coefficients $(\beta_1, \dots, \beta_k)$ of the independent variables and the intercept β_0 are often estimated using the Ordinary Least Squares (OLS) technique. A full overview of the multiple linear regression model is given in Greene (1997).

2.2.4 Artificial Neural Networks

Artificial neural networks are used to represent complex input and output relationships. Interconnected neurons are ordered in two or more layers. At a minimum there should be an input layer and an output layer; any layers in between are referred to as hidden layers. These neurons are linked to their neighbours with interconnection weights. Learning is

accomplished by adjusting these weights to cause the network to produce appropriate output results. An example of a neural network is shown in Figure 3.

Figure 3. A neural network with two hidden layers
(Source: <http://www.nd.com/welcome/whatisnn.htm>).

The neural network process of Figure 3 is as follows:

Inputs are fed into the input layer and multiplied by interconnection weights, then summed and processed by a nonlinear function (often sigmoid) within the first hidden layer. The processed data is again multiplied by interconnection weights, then summed and processed by a nonlinear function within the second hidden layer. Finally, the resulting data is multiplied by interconnection weights, then processed within the output layer to produce the final neural network output.

A self-organizing map is a special kind of neural network that combines regression, projection and clustering (Kohonen, 1997). A SOM uses unsupervised learning rather than the more common supervised learning common to neural networks. The structure and algorithm of SOM and its important properties will be introduced in the next chapter.

Chapter 3

SELF-ORGANIZING MAPS

The Self-organizing map is a data visualization technique invented by Kohonen (1982). SOM is a non-parametric regression technique that forms a two-dimensional (or three-dimensional) topological representation of higher-dimensional input data. The neurons of SOM present the general form of the data. The SOM is trained through successive presentation of sample data vectors, producing an “elastic net” that stretches to cover the input space of the data. Instead of looking at extensive tabular data or statistics, we use a map that allows us to visualize an abstraction of the original data.

3.1 Structure

SOM is a special kind of neural network, consisting of two layers - the input layer and the output layer as illustrated in Figure 4 (Dutta, 1993). The layers consist of data processing units called neurons. All output neurons are connected to all input neurons with scalar weights and connections represented by lines.

The input layer has exactly as many neurons as the input vector. Each input vector is n -dimensional representing the attributes of the data. These elements are usually numerical but may be non-numerical.

Figure 4. Structure of Self-organizing Maps (source: Dutta, 1993).

For this thesis, the output layer consists of neurons organized on a two-dimensional grid although other configurations are possible. The number of neurons (from a few dozen to several thousand) and their order are user-defined, depending on the application. Each neuron is called a n -dimensional prototype, weight, or codebook vector, where n is equal to the dimension of the input vector. Input vectors are compared to these prototype vectors.

The topology or structure of the map is dictated by neighbourhood relations that connect neurons to other neurons. Immediate neighbours refer to adjacent neurons i.e. belonging to the 1-neighbourhood $N_{i,1}$ of the neuron i . In the 2-dimensional case, the neurons of the map can be arranged on either a rectangular or a hexagonal lattice as displayed in Figure 5 (the innermost polygon corresponds to 1-neighbourhood, the second to the 2-neighbourhood, and the biggest to the 3-neighbourhood). However, hexagonal is suggested because of its effective visual display (Kohonen, 1997).

Figure 5. Two examples of topological neighbourhood with size 1 to 3 (source: Vesanto et al, 2000).

3.2 Basic Algorithm

Let $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ represent an n -dimensional sample *input* data vector, where t is the index of the sample vector space ($t = 1, 2, \dots$).

The algorithm determines the values of prototype vectors that form n -dimensional neurons

$$m_i(t) = [m_{i1}(t), m_{i2}(t), \dots, m_{in}(t)]$$

where i denotes the index of the current neuron in the *output* map and t is as shown above.

Placement of neurons in the output map is fixed, so neuron i does not change position.

The basic SOM algorithm is iterative as follows:

Setp 1. Initialize network

For each neuron i , the prototype vector $m_i = [m_{i1}, m_{i2}, \dots, m_{in}]$ is generated, where n is the dimension of the input data vector.

Prototype vectors are generated in one of 3 ways:

- **Random initialization** uses random initial values to initialize the prototype vectors. This is used if little is known about the input data but may not be the best way to train the map.
- **Linear initialization** uses principal component analysis (PCA) of the input data. It is generally more efficient for training the map.
- **Sample initialization** sets the prototype vector to a randomly chosen from the input data vector. Thus the points automatically lie in the same part of the input space as the data.

Step 2. Present input vectors

Present the input vector $x(t)$ at time t ($0 < t < s$, where s is the number of iterations defined by the user) to all nodes in the network simultaneously. The input vector may be chosen at random or cyclically from the training dataset.

Step 3. Calculate winning node

Find the winning node c , i.e. the one with smallest distance between the prototype vector and the input vector, so

$$\|x(t) - m_c(t)\| = \min_i \{ \|x(t) - m_i(t)\| \}$$

where $\|\cdot\|$ denotes the distance measurement (using Euclidean distance measure).

The winning node is referred to as the Best-Matching Unit (BMU), indicating the node whose prototype vector has the greatest similarity with the input vector.

Step 4. Update prototype vectors

After finding the BMU, the prototype vectors of the SOM are updated. The BMU and its topological neighbours are moved closer to the input vector in the input space as illustrated in Figure 6, where the solid and dashed lines correspond to the situation before (solid) and after (dashed) updating, respectively.

Figure 6. Updating the best matching unit (BMU) and its neighbours
(source: Vesanto et al, 2000)

The update rule is:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)].$$

where $h_{ci}(t)$ is the neighbourhood kernel around the BMU c at time t . It is a decreasing function of time t and of the distance of unit i , from the BMU c . The kernel (h_{ci}) consists of two parts: the neighbourhood function $h(d, t)$ and the learning rate function $\alpha(t)$ where:

$$h_{ci}(t) = h(\|r_c - r_i\|, t)\alpha(t)$$

where r_i denotes the location of unit i on the map grid.

The learning-rate $\alpha(t)$ ($0 < \alpha(t) < 1$) is also a decreasing function of time, usually a linear function or an inverse-time function. The inverse-time function has the form

$$\alpha(t) = \frac{A}{t + B}$$

where A and B are some suitably selected constants.

In the learning process, prototype vectors that are topographically close in the output array (up to a certain geometric distance) activate each other to learn from the input data. Hence all input vectors have approximately equal influence on the training result (Mulier and Cherkassky, 1994).

The neighbourhood function often has a Gaussian form:

$$h(d, t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right)$$

where $\sigma(t)$ is the neighbourhood radius at time t , and $\|r_c - r_i\|^2$ is the distance between map units c and i on the map grid. The Gaussian form is computationally intensive but ensures a best ordering of the map, since the quantization error (3.4.1) arrives at a global minimum instead of a local minimum (Kohonen, 1997).

Step 5. Present next input vector

Decrease h_{ci} so that $h_{ci}(t+1) < h_{ci}(t)$. Reduce the neighbourhood radius so $\sigma_i(t+1) < \sigma_i(t)$ for all nodes i and decrease the learning rate $\alpha(t)$. Repeat from step 2, choosing a new unique input vector $x(t+1)$ until all iterations have been made.

3.3 SOM Visualization

Visualization is used to synthesize large amounts of detailed information. As Fufte, (1983) indicates, visualization of SOM can be divided into three categories: visualization of cluster structure and shape, visualization of components, and visualization of data on the map.

3.3.1 Cluster Structure and Shape

We begin with getting an idea about the overall shape and possible cluster structure of the dataset. Projection and distance matrices techniques are commonly used to determine what and where the clusters are and their shape.

Projection methods. Projection approaches include linear or non-linear projections. The most often used methods are principal component analysis (PCA) (Bishop, 1995), multidimensional scaling (MDS) (Ripley, 1996; Kaski, 1997), Sammon's mapping (Sammon, 1969), and Curvilinear Component Analysis (CCA) (Demartines and Herault, 1997). The linear method of PCA is computationally easier than iterative nonlinear methods such as Sammon's mapping, MDS and CCA. Sammon's mapping and CCA emphasize local distances over global ones and thus are particularly good for showing the local shape of the dataset.

Distance matrices. A distance matrix is a matrix of distances between neighbouring map units. It may consist of all distances between map units and their immediate neighbours, as in the U-matrix (Ultsch and Siemon, 1990), or just a single value for each map unit, for example the median of the distances to the neighbours (Kraaijveld et al, 1995). Some distance matrix techniques are represented in Figure 7.

7(a) U-matrix 7(b) Distance Matrix 7(c) Similarity colouring

Figure 7. The representation of distance matrix techniques
(Source: constructed from SOM Toolbox demo).

Figure 7(a) shows U-matrix visualization. It is the most commonly used method to display distances between prototype vectors of neighbouring map units. The U-matrix can be visualized using grayscale (Iivarinen et al, 1994), where a dark colouring between neurons corresponds to a large distance and thus a gap between the codebook values in the input space, and a light colouring between the neurons signifies that the codebook vectors are close to each other in the input space. Light areas can be thought of as clusters and dark areas as cluster separators. Figure 7(b) displays the size of each map unit proportional to its average distance to its neighbours (the size of each unit is between 0 and 1). It gives essentially the same information as Figure 7(a). Figure 7(c) shows similarity colouring visualization where similar colours indicate similar map units (Himberg, 2000; Kaski et al, 2000). The colour map can also visualize clusters or link different visualizations together (Xu et al, 1998).

Without prior information about the clusters, we can use various distance matrices to visualize of cluster structure and shape in the input data. Training a SOM and representing it by a distance matrix, especially a U-matrix, provides quick insight into the data.

3.3.2 Components of the Map

While a U-matrix visualization provides overall cluster characteristics of the data, the component planes analyze the prototype vectors more closely, looking at values of the variables and what values or value combinations are typical for different clusters.

Figure 8. U-matrix and its component planes
(source: constructed from SOM Toolbox demo).

Figure 8 is an example of a U-matrix and its component planes visualization. Each component plane is a slice of the map units, consisting of the values of a particular vector component across all map units. Examination of a component plane indicates the range of values of that component and the values of the variables in each cluster. One can detect

correlation between variables by comparing component planes (correlations are revealed as similar patterns in identical positions of the component planes).

3.4 Map Quality

The SOM allows us to visualize an abstraction of the original data. To assess whether the map has properly adapted itself to the training data, several map quality measures have been proposed (Kraaijveld et al., 1995). Generally, SOM quality is evaluated based on mapping precision and topology preservation.

3.4.1 Mapping Precision

Mapping precision refers to how accurately the neurons 'respond' to the given dataset. If the reference vector m_i of the BMU, calculated for a given testing vector x_i , is exactly the same as x_i , the error in precision is zero. Normally, the number of data vectors exceeds the number of neurons resulting in nonzero precision error.

Data histograms (hit matrices) help to display how input data vectors are clustered by the SOM, and the number of vectors belonging to a cluster are defined by each neuron. A data histogram is generated using the trained SOM and corresponding dataset. A hit counter is associated with each BMU; the counter represents the number of input data vectors associated with that BMU. There are several possible ways of visualizing the resulting histograms, one of which is shown in Figure 9(a).

A common measure of mapping precision is the average quantization error over the entire dataset:

$$E_q = \frac{1}{N} \sum_{i=1}^N \|x_i - m_c\|$$

where N is the total number of units on the map.

Figure 9. Visualization of data on the map
(source: constructed from SOM Toolbox demo).

Figure 9(b) is a quantization error (QE) plot, where the vertical line shows the match accuracy. The height is proportional to the ratio between quantization errors $q_i = \|x - m_i\|$ for the best-matching and worst-matching prototype vectors (i.e. $\min_i q_i / \max_i q_i$). The QE plot can be used to evaluate how accurately the map represents the data.

3.4.2 Topology Preservation

The topology preservation measure describes how well the SOM preserves the topology of the dataset. Kiviluoto (1995) calculated topographic error as

$$E_t = \frac{1}{N} \sum_{k=1}^N u(x_k),$$

where $u(x_k) = 1$ if the first and second BMUs of x_k are not adjacent units, zero otherwise. E_t thus represents the proportion of sample vectors for which two best matching weight vectors are not adjacent units:

3.5 SOM Clustering

Clustering looks for “interesting” groups in the data (Cherkassky and Mulier, 1998). It does not require that a specific size or number of groupings be found. No a priori knowledge of group membership or possible number of natural clusters is needed. In cluster analysis, data are grouped based on similarities of their observed variables.

SOM is a clustering tool that can be used to detect hidden structures in a massive dataset. Clustering can be done in two stages using the SOM as an intermediate step. As illustrated in Figure 10 the data is first clustered using the SOM, and then the SOM is clustered either by using existing cluster algorithms or by manual selection based on the visualization.

Figure 10. Two-level approach of SOM clustering

(Source: Vesanto, 2000).

When using SOM as a clustering method, a common strategy is to calculate a distance matrix for the prototype vectors, and use a high value of the matrix (i.e. a large distance) as an indication of a cluster border (Van Gils, 1995; Kraaijveld et al, 1995; Ultsch and Siemon, 1990). Other approaches use as k-means, agglomerative and divisive algorithms as discussed in Section 2.2.1..

Chapter 4

PATTERN RECOGNITION OF MUTUAL FUNDS

The knowledge discovery process, together with general data mining techniques, and specifically SOM clustering and its visualization have been discussed in the previous chapters. In this chapter, we concentrate on how to use the knowledge discovery process and the SOM algorithm to find potential hidden patterns in mutual fund data. We explore two kinds of SOMs - a “small SOM” for pattern recognition and a ‘big SOM’ for investment selection. We also extend the capability of the SOM Toolbox in Matlab by implementing an algorithm for explicit clustering that more closely corresponds to the original U-matrix visualization than do existing algorithms. A post-cluster summary with corresponding boxplot is also provided for interpretation purposes.

4.1 Data

The dataset for this thesis is extracted from HTML files that originated from Kiplinger’s Mutual Fund Finder (<http://www.kiplinger.com/php/fundfinder2001/fundfinder2001.html>). The extracting program employed is available in Appendix K. In total, nearly 4000 mutual funds were retrieved at the close of the last trading day of May 2002.

There are 18 variables in each fund: Fund Name, Total Return (including 1-year, 3-year, 5-year return), Universe, Category, Investment Style, Decile Rank, Down Market, Volatility, Turnover, Assets, Manager Since, Minimum Investment, Maximum Load, Expense Ratio,

Yield, Average Maturity, and Bond Rating. A detailed explanation about how to use the fund data, reproduced from <http://www.kiplinger.com/test/glossary01.htm> is in Appendix A.

4.2 Basic Data Analysis

The basic data analysis involves the first two steps of the knowledge discovery process: data selection and data pre-processing.

4.2.1 Data Selection

The mutual funds in this dataset can be categorized into four basic groups: U.S. Stocks, International Stocks, Taxable Bonds, and Municipal Bonds (tax-free). Thus the entire dataset is divided into four sub-datasets. The resulting partitioning of the entire dataset of 3965 mutual funds is shown in Figure 11:

Figure 11. Partitioning of mutual funds

In this thesis, we will use only Taxable Bonds and U. S. Stocks for pattern recognition, clustering visualization, and investment selection, using SOM techniques.

4.2.2 Data Pre-processing

Data pre-processing and cleaning is a time-consuming but extremely important step in the data mining process. The pre-processing involves statistical summaries, boxplots of variables, dealing with missing data and extreme values, correlation hunting, and variable normalization.

4.2.2.1 Statistical Summary

A statistical summary is a descriptive data analysis that provides a general idea of the data distribution. The summary statistics include mean, median, standard deviation, minimum, maximum, count, and the count of “not available” or “missing value”. The detailed statistical summaries for both Bond funds and Stock funds appear in Table 1 and Table 2 respectively.

Table 1. Summary Statistics per variable (Bond funds).

Variable	I_Y return	3_Y return	5_Y return	DR02	DR01	DR00	DR99	D_Market
Mean	5.37%	5.13%	5.54%	5	5	5	5	-1.94%
Median	6.40%	6.20%	6.20%	5	5	5	5	-2.20%
Minimum	-23.70%	-20.60%	-6.80%	1	1	1	1	-19.20%
Maximum	17.40%	18.00%	12.60%	10	10	10	10	20.90%
Stdev	0.04	0.04	0.02	2.79	2.79	2.77	2.88	0.03
Count	749	711	622	738	711	672	623	698
# of NA	2	40	129	13	40	79	128	53

Variable	Volatility	Assets (million)	Manager Since	Min_invest	Max_load	E_ratio	Yield	Ave_mt
Mean	5	\$412	1996	\$3,121	4.00%	0.98%	5.90%	8
Median	5	\$81	1997	\$1,000	4.50%	0.95%	5.30%	7
Minimum	1	\$0	1935	\$1	1.00%	0.02%	0.20%	0
Maximum	10	\$17,234	2002	\$50,000	6.25%	2.72%	17.50%	30
Stdev	2.86	1204	4.71	5,907	0.01	0.003	0.02	5.06
Count	751	750	751	730	422	750	747	607
# of NA	0	1	0	21	329	1	4	144

Table 2. Summary Statistics per variable (Stock funds).

Variable	1_Y return	3_Y return	5_Y return	DR02	DR01	DR00	DR99	D_Market
Mean	-9.53%	0.94%	6.44%	5	5	5	5	-19.70%
Median	-10.60%	0.10%	6.30%	5	5	5	5	-20.30%
Minimum	-65.10%	-50.20%	-33.20%	1	1	1	1	-94.90%
Maximum	138.50%	94.30%	35.90%	10	10	10	10	283.10%
Stdev	0.18	0.1	0.06	2.91	2.9	2.91	2.93	0.28
Count	2435	2106	1614	2312	2113	1897	1621	2278
# of NA	4	333	825	127	326	542	818	161

Variable	Volatility	Turnover	Assets (million)	Manager	Min_inves	Max_load	E_ratio
Mean	6	124.08%	809.86	1997	2573	4.93%	1.34%
Median	6	75.85%	92.15	1998	1000	5.50%	1.29%
Min	1	0.30%	0.1	1931	1	0.20%	0.06%
Max	10	7338.20%	72473	2002	50000	8.50%	14.70%
Stdev	2.87	2.5	3478.6	4.93	4422	0.01	0.01
Count	2439	2426	2426	2439	2275	1332	2428
# of NA	0	13	13	0	164	1107	11

A scatterplot matrix with histograms of eight important variables on the diagonal is in Appendix C. A set of all variable histograms can be found in Appendix D.

4.2.2.2 Missing Data

The summary statistics reveal that most of the variables contain a substantial amount of missing values. That is often the case with financial data. For the same fund, there often exists a phenomenon of sequential missing data. For instance when the three-year return column is missing, the five-year return column may be missing as well. This is often because a fund has not existed that long.

SOM Toolbox (Vesanto et al, 2000) deals with missing values by assuming the missing values have the same values as the values present in prototype vectors. Therefore, it uses only known values for distance calculations when searching for the best matching unit. However, this could produce an unrealistic mapping.

In order to get an objective result, we delete the incomplete observations. Although this may lead to losing information on ‘young’ funds, we believe it preferable to making assumptions about the value of the missing observations.

4.2.2.3 Extreme Values

For purposes of this thesis, an extreme value has been defined as a value that is more than three (3) standard deviations from the mean. Boxplots of our data show that some variables contain local extreme values. We have checked these values and cannot find any evidence that the extreme values present do not represent accurate values (for examples, Assets are for individual share classes, but some funds are several times larger than other funds). While these may be outliers there is no reason to discard them as inaccurate.

4.2.2.4 Normalization of the Data

Scaling of variables is important. The SOM algorithm uses a Euclidean metric to measure distances between vectors so the variables with larger ranges will almost completely dominate the map organization because of the distance calculation. To make each variable equally important (a good first choice in a preliminary clustering test) and to improve the map quality, all training datasets have been normalized. The normalization of input variables has been done by centering and scaling according to the following formula:

$$x' = (x - \bar{x}) / s$$

where $\bar{x}$ is the arithmetic mean of the variable x and s is its standard deviation.

4.2.2.5 Correlation Hunting

Pairwise (Pearson) correlation analysis is suggestive of some correlation among variables. The detailed pairwise Pearson correlation of the dataset can be found in Appendix B. Important variables such as 1-year, 3-year, and 5-year returns, while somewhat correlated, are retained in the hunt for cluster structure.

4.3 Training SOM Maps

Unlike most neural networks that are trained using supervised learning, a self-organizing map is trained using unsupervised learning. During the training process, the network has no knowledge of the desired output. The training process involves a competition between the output nodes. The output node with a reference vector closest to the input vector is the BMU. The reference vector of the BMU is moved in the direction of the input vector, as are the reference vectors of the surrounding nodes in the output array. The amount of adjustment of the neighbouring reference vectors depends on the distance of those nodes from the BMU in the output array. In this way, output nodes begin to represent features in the input vector and form clusters.

The software “SOM Toolbox” for MatLab 5 (Vesanto et al, 2000) is used in this stage. It is available free from <http://www.cis.hut.fi/projects/somtoolbox>. The package was developed by the SOM Toolbox Team at the Helsinki University of Technology based on the Kohonen self-organizing map algorithm.

4.3.1 Determine Training Parameters

The resulting map can be affected by map shape, the number of neurons chosen, the number of training steps, the learning rate, and the neighbourhood radius (Kohonen, 2001) so these parameters must be carefully determined.

As mentioned in previous sections, a lattice type of array can be defined as hexagonal or rectangular. Hexagonal is visually attractive and has the advantage of equidistant neighbours so we have chosen it for pattern recognition of mutual funds. For a stable orientation in the data space (Kohonen et al, 1996), recommended a rectangular shape rather than a square shape.

Another recommendation is that the training length of the second (fine tuning) phase should be at least 500 times the number of neurons (Kohonen, 1997). The training length must be fixed prior to training the SOM because the rate of convergence for the neighbourhood function and the learning rate are calculated accordingly.

The choice of learning rate depends on the SOM algorithm. If a batch algorithm is used to train the SOM, the learning rate is ignored; if the sequential SOM algorithm is used, the learning rate in the first phase is taken to be large (0.5 by default) and decreased in the fine-tuning phase (0.05 by default).

The neighbourhood radius is taken larger at the start (about $\frac{1}{4}$ of map size) and is decreased linearly to one (1) during the training.

4.3.2 Creating Suitable Maps

SOM can be trained as a ‘small map’ (the number of neurons is smaller than the input vector) or a ‘big map’ (the number of neurons is greater than the input vector). In general, big maps produce more detailed information since input vectors are spread out on nodes larger than the input vectors. Small maps can contain big clusters, or more input vectors can cluster on a small set of nodes. The choice of small or big map depends on the use of the map. To study clustering using SOM, it is recommended (Kaski, 2000), that the number of neurons should not exceed 10% of the total number of inputs. If one is trying to visualize more detailed information through the map, then a large map may need to be considered. In this thesis, we use both maps -- the small one for pattern recognition and the big one for investment selection.

Where prior knowledge of cluster structure exists in the form of a label, one can assess the trained map by displaying that cluster label for each map unit and thus visually determine whether the found cluster structure matches prior knowledge. One can also incorporate a priori information about the data by training a SOM by using sample initialization rather than random initialization.

During the training process, many SOMs have been created. As every SOM is different, even for the same dataset, choosing a ‘good’ map is problematic. Average quantization error and topographic error are generally used as criteria for choosing an optimal map. The average quantization error serves as an indicator of the accuracy of data representation. Generally the larger the map unit, the less the quantization error. Also using a larger neighbourhood radius

results in a greater quantization error since more neighbouring neurons are likely to be adjusted.

4.4 Clustering and Visualization

The SOM technique is a form of neural network that can be used for clustering and visualization. From the input data a two-dimensional map resembling a landscape is created. Borders define different clusters where a cluster consists of cases where the input variables have similar characteristics.

As discussed in Section 3.3.1, the U-matrix technique is widely used for visualizing cluster structure. Training a SOM and representing it with a U-matrix offers a quick insight into the data structure. Therefore our pattern recognition for mutual fund is mainly based on U-matrix visualization. However, determining a good clustering is a non-trivial task and there exist some problems visualizing clusters with the U-matrix. When a map is created, clusters are selected manually based on the visualization. This also means the number of clusters may be chosen subjectively according to the user's preference. The process is tedious and nothing guarantees that the manual selection is done consistently. Also there is no way to know which funds are included in a particular cluster. So an automatic method is needed in practice.

4.4.1 Improving Explicit Clustering

The current SOM Toolbox provides a function to do explicit clustering using existing algorithms such as hierarchical or non-hierarchical clustering based on local minima of the U-matrix. However, all output maps that are associated with existing clustering algorithms differ from their original U-matrix visualization. Maps resulting from explicit clustering using the existing algorithms are displayed in Figure 12 and Figure 13.

Figure 12. U-matrix and its explicit clustering based on the existing clustering algorithms of SOM Toolbox focused on Bond funds.

Figure 13. U-matrix and its explicit clustering based on the existing clustering algorithms of SOM Toolbox focused on Stock funds.

Figure 12(1) is the original U-matrix visualization for Bond funds; it assigns similar colours to similar map neurons. The rest are three explicit clustering maps, based on the result of Figure 12(1), using the centroid, single linkage, and complete linkage clustering algorithms. All three maps indicate four clusters, where the same cluster is identified with the same colour in each map, but not between maps - all maps are linked by the location of their neurons. Figure 13 is the U-matrix visualization and explicit clustering focused on the total of 1478 Stock funds. A simple glance at these maps shows none matches well with the corresponding U-matrix

visualization. Further examination of components also verifies that there are no clear differences among clusters. Table 3 and Table 4 display counts and averages for variables associated with Bond funds and Stock funds respectively, based on centroid clustering.

Table 3. Counts and averages for Bond Funds using Centroid

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Yield (%)
C1	107	7.5	7.0	7.3	-4.9	\$149.4	1994	0.81	5.3
C2	416	6.1	6.1	6.0	-1.6	\$586.1	1995	0.96	5.4
C3	36	4.7	5.2	5.5	-0.8	\$1037.2	1991	0.96	6.1
C4	45	7.6	6.9	6.9	-3.7	\$83.5	1995	0.9	5.2

Table 4. Counts and averages for Stock Funds using Centroid

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Turnover (%)
C1	206	-6.0	2.3	8.2	-7.1	\$4911.0	1988	0.96	55
C2	72	-9.1	-1.5	5.6	-18.0	\$509.8	1995	0.64	30
C3	174	-5.0	2.4	7.8	-4.7	\$908.8	1988	0.94	48
C4	350	-14.0	-3.6	4.8	-25.0	\$408.2	1993	1.2	54
C5	225	-18.0	-6.3	3.5	-34.0	\$361.8	1995	1.3	61
C6	256	-15.0	-4.4	4.5	-27.0	\$378.3	1993	1.2	53
C7	195	-14.0	-4.1	4.4	-26.0	\$334.1	1993	1.3	56

Table 3 shows the performance of 1-year, 3-year and 5-year returns vary little among clusters for Bond funds. It also shows the value of Down Market, Expense Ratio and differ little. Other algorithms, such as single linkage and complete linkage as in Appendix H, also produce similar results. The same conclusion can be drawn for Stock funds (see Table 4). Thus existing explicit clustering based on the SOM Toolbox does not conform with the U-matrix and does not appear to produce verifiable clusters.

As a result a new algorithm is proposed to produce clustering that better matches the original U-matrix visualization. The new method is developed based on the U-matrix map result. The algorithm is as follows:

1. Each node is compared with its nearby neighbours.
2. If no neighbour is classified then this node will be assigned to a new cluster.
3. This node is assigned to the same cluster as the neighbour node that has the smallest difference with the node if the relative difference of the node to the average value of the cluster is within a given criterion (which determines the fineness of the clustering level).
4. The node will be assigned to a new cluster if the above condition is not met.

The detailed implementation of the above algorithm can be found in Appendix J. The discovered structures based on this improved SOM method are discussed in Section 4.5.1.2 and Section 4.5.2.2. The resulting maps in Figure 16(2) and Figure 21(2) visually demonstrate a significant improvement over the maps using existing algorithms. In 4.5.1.3 and 4.5.2.3, we quantify the difference between clusters for the existing and proposed algorithms that provide explicit clustering using SOM.

In order to examine the flexibility of the above algorithm, we also use other values, such as minimum rather than average value of a cluster in step 3, to do the clustering. The comparison of the results (Appendix I) shows that the use of minimum value of a cluster in step 3 also produces a reasonable clustering, although the use of average appears better for our application.

4.4.2 Post Clustering Summary and Boxplot

After doing explicit clustering, we wish to have an automatic post-clustering summary, especially when the node contains many original data elements (it is impossible to do a

statistical summary by hand in a small map). This provides the characteristics of each cluster and indicates which component(s) define(s) the cluster.

Most software implementations of the self-organizing map do not have functionality for a post-clustering summary, except the commercial software package of Viscovery SOMine (<http://www.eudaptics.com/technology/somine4.html>) so this feature has been also added into the SOM Toolbox as well. The post-cluster summary includes mean, median, minimum, maximum and standard deviation for each cluster. A corresponding boxplot of each variable for different clusters is also produced for visualization purposes.

4.5 Result and Interpretation

Two kinds of SOMs are explored and interpreted in this section. One is a small map that is used to detect mutual fund structure for both Bond and Stock funds. Another is a big map that is used to support choosing mutual funds.

4.5.1 Bond Funds Pattern Recognition

The most important eight variables associated with Bond funds were used for SOM clustering; these are 1-year return, 3-year return, 5-year return, Down Market, Assets, Manager Since, Expense Ratio, and Yield. A total of 604 Bond funds were used for training the map.

The selection of significant variables was based on the histogram distribution, correlation between variables, as well as basic knowledge of mutual fund investments. These eight variables for Bond funds were divided into four classes according to their characteristics: performance, fund size, management, and expenses.

A SOM was trained using the batch training algorithm with linear initialization. The batch algorithm is a version of the SOM algorithm where the learning rate α is not used (Kohonen, 1997). Instead of using a single data vector at a time, the whole dataset is presented to the map before any adjustments are made—hence the name ‘batch’.

The maps were trained in two phases: a rough training with large initial neighbourhood width ($\frac{1}{4}$ of map size) and a fine-tuning phase with small initial neighbourhood width (reducing to 1 at the final stage). The purpose of the first training phase was to order the randomly initialized vectors of the map to approximately correct values, so that the prototype vectors resemble more closely the original dataset. During the second phase, the map was ‘fine-turned’. The neighbourhood function used was the Gaussian function.

Several self-organizing maps were trained with different training lengths as shown in Table 5. It can be seen that the average quantization error and topographic error of the SOM differ very little for different training length pairs, but the elapsed time is very different. For example, the elapsed time in the pair of 500/50000 epochs (iterative steps) took 1.21 hours while the pair of 10/40 epochs only took 5.3 seconds.

Of all these SOMs, the one with the smallest quantization error was picked. This was the SOM with 20 epochs in the first phase and 80 epochs in the second phase.

Table 5. Errors of different training length (for Bond funds)

First Phase	Second Phase	Quantization Error	Topographic Error	Training elapsed time
10 epochs	40 epochs	1.176	0.029	5.3 s
20 epochs	80 epochs	1.173	0.028	7.3 s
200 epochs	800 epochs	1.175	0.035	1.21 min
500 epochs	50000 epochs	1.174	0.034	1.21 h

4.5.1.1 Distance Matrix Visualization

Visualization takes distance matrices into account, especially the U-matrix. The U-matrix and its component maps are shown in Figure 14, which shows the following information:

1. There are about four or five clusters in the map.
2. There exists correlation between three components: 1-year return, 3-year return and 5-year return.
3. Most Bond funds (over 75%) have good short-term or long-term returns.
4. High yield and low return appear in the right bottom corner.
5. The largest Assets appear in the right upper corner and the largest Expense Ratio (E-ratio) is in the left bottom corner.

Figure 14. The U-matrix and component maps focused on Bond funds.

Alternatively, the distance matrix visualization can be examined by using the size of map units (see Figure 15(b)) or by using the similarity colouring of map units (see Figure 15(c)). The size of each unit in Figure 15(b) is between 0 and 1 with the smaller the size meaning the larger the distance to the neuron. The similarity colouring in Figure 15(c) is done by giving a colour to each map unit, so those similar map units get similar hues. All three different presentations in Figure 15 indicate existence of about four or five clusters.

Figure 15. Distance matrix techniques of SOM visualization that focused on Bond funds.

4.5.1.2 *Discovered Data Structure*

SOM is a popular tool for clustering and visualization. As the clustering is mainly based on human judgement it is not objective and is therefore hard to use for further data modeling. In addition, as we have discussed in Section 4.4 the existing explicit clustering of the SOM Toolbox is based on local minima of the U-matrix and the result differs substantially from the original U-matrix. A new clustering algorithm has been developed and implemented and shows a very promising result as displayed in Figure 16 (2).

Figure 16. Two explicit clusterings of SOM based on their U-matrix visualization (focused on Bond funds).

Figure 16(1) is the U-matrix visualization of mutual funds focused on Bond funds. Figure 16(3) is an explicit clustering map that is based on local minima of the original U-matrix. We can see the map in Figure 16(2), associated with the new clustering method, matches its original U-matrix better than the map in Figure 16(3).

Further analysis of the clusters found by the new explicit clustering method produces the following summary of statistical characteristics for the clusters. The summary is represented in tabular form (see Table 6) and graphically (see Figure 17) (only the mean value of variables for each cluster and the graph for one-year return of all clusters of Bond funds are shown here).

Table 6. Cluster post-descriptive summary for Bond funds

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Yield (%)
C1	481	6.5	6.1	6.2	-2.8	\$214.5	1995	0.96	4.9
C2	31	6.8	6.7	6.7	-2.3	\$4,476.7	1993	0.6	5.6
C3	21	7.1	8	4.7	4.8	\$605.2	1994	1.3	9.5
C4	11	1.6	3.6	4.3	-0.7	\$532.4	1991	1.1	7.2
C5	60	-2.1	-2.4	0.5	1.1	\$521.5	1997	1.2	9.5

A more detailed summary that includes mean, median, minimum, maximum and standard deviation for each cluster is in Appendix E. The corresponding boxplots can be found in Appendix F as well.

Figure 17. Boxplot of 1-year return for each cluster focused on Bond funds (thick lines represent the mean and standard deviation; horizontal thin lines represent the minimum, median, and maximum respectively. The number in upper right corner denotes cluster number and the number of funds in the cluster).

The results show that there are five major clusters, labeled from C1 to C5, for the 604 Bond funds in total. The symbols for specific funds contained in each cluster are in Appendix I. The detailed interpretations are as follows:

The **first cluster** (called “healthy and small size” funds) consists of 481 Bond funds with good performance in both the short-term and long-term (averaging 6.5% for 1-year return, 6.1% for 3-year return and 6.2% for 3-year return). The average Assets are about \$214.5 million and the average Yield is 4.9% (the lowest). It is down market 2.8% (on average) during the bear market (from January 29, 1999 through June 11, 1999).

The **second cluster** (called “well-performing big size” funds) consists of 31 Bond funds with lowest expense ratio. The short-term and long-term returns are very stable (average over 6.7%). The management has existed since 1993.

The **third cluster** (called “super-performing middle-size” funds) consists of 21 Bonds with the highest expense ratio and yield. It has the highest short-term and mid-term return (the average of 1-year return is 7.1%, and 3-year return is 8.0%). It is up market 4.8% (average) during the bear market.

The **fourth cluster** (called “under-performing middle-size” funds) has about 11 Bonds. The (average) 1-year return in this group is only 1.6%, its 3-year return is 3.6%, and its 5-year return is 4.3% with its management in play since 1991.

The **last cluster** (called “the poorest middle-size” funds) consists of 60 Bonds with negative returns in 1-year and 3-year. This group has very bad performance over five years with its management in play since 1997.

4.5.1.3 Comparisons of Clustering Results

In the previous section, we have shown visually that the discovered structures using our improved SOM have a better clustering effect than the one that is constructed using existing algorithms. The question here is whether this visualization reflects actual improvement when comparing the results from their post-clustering summary.

To justify that our proposed explicit clustering method has a better output than the existing one, multivariate analysis of variance (MANOVA) (Johnson and Wichern, 1998) is used

empirically. We apply MANOVA to analyze the results from the two clustering methods, examining variation between and within clusters.

The idea of MANOVA, like univariate analysis of variance (ANOVA), is aimed at testing the null hypothesis $H_0: \tau_1 = \tau_2 = \dots = \tau_g = \mathbf{0}$, which means the group effects (or clustering effects in our case) are identical (and hence can be taken as zero). Rejection of this hypothesis is generally accompanied by the conclusion that at least some of the groups or clusters are indeed different. We use the MANOVA test empirically to compare within and between cluster variation, with a stronger MANOVA test result indicating a better clustering.

A MANOVA model for comparing g population mean vectors is as follows:

$$X_{\ell j} = \mu + \tau_{\ell} + e_{\ell j}, \quad j = 1, 2, \dots, n_{\ell} \text{ and } \ell = 1, 2, \dots, g$$

where $X_{\ell j}$ are response vectors; μ denotes an overall population mean; τ_{ℓ} is the ℓ th cluster (group) effect satisfying $\sum_{\ell=1}^g n_{\ell} \tau_{\ell} = \mathbf{0}$; $e_{\ell j}$ are independent $N_p(\mathbf{0}, \Sigma)$ variables; ℓ represents the group or cluster, and j represents the observation number within the cluster.

For example, in the case of Stock Funds, $X_{\ell j}$ are response vectors where the vector is composed of the variables measured on the Stock funds; μ denotes an overall population mean vector corresponding to these variables; τ_{ℓ} is the ℓ th group or cluster effect satisfying

$$\sum_{\ell=1}^g n_{\ell} \tau_{\ell} = \mathbf{0}; \quad e_{\ell j} \text{ are independent } N_p(\mathbf{0}, \Sigma) \text{ variables; } \ell \text{ represents the Stock cluster, and } j$$

represents the observation number within the Stock cluster.

A sample observation can be decomposed into sample mean, estimated cluster (group) effect, and residual by the model:

$$\mathbf{x}_{\ell j} = \bar{\mathbf{x}} + (\bar{\mathbf{x}}_{\ell} - \bar{\mathbf{x}}) + (\mathbf{x}_{\ell j} - \bar{\mathbf{x}}_{\ell})$$

or
$$\mathbf{x}_{\ell j} - \bar{\mathbf{x}} = (\bar{\mathbf{x}}_{\ell} - \bar{\mathbf{x}}) + (\mathbf{x}_{\ell j} - \bar{\mathbf{x}}_{\ell})$$

where $\bar{\mathbf{x}}, \bar{\mathbf{x}}_{\ell}$ and $\mathbf{x}_{\ell j}$ represent the sample mean, the cluster mean and sample observations, respectively, and where $\ell = 1, \dots, g$ represents the group or cluster and $j = 1, \dots, n_{\ell}$ represents the observation number within the cluster.

After sum of squares and cross products (SSP), the above expression becomes:

$$\sum_{\ell=1}^g \sum_{j=1}^{n_{\ell}} \left(\mathbf{x}_{\ell j} - \bar{\mathbf{x}} \right) \left(\mathbf{x}_{\ell j} - \bar{\mathbf{x}} \right)' = \sum_{\ell=1}^g n_{\ell} \left(\bar{\mathbf{x}}_{\ell} - \bar{\mathbf{x}} \right) \left(\bar{\mathbf{x}}_{\ell} - \bar{\mathbf{x}} \right)' + \sum_{\ell=1}^g \sum_{j=1}^{n_{\ell}} \left(\mathbf{x}_{\ell j} - \bar{\mathbf{x}}_{\ell} \right) \left(\mathbf{x}_{\ell j} - \bar{\mathbf{x}}_{\ell} \right)'$$

The above equation can also be represented as

$$T = B + W,$$

where W denotes the within-group SSP, B denotes the between-group SSP and T denotes the total SSP.

To apply MANOVA to compare two clustering methods, three assumptions must be made:

1. All observations are independent.
2. The populations for each cluster or group are multivariate normal.

3. The variance-covariance matrix is the same for each population.

Most of the variables in Bonds and Stocks, are univariate normally distributed. For those not satisfying this constraint (Assets and Turnover), a base 10 logarithm transformation was applied to normalize the data. The variable Manage Since was taken as the number of years in charge, starting from 1931, instead of its calendar year. These actions resulted in near normality for the transformed data.

Wilks' Lambda (Wilks, 1932) is the most commonly used test statistic in MANOVA. It has the following form

$$\lambda^* = \frac{|W|}{|W + B|} = \frac{|W|}{|T|}$$

The determinant $|W|$ is called the generalized variance. As the within-group SSP (W) gets smaller relative to the total SSP (T), λ^* decreases, indicating rejection of the null hypothesis. In the case of large sample sizes, a modification of λ^* (Bartlett, 1938) is used to test the hypothesis of H_0 . Bartlett has shown that if H_0 is true and the total number of observations is large, then

$$-\left(n-1-\frac{(p+g)}{2}\right)\ln \lambda^* = -\left(n-1-\frac{(p+g)}{2}\right)\ln\left(\frac{|W|}{|W+B|}\right)$$

has approximately a chi-square distribution with $p(g-1)d.f.$ where p, g denotes the number of variables and the number of clusters (groups), respectively. Consequently, for large sample sizes, we reject H_0 at significance level α if

$$-\left(n-1-\frac{(p+g)}{2}\right)\ln\left(\frac{|W|}{|W+B|}\right) > \chi^2_{p(g-1)}(\alpha)$$

where $\chi^2_{p(g-1)}(\alpha)$ is the upper (100α) th percentile of a chi-square distribution with $p(g-1)$ degrees of freedom.

We apply MANOVA analysis to the data from Table 3 in Section 4.4.1 and Table 6 in Section 4.5.1.2, and get the following results as shown in Table 7 (more details can be found in Appendix G).

Table 7. MANOVA Analysis Comparison of Two Clustering Methods for Bond Funds

	Clustering of Centroid	Clustering of New Method
Number of Clusters	4	5
$ W $	1.0965e+007	8.1786e+006
$ B + W $	1.232e+008	1.232e+008
$\lambda^* = \frac{ W }{ B + W }$	0.0890	0.0644
Chi-square $\chi^2_{p(g-1), (0.01)}$	42.98	53.45
Bartlett's calculation $-(n-1-(p+g)/2) \ln \left(\frac{ W }{ B+W } \right)$	1444.8	1360.3

From the above table, we can see the value of Bartlett's calculation is greater than chi-square value for both methods. This indicates that both methods reject the null hypothesis at a significance level of 0.01. Furthermore, the value of $|W|$ using the centroid-clustering method

is greater than the value of $|W|$ that is calculated from our new method while the value of $|T|$ is the same for the two methods. So, we can conclude that the new method produces a stronger clustering result although both methods pass the significance test.

4.5.1.4 Quantization Error Analysis

Our criteria for the quality of an optimal map was based on the average quantization error, which is an average of the Euclidean distance of each input vector from its best matching reference vector in the map. We choose the optimal solution as that which minimizes average quantization error. A visual representation of the quantization error per neuron of the map is shown in Figure 18, where bigger neurons indicate larger quantization errors.

Figure 18. Quantization error per neuron of Bonds clusters

As noticed from Figure 18, large quantization errors appear at the edge of the map, showing that SOM has a border effect. The neighbourhood definition is not symmetric on the borders of the map. Therefore, the density estimation is different for the border units than for the center units of the map (Vesanto and Alhoniemi, 2000). In general, an optimal map shows low and equally distributed quantization error. Figure 18 satisfies this condition, suggesting that our previously constructed map is an optimal solution.

The quantization error of each cluster is also calculated and is given in Table 8.

Table 8. Cluster quantization error for Bond funds

Cluster	Quantization Error				
	Mean	Median	Minimum	Maximum	Std. Dev.
1	1.0809	0.9342	0.5993	2.6667	0.5093
2	2.0772	1.4751	1.3655	3.9931	1.2794
3	2.9631	2.9631	2.0451	3.881	1.2982
4	1.6765	1.6765	1.6765	1.6765	0
5	1.8466	1.7993	1.043	2.5215	0.5181

As shown in Table 8, the quantization error per neuron is between 0.5993 and 3.9931. The first cluster, containing the majority of Bond funds (79.6% of total Bond funds), has the smallest average quantization error (1.0809).

4.5.2 Stock Funds Pattern Recognition

The most significant eight variables for stock funds were used for SOM clustering; these are 1-year return, 3-year return, 5-year return, Down Market, Turn-over, Assets, Manager Since, Expense Ratio. A total of 1478 stocks were used for training the map.

The selection of significant variables was based on the histogram distribution, the correlation between variables, and a basic knowledge of mutual fund investments. The training process and parameters chosen are similar to that for Bond funds in Section 4.5.1.

4.5.2.1 Distance Matrix Visualization

The visualization and clustering follow the same steps as described in Section 4.5.1.1. The U-matrix and component maps for Stock funds are displayed in Figure 19. They show the following information:

1. There are at least four or five clusters in the map.
2. The highest short-term return appears in the upper right corner and the largest assets appear in the left upper corner.
3. There is no obvious correlation between variables.
4. The Turnover ratio is relatively low for most stock funds.

Figure 19. The U-matrix and component maps that are focused on the Stock funds using map size 13 by 10.

Figure 20. Distance matrix techniques of SOM visualization that focused on Stock funds.

The alternative different representation of distance matrices is in Figure 20. From these visualizations, it can be seen clusters of Stock funds will vary according to the user's preference.

4.5.2.3 Discovered Data Structure

By differing thresholds, we can have different clustering results. Again, by analyzing the output map more carefully, the one matching well the original U-matrix was chosen as the final map. The discovered data structure is displayed in Figure 21(2), where nine major clusters of Stock funds were identified. These clusters have been labeled from C1 to C9.

Figure 21. The explicit clusters of SOM (focused on Stock funds).

Further analysis based on the new method produces the following summary of statistical characteristics for all clusters. The summaries are represented in both tabular (see Table 9) and graphical (see Figure 22) forms (only the mean value of variables for each cluster and the graph for one-year return of all clusters of Stock funds are shown here).

Figure 22. Boxplot of 1-year return of clusters focused on Stock funds (thick lines represent the mean and standard deviation; horizontal thin lines represent the minimum, median, and maximum respectively. The number in upper right corner denotes cluster number and the number of funds in the cluster).

Table 9. Cluster Summary of Stock funds.

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-market(%)	Turnover (%)	Assets (million)	Manage since	E-ratio (%)
C1	24	-6.4	2.1	9.3	-7.9	41.6	\$24,505.3	1987	0.65
C2	382	4.2	9.9	10.5	5.4	74.1	\$635.6	1993	1.3
C3	15	92.6	24.2	0.5	-4.3	71.6	\$177.2	1996	2.1
C4	8	-9.2	2.4	8.1	-31.1	85	\$5,278.1	1988	0.65
C5	34	-13.1	-2.2	7.3	-24.7	57.3	\$6,587.3	1994	0.85
C6	909	-13.9	-2.5	5.3	-26	85.2	\$519.9	1997	1.2
C7	71	-20.5	-6	3	-38.9	387	\$409.1	1995	1.6
C8	27	-31.7	-16.9	-3.5	-63.2	170.8	\$173.6	1998	1.5
C9	8	-30.2	-22.2	-11.4	-59.5	182	\$5.8	1995	6.4

A more detailed summary that includes mean, median, minimum, maximum and standard deviation for each cluster is in Appendix E. The corresponding boxplots can also be found in Appendix F as well.

The results show that there are nine clusters in total for 1478 Stock funds, with the following details:

The **first cluster** consists of 24 very “large size” stock funds with average -6.4% for 1-year return, 2.1% for 3-year return and 9.3% for 5-year return. The average assets are about \$24,505.3 million and the average expense-ratio is 0.65% (the lowest when compared with other clusters). It has not performed very badly in the current market.

The **second cluster** is “well performing middle-sized” stock funds that consist of 382 funds. Compared with other groups, they have performed well in terms of short-term returns, and their middle and long-term returns are very high (9.9% for 3-year return and 10.5% for 5-year return). This group is up market an average of 5.4% during the bear market, (March 24, 2000 to April 4, 2001) when the S&P 500 returned -26.9% during that span.

The **third cluster** is “super-performing small-sized” funds, consisting of 15 funds with very high 1-year return (92.6%) and 3-year return (24.2%). Their assets average only \$177.2 million with average expense ratio of 2.1%.

The **fourth cluster** consists of 8 funds with very low expense-ratio (0.65%) and high 5-year return (8.1%). It is down on average 31.1% during the bear market.

The **fifth cluster** consists of 34 funds that were similar to, but performed slightly worse than, cluster four. It has low expense ratio (0.85%) and high 5-year return (7.3%).

The **sixth cluster** consists mainly of 909 small-sized funds having a negative 1-year return (-13.9%) and 3-year return (-2.5%). The average expenses ratio in this cluster is about 1.2% with the management in play since 1997.

The **seventh cluster** consists of 71 funds with the highest turnover ratio (387%). The performance in one and three years is not good. It is down on average 38.9% during the bear market.

The **eighth cluster** consists of 21 small sized funds with high turnover ratio (170.8%). The performance over one year and three years is very bad. It is down on average 63.2% during the bear market.

The **last cluster** consists of 8 funds with very small assets (5.8 million) and very high expense ratio (6.4%). Overall, it has the worst performance over three and five years and poor performance over one year.

4.5.2.3 Comparisons of Clustering Results

We carried out a similar MANOVA analysis as discussed in Section 4.5.1.3, using the result of Table 4 in Section 4.4.1 and Table 9 in Section 4.5.2.2 that focused on the Stock funds, and produced the following test results.

Table 10. MANOVA Analysis Comparison of Two Clustering Methods for Stock Funds

	Clustering of Centroid	Clustering of New Method
Number of Clusters	7	9
$ W $	4.03e+013	2.6795e+013
$ B + W $	3.5760e+014	3.5760e+014
$\lambda^* = \frac{ W }{ B + W }$	0.1127	0.0749
Chi-square $\chi^2_{p(g-1),(.01)}$	73.65	93.2
Bartlett's caculation $-(n-1-(p+g)/2)\ln\left(\frac{ W }{ B+W }\right)$	3208.0	3805.2

This table also shows that both methods reject the null hypothesis (that the cluster effects are all the same) at a significance level of 0.01 for Stock funds. While both clustering methods appear to identify cluster structure in the dataset, the value of $|W|$ using the centroid-clustering method is greater than the value of $|W|$ that is calculated from our new method while the value of $|T|$ is the same for the two methods. We conclude therefore that our new method produces a stronger clustering result than the existing algorithm (i.e. forming clusters that are more homogeneous within and more heterogeneous between).

4.5.2.4 Quantization Error Analysis

The previous best map was also chosen based on the minimum quantization error. The visual representation of the quantization error per neuron of Stock funds of the map is shown in Figure 23, where larger neurons indicate larger quantization errors. An optimal map shows low and equally distributed quantization error. Figure 23 also satisfies this condition, implying our previously constructed map is an optimal solution. The quantization error for each cluster of the Stock funds is calculated and is given in Table 11.

Figure 23. Quantization error per neurons of Stocks clusters.

Table 11. Cluster Quantization Error for Stock funds.

Cluster	Quantization Error				
	Mean	Median	Minimum	Maximum	Std
1	2.8199	2.5081	1.9206	4.0309	1.0892
2	1.1939	1.1019	0.7865	2.9541	0.4522
3	4.0399	4.0399	4.0399	4.0399	0
4	1.9097	1.9097	1.3998	2.4196	0.7211
5	1.1522	1.0517	0.7699	1.6349	0.4411
6	0.9009	0.8578	0.475	1.5627	0.2427
7	2.186	1.5152	1.0355	4.5811	1.2359
8	1.4772	1.4772	1.2243	1.7301	0.3577
9	6.9724	6.9724	6.9724	6.9724	0

As shown in Table 11, the quantization error per neuron is between 0.475 and 6.9724. The sixth cluster, containing the majority of Stock funds (61.5% of total Stock funds), has the smallest average quantization error (0.9009).

4.5.3 Mutual Funds Selection

Mutual funds fall into three broad groups: stock funds, bond funds and money-market funds. Stock funds deliver greater returns over the long term than bond funds. However, stock funds are less predictable and more likely to deliver sharp losses or gains over short time periods. Bond funds achieve much of their return by collecting interest on the securities they own. They are much safer, less volatile, and less rewarding when compared with stock funds. When

allocating money among stocks, bonds and cash, two important things need to be considered: financial goals and tolerance for risk and volatility.

When picking mutual funds, one can select them by category which reflects the mutual fund objective and risk volatility (examples of category are Aggressive Growth, Long-Term Growth, Growth & Income, Balanced, Diversified International, Corporate Bond), or by sector, which invests in a single industry or economic sector (examples of sector funds are Financial, Medical, Property, Natural Resources, Technology, and Utility).

In this section, we will use 'big SOM' to analyze three types of mutual funds that demonstrate the differences among seemingly similar funds and use this to help to improve fund selection as well as composition of portfolios.

4.5.3.1 *Natural Resource Stocks*

Natural resource stocks are invested in the natural resource sector. When selecting a mutual fund, it is essential to look at several criteria such as the performance of the fund (the return achieved over one, three or five years), the risk or volatility of the return via a benchmark, the size of the fund, the turnover rate, the expense ratio and the management of the fund.

Two SOMs of Stock funds focused on the Natural Resource sector are constructed and shown in Figure 24 and Figure 25. Figure 24 is the U-matrix and components map, which visualizes the following information:

1. There exist about four clusters.
2. There is fairly high correlation between Turnover and Minimum Investment.

3. Most funds have middle-term (3-year) and long-term (5-year) returns that are greater than 5.0%. The highest 3-year return is in the bottom right corner and the highest 5-year return is in the upper right corner of the map.
4. The largest Assets appear in the upper right corner and the largest Expense Ratio is in the bottom left corner.

Figure 25 adds labels onto the map, so it visualizes the fund location on the map and gives clustering information. There are four clusters as indicated on the map. The statistical summary as displayed in Table 12 for each cluster is as follows:

Cluster 1 consists of nine stocks that have a negative one-year return, but the highest and positive average Down Market (27.4%). This indicates the funds in this group have performed well during the bear market (March 24, 2000 to April 4, 2001), but have not performed well in the current economic market. However, their performance is good when traced over three to five years.

Cluster 2 contains two small-sized stocks that have a good one-year return (averaging 32.2%), but have been down market 8.65% on average during the bear market. This group has very high turnover rate (1711.00%) and asks for an average minimum investment of \$25,000. The success of their short-term performance is that the manager frequently trades stocks in the current volatile market.

error: 0.805, and final topographic error: 0.000.

Figure 25. O-matrix and labeled map of 49 industries focused on investment in Natural Resources sector.

Table 12. Analysis of mutual funds focused on Natural Resources sector

Cluster	No. funds	1_Y return	3_Y return	5_Y return	Down_market	Turnover	Assets (million)	Manager Since	Min_invest	Max_load	E_ratio
1	9	-5.98%	15.52%	7.33%	27.14%	69.41%	\$436.24	1994	\$2,389	1.08%	1.19%
2	2	32.20%	7.00%	-2.40%	-8.65%	1711.00%	\$51.35	1998	\$25,000	0.00%	1.35%
3	5	76.58%	12.58%	-8.06%	-8.58%	76.06%	\$47.70	1999	\$3,400	0.00%	3.31%
4	33	35.59%	17.15%	4.33%	2.96%	169.92%	\$126.12	1997	\$3,030	3.73%	1.64%

Cluster 3 contains five stocks that have super performance in their one-year return (76.58%), but they have a very bad record in five-year (-18.5%) and three-year (-8.06%) returns. Their expense ratio is the highest (3.31%) and no load fee is required. Although funds in this group have been down market an average 18.58% during the bear market, they have performed very well in the current market. The success of the short-term performance in this group may be due to management efforts.

Cluster 4 contains 33 funds and has performed well in its short-term (averaging 35.59%) and middle-term (averaging 17.15%). The performance of this group is very stable, but most of these funds charge a load fee.

4.5.3.2 Financial Stocks

A SOM of 18 mutual funds with three clusters that focused on the financial sector is shown in Figure 26. Cluster 1 contains 8 stocks, Cluster 2 contains 4 stocks and Cluster 3 contains 6 stocks. A statistical summary showing averages by variable for each cluster is provided in Table 13.

Table 13. Analysis of mutual funds focused on Financial Stocks

Cluster	No. funds	1_Y return	3_Y return	5_Y return	Down_Market	Volatility	Turn over	Assets (million)	Manager since	Min_invest	Max_load	E_ratio
1	8	1.85%	7.71%	12.88%	18.01%	7	74.98%	\$379.00	2000	\$2,313	2.31%	1.12%
2	4	-4.35%	2.78%	8.58%	9.10%	7	70.70%	\$526.20	1994	\$1,000	5.06%	1.24%
3	6	10.15%	10.62%	13.65%	21.83%	5	99.17%	\$212.40	1998	\$1,166	3.38%	1.83%

Figure 26. SOM of 18 mutual funds focused on Financial Stocks

Financial stocks in **Cluster 1** have performed well both in the middle-term (7.71%) and long-term (12.88%), with the lowest expense ratio (1.12%). Management has been in place, on average, since 2000. During the bear market, (March 24, 2000 to April 4, 2001), this cluster has performed well and is up market about 18% on average.

Cluster 2 has not performed well in the current market (-4.35% average for 1-year return) and even in the middle-term return (2.78% average for 3-year return), although the management has been in place on average since 1994. This cluster charges the highest load fee for purchasing shares. However, during the bear market, it is still up market on average 9.1%.

Cluster 3 has super performance over all five years, and it is up much more than other clusters during the bear market. This cluster has the smallest Assets but the highest expense ratio when compared with the other clusters. This is a recommended group for investment selection.

4.5.3.3 Aggressive Growth Stocks

Aggressive growth stocks invest in smaller companies or invest in those that take greater-than-normal risks. A total of 20 Aggressive-growth stocks that had load charges were selected. The sub-criterion we used was a 3-year return greater than 5.0%, an expense ratio less than 1.20%, and management in place before 1999.

The SOM shown in Figure 27 maps 20 Aggressive-growth stocks using map size 5 by 4. It shows that there exist three clusters among these seemingly similar funds. A summary showing averages by variable for these three clusters is provided in Table 14

Figure 27. SOM of 20 mutual funds focused on Aggressive-growth stocks that have front-load.

Table 14. Analysis of mutual funds focused on Aggressive Growth Stocks

Cluster	No. funds	1_Y return	3_Y return	5_Y return	Down-Market	Volatility	Turn over	Assets (million)	Manager	Min_invest	Max load	E_ratio
1	7	9.81%	13.29%	11.32%	-1.77%	4	4.68%	\$345.0	1991	\$1,321	4.04%	1.04%
2	9	-11.63%	9.19%	10.62%	-24.63%	7	124.91%	\$620.7	1993	\$1,178	5.03%	1.10%
3	4	1.38%	16.98%	19.35%	-13.63%	7	58.00%	\$1,805.3	1998	\$8,000	1.25%	0.84%

A detailed summary is as follows:

Cluster 1 is the best group and contains seven funds. The performances over short or long term are stable and their returns average above 9.8%. The average turnover ratio is only 4.68%. The volatility in this group is low when compared to other clusters. This is a recommended group for short or long term investment or for mutual fund portfolio composition.

Cluster 2 consists of 9 funds with the worst performance in the current stock market. The average 1-year return in this group is -11.63%, and it has been down market an average 24.63% during the bear market. However, it has performed well over three to five years. For a long term investment goal, this could be considered as a good group to purchase.

Cluster 3 is a group that contains four large-sized stocks with very good performance over three to five years. Their performance in the short-term is not very bad when comparing with other clusters. Their average Assets are about \$1,805.25 million and on average require a minimum \$8,000 investment. The expense ratio is the lowest (0.84%) but they have the maximum load charge (1.25% only).

Chapter 5

CONCLUSION

This chapter provides the conclusion of this thesis, the contribution it makes, and recommendations for future research.

5.1 Results

In this thesis, we extracted 3965 mutual funds measured on 18 variables from Internet sources and applied an unsupervised neural network approach (SOM) to visualize and compress multi-dimensional data into two-dimensional maps. We concentrate on pattern recognition for Bond funds and Stock funds and draw the following three conclusions:

1. SOM is an efficient clustering method that can detect patterns in massive data. It has flexible visualization capabilities to provide an informative picture of the data space as both ‘big SOM’ and ‘small SOM’ have illustrated in this thesis. With SOM, it is easy to recognize different patterns in massive mutual fund data and find attributes associated with those patterns.
2. There are clear clusters among Bond funds and Stock funds when using the improved method for ‘small SOM’ pattern recognition. Our results show that the improved SOM gives improved clustering results, both on visualization and in quantitative analysis, when using a complicated and massive dataset for pattern recognition in mutual funds

3. After reviewing the suitability of “big SOM” to serve as a decision support tool, we conclude that SOM gives a better basis, than traditional methods, for portfolio selection and performance comparison. For instance, SOM provides a data visualization map which traditional methods do not. Furthermore, we can get cluster information that traditional method are unable to offer.

5.2 Contribution

This thesis focuses on applying SOM and improving its explicit clustering in order to achieve better cluster visualization and quantitative analysis. We have developed a simple, effective clustering method that better matches the U-matrix visualization than existing explicit clustering algorithms. A post-clustering descriptive summary with corresponding boxplot is also provided to quantify characteristics that associate with the found clusters and to enhance interpretation and visualization. The comparative results for the improved SOM versus the existing SOM are encouraging.

5.3 Future Work

Some work is still needed regarding the post-processing stage and the interpretation of results. The discovered structures could be used as a ‘big SOM’ for further clustering analysis as discussed in Section 4.5.3. For ‘small SOM’, only taxable Bonds and U.S. Stocks were clustered and visualized; International Stocks and Municipal Bonds could be analyzed using the improved SOM proposed in this thesis. Also it would be possible to collect time series data from an Internet source to build mutual fund predictive models using the ideas in this thesis.

BIBLIOGRAPHY

- Ambroise, C., Seze, G., Badran, F., and Thiria, S., 2000. "Hierarchical Clustering of Self-organizing Maps for Cloud Classification". *Neuro Computing*, 30 (1), 47-52.
- Bartlett, M.S., 1938. "Further Aspects of the Theory of Multiple Regression". *Proceedings of the Cambridge Philosophical Society*, 34, 33-40.
- Becanovic, V., 2000. "Image Object Classification Using Saccadic Search, Spatio-Temporal Pattern Encoding and Self-organization". *Pattern Recognition Letters*, 21 (3), 253-263.
- Bendell, T., Boulter, L. and Kelly, J., 1998. *Benchmark for Competitive Advantage*. Pitman Publishing, London.
- Bishop, C. M., 1995. *Neural Network for Pattern Recognition*. Oxford University Press.
- Chen, D., Chang, R., and Huang, Y., 2000. "Breast Cancer Diagnosis Using Self-organizing Map for Sonography". *Ultrasound in Medicine and Biology*, 26, 405-411.
- Cherkassky, V. and Mulier, F., 1998. *Learning From Data: Concepts, Theory, and Methods*. John Wiley & Sons, Inc.
- Deboeck, G. and Kohonen, T., 1998. *Visual Explorations in Finance with Self-organizing Maps*. Springer-Verlag, p 250.

Bibliography

- Demartines, P. and Herault, J., 1997. "Curvilinear Component Analysis: A Self-organizing Neural Network for Nonlinear Mapping of Datasets". *IEEE Transaction on Neural Networks*, 8, 148-154.
- Dutta, S., 1993. *Knowledge Processing and Applied Artificial Intelligence*. Oxford Butterworth-Heinemann.
- Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R., 1996. *Advances in Knowledge Discovery and Data Mining*. Menlo Park, California, United States of America, American Association for Artificial Intelligence Press.
- Frawley, W., Piatetsky-Shapiro, G. and Matheus, C., 1992. "Knowledge Discovery in Databases: An Overview". *AI Magazine*, AAAI/MIT Press, 213-228.
- Fufte, E., 1983. *The Visual Display of Quantitative Information*. Graphics Press.
- Gedeon, T., Wong, P., Halgamuge, S., Kasabor, N., Nauck, D., and Fukushima, K., 1999. "Coloring that Reveals High-dimensional Structures in Data". *Proceedings of ICONIP'99, 6th International Conference on Neural Information Processing*, 2, 729-734, IEEE Service Center, Piscataway, NJ.
- Greene, W.H., 1997. *Econometric Analysis*, Third Edition. Upper Saddle River, New Jersey, USA, Printice-Hall.
- Himberg, J., 2000. "A SOM Based Cluster Visualization and Its Application for False Coloring". *Proceedings of International Joint Conference on Neural Networks (IJCNN2000)*, 3, 587-592.

Bibliography

- Iivarinen, J., Kohonen, T., Kangas, J., and Kaski, S., 1994. "Visualizing the Clusters on the Self-organizing Map". *Proc. of the Conf. on AI Research* (in Finland, Helsinki), 122 - 126.
- Johnson, R.A. and Wichern, D.W., 1998. *Applied Multivariate Statistical Analysis*. Fourth edition, Upper Saddle River, New Jersey, United States of America, Printice-Hall.
- Kangas, J.A., Kohonen, T.K., and Laaksonen, J.T., 1990. "Variants of Self-organizing Maps". *IEEE Transactions on Neural Networks*, 1(1), 93-99.
- Kaski, S., 1997. "Data Exploration Using Self-organizing Maps". Ph.D. thesis, Helsinki University of Technology.
- Kaski, S., Venna, J., and Kohonen, T., 2000. "Tips for Processing and Color-Coding of Self-organizing Maps". *Visual Explorations in Finance with Self-organizing Maps* (Deboeck and Kohonen), 195-202.
- Kiviluoto, K., 1995. "Topology Preservation in Self-organizing Maps". Technical Report A29, Helsinki University of Technology, Laboratory of Computer and Information Science.
- Kohonen, T., Hynninen, J., Kangas, J. and Laaksonen, J., 1996. SOM_PAK: The Self-organizing Map Program Package. Helsinki University of Technology.
- Kohonen, T., 2000. 5384 Works That Have Been Based on the Self-organizing Map (SOM) Method Developed by Kohonen. <http://www.cis.hut.fi/nnrc/refs/>.
- Kohonen, T., 1997. *Self-Organizing Maps*. Springer Verlag. 2nd edition, Heidelberg, Germany, Springer-Verlag.

Bibliography

- Kohonen, T., 2001. *Self-Organizing Maps*. Springer Verlag. 3rd edition, Heidelberg, Germany, Springer-Verlag.
- Koikkalainen, P. and Oja, E., 1990. "Self-organizing Hierarchical Feature Maps". *Proceedings of International Joint Conference on Neural Networks (IJCNN'90)*, 279-284.
- Koikkalainen, P., 1995. "Fast Deterministic Self-organizing Maps". *Proceedings of International Joint Conference on Neural Networks (ICANN'95)*, 63-68.
- Kraaijveld, M.A., Mao, J., and Jain, A.K., 1995. "A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps". *IEEE Transactions on Neural Networks*, 6 (3), 548-559.
- Moxon, B., 1996. "Defining Data Mining, the Hows and Whys of Data Mining, and How It Differs From Other Analytical Techniques", Online Edition of DBMS Data Warehouse Supplement. *IEEE Transactions on Neural Networks*, 11(3), 586-600.
- Mulier, F., and Cherkassky, V., 1994. "Learning Rate Schedules for Self-organizing Maps". *Proceedings of 12th International Conference on Pattern Recognition*, 224-228.
- Raivio, O., Riihijarvi, J., and Mahonen, P., 2000. "Classifying and Clustering the Internet Traffic by Kohonen Network". *Proceeding of the ICSS Symposia on Neural Computation (NC'2000)* May 23-26, 2000 in Berlin, Germany, ICSC Academic Press.
- Ripley, B.D., 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press.
- Sammon, J.W., 1969. "A Nonlinear Mapping for Data Structure Analysis". *IEEE Trans.Comput.*, C-18 (5), 401-409.

Bibliography

- Utsch, A., and Siemon, H.P., 1990. "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis". *Proceedings of International Neural Network Conference (INNC'90)*, 305-308.
- Van Gils, M.J., 1995. "Peak Identification in Auditory Evoked Potentials Using Artificial Neural Networks". PhD thesis, Eindhoven University of Technology.
- Varfis, A. and Versino, C., 1992. "Clustering of Social-Economic Data with Kohonen Maps". *Neural Network World*, 2 (6), 813-834.
- Vesanto, J., and Alhoniemi, E., 2000. "Clustering of the Self-Organizing Map". *IEEE Transactions on Neural Networks*, 11 (3), 586-600.
- Vesanto, J., Himberg, J., Alhoniemi, E. and Parhankangas, J., 2000. SOM in Toolbox for Matlab 5. Helsinki University of Technology, Finland.
- Visa, A., Toivonen, J., Back, B. and Vanharanta, H., 2000. "A New Methodology for Knowledge Retrieval from Text Documents". *Proceedings of TOOLMET 2000 Symposium- Tool Environments and Development Methods for Intelligent Systems*, 147-151.
- Wilks, S.S., 1932. "Certain Generalizations in the Analysis of Variance". *Biometrika*, 24, 471-494.
- Xu, L., Chan, W. and King, I., 1998. "Enhancing SOM-based Data Visualization by Linking Different Data Projections". *Intelligent Data Engineering and Learning (IDEAL)*, Springer, 427-434.

Appendix A: How to Use the Fund Data

The information below is provided by Kiplinger.com and is reproduced in its entirety from <http://www.kiplinger.com/test/glossary01.htm> purely for reference purposes and as a glossary.

Total return measures the change in the value of an investment, assuming that distributions of dividends and capital gains were reinvested. Returns for three- and five-year periods are annualized -- that is, stated on an average annual basis after compounding. Returns, minimum investments and expense data are as of the last trading day of the previous month. All information is updated monthly by Standard & Poor's. Expenses (management and 12b-1 fees, if any) are reflected in returns; sales and redemption fees are not. A dash in the three- or five-year column means the fund hasn't existed that long.

Fund universe describes what the fund invests in, in the broadest sense -- U.S. stocks, international stocks, taxable bonds or tax-free bonds.

Fund category reveals how the portfolio manager intends to invest the fund's assets and gives you a sense of how risky the fund may be. A fund's category -- aggressive-growth or growth-and-income, for example -- should reflect your investment objectives, time horizon and risk tolerance.

AggGro (aggressive-growth) funds seek maximum capital gains, often in somewhat volatile small-company stocks.

Growth (long-term-growth) funds also seek capital gains but tend to invest in stocks of larger, more stable companies.

Gro-Inc (growth-and-income) funds concentrate on safety and dividend income.

Balanced funds invest for income by holding bonds as well as stocks.

Sector funds limit investments to one industry or economic sector.

Utility funds own primarily stocks of utilities and usually emphasize yield.

Diversified International stock funds invest primarily in stocks of companies based outside the U.S.

Specialized International stock funds invest in a single country or region of the world.

Global stock funds can invest in the U.S. as well as overseas.

Corp (*investment-grade corporate*) bond funds emphasize credit safety by owning mostly BBB-rated or better corporate bonds.

Appendix A: How to Use Fund Data

Yld Corp (high-yield-corporate) bond funds seek higher yields by owning mostly lower-rated corporate bonds.

Govt (U.S. government) funds hold Treasury securities or securities of federal agencies.

Mortgage funds own mortgage-backed securities, such as Ginnie Maes or collateralized mortgage obligations.

Global Bond funds can hold either U.S. or foreign bonds.

Muni funds emphasize credit safety by investing in mostly BBB-rated or better municipal bonds.

Yld muni funds seek higher yields by owning mostly lower-rated or unrated 'junk' muni bonds.

Investment style describes the personality of the fund as determined by the securities in which it invests.

For stock funds, style is divided into large (with a stock-market value of \$10 billion or more), midsize, and small (less than \$2 billion) companies, and characterized as rapidly growing, undervalued or a blend of the two. Nine styles derive from these elements: LarGro, LarBlnd, LarVal, MidGro, MidBlnd, MidVal, SmlGro, SmlBlnd and SmlVal. Other styles: AllCap (companies of all sizes), Sector (a single economic sector) and Hybrid (balanced, convertible-bond and asset-allocation funds).

Similarly, bond-fund styles describe the types and maturities of bonds in the fund's portfolio. The nine bond styles are: Short Term Gov't, Intermed. Term Gov't, Long Term Gov't, Short Term Corp., Intermed. Term Corp., Long Term Corp., High Yield Corp., Mortgage, Int'l.

Decile rank shows performance for each of the past five 12-month periods compared with other mutual funds using the same investment style. Funds are ranked 1 (top 10%) to 10 (bottom 10%). A decile rank of 5 or 6 is average.

The decile ranking offers a quick gauge of how a fund performed relative to its peers. Use the decile rankings rather than total returns to gauge consistency year to year. You will never find a fund ranked 1 across the row, but most good funds have at least three years of above-average results out of every five.

Down Market (*NOTE: To search for down market, you must select a fund universe.) Shows performance during the bear market, March 24, 2000, to April 4, 2001. During that span, the S&P 500 returned -26.9%. This figure indicates the extent to which a fund was taking on risk by investing in high-P/E tech stocks. For foreign stocks, use the return shown for the most recent down market for U.S. stocks to see how an international fund fared during a severe downturn. For taxable bonds, the last down market extended from January 29, 1999, through June 11, 1999. For tax-free bonds, the last down market extended from April 16, 1999, through October 22, 1999. Since each

fund category looks at a different period of time, you must select a "fund universe" for this search to operate properly.

Volatility rank measures volatility among all stock (or bond or muni) funds on a scale of 1 (least volatile) through 10 (most volatile) and indicates the degree to which a fund could decline in value over the short term in a down market or rise in an up market. The higher the volatility, the more potential for gain or loss. Volatility is based on standard deviation -- how much a fund's month-to-month returns varied from its average rate of return.

Turnover indicates how frequently the fund's manager trades stock. A turnover of 10% means that, on average, the manager replaced 10% of the portfolio over the past year. A turnover of 200% means that, on average, the manager replaced the entire portfolio twice over the past year. The biggest downside of high turnover is that it can add to a fund's trading costs. Otherwise, rely on turnover to get a better sense of a manager's style.

Assets are for individual share classes. A couple of rules of thumb about the size of funds: Bigger is not always better for funds that invest in small companies, primarily aggressive-growth funds, and too much money sometimes drags down performance of the very largest funds.

Manager since shows how long the manager (or if there's more than one, the lead manager) has been on board. This gives you assurance that, for example, the person who racked up sterling results is still in charge.

Minimum investment is what it takes to open an account. For subsequent investments and for IRAs, it's usually lower. Some funds also accept lower initial amounts if you sign up for regular investments via electronic transfer from your bank account.

Maximum load is the highest sales fee for purchasing shares. None means the fund charges no sales or redemption fees. Sales charges are not included when calculating total returns in these tables. But the shorter the time you own a fund, the greater the impact of the sales charge on your total return. You'll also pay the load each time you buy subsequent shares (but seldom when reinvesting dividends or capital gains).

Expense ratio is the percentage of your assets in a fund you'll pay, measured on an annual basis, to cover management fees and other expenses. Expenses are subtracted before total return is figured.

Yield shows total dividends for the past 12 months divided by the sum of the last available share price plus capital-gains distributions over the 12-month period. As with total returns, yields are calculated after expenses have been subtracted.

Average maturity is the average number of years the fund's income securities will take to mature, weighted by the value of each security. In general, the higher the number, the more susceptible the fund is to interest-rate swings. Longer maturities benefit you when interest rates go down (causing the prices of bonds to rise) and hurt you when rates rise.

Bond rating average measures how safe a fund's average bond, weighted by the value of each security, is from default. The higher the letter grade, the sturdier the fund's bonds. Funds rated AAA are safest. AA is next, and so on.

Appendix B: Correlation Matrices

Table B-1. Correlation matrix of bond funds

	I_Y return	3_Y return	5_Y return	DR02	DR01	DR00	DR99	D_Market	Vola- tility	Assets	Manage r	Min_ invest	E_ ratio	Yield
I_Y return	1													
3_Y return	0.76	1												
5_Y return	0.67	0.87	1											
DR02	-0.2	-0.42	-0.4	1										
DR01	-0	-0.3	-0.2	-0.04	1									
DR00	-0	-0.01	-0.2	-0.07	0.38	1								
DR99	-0	0	-0.2	0.25	-0.2	-0.2	1							
D_Market	-0.3	-0.12	-0.4	0.07	-0.3	-0.1	0.25	1						
Volatility	-0.3	-0.42	-0.4	-0.03	0.26	0.19	-0.3	-0.09	1					
Assets	-0	0.01	0.05	-0.12	-0.1	-0.1	-0.1	0.05	0	1				
Manager	-0	-0.06	-0.1	0.05	0.01	0.02	0.03	0.06	-0	-0.1	1			
Min_invest	0.06	0.05	0.06	0.04	-0.1	-0.2	0.05	0	-0.1	-0	0	1		
E_ratio	-0.2	-0.27	-0.4	0.21	0.22	0.24	0.14	0.1	0.31	-0.2	0	-0.2	1	
Yield	-0.5	-0.4	-0.4	-0.09	-0	0.16	-0.2	0.37	0.53	0.11	0	-0.1	0.1	1

Table B-2. Correlation matrix of stock funds

	I_Y return	3_Y return	5_Y return	DR02	DR01	DR00	DR99	Down Market	Vola- tility	Turn over	Assets	Manager	Min_ invest	E- ratio
I_Y return	1													
3_Y return	0.74	1												
5_Y return	0.32	0.7	1											
DR02	-0.5	-0.61	-0.5	1										
DR01	0.49	0.18	-0	-0.62	1									
DR00	0.38	0.31	-0.2	-0.45	0.43	1								
DR99	0.09	0.05	-0.3	-0.03	0.01	0.28	1							
D_Market	0.61	0.65	0.44	-0.86	0.62	0.44	0	1						
Volatility	-0.3	-0.19	-0.2	0.54	-0.5	-0.3	0.02	-0.66	1					
Turnover	-0.1	-0.07	-0.1	0.18	-0.2	-0	0.03	-0.19	0.24	1				
Assets	-0	-0.01	0.09	-0.06	-0	-0.1	-0.1	0.02	-0.1	-0.1	1			
Manager	-0.1	-0.09	-0.1	0.14	-0.1	-0	0.02	-0.13	0.13	0.1	-0.3	1		
Min_invest	0.02	0.03	0.03	-0.01	0.01	-0	0.01	0.02	0.02	0.02	0.02	-0	1	
E_ratio	0.03	-0.05	-0.3	0.11	0.02	0.09	0.14	-0.1	0.21	0.12	-0.2	0.05	-0.1	1

Appendix C: Scatterplot Matrices

Figure C-1. Scatterplot matrix with histograms on diagonal (for Bonds).

Figure C-2. Scatterplot matrix with histograms on diagonal (for Stocks).

Appendix D: Histograms of Variables

Figure D-1. Histograms of variables for Bond funds (cont'd. on next page)

Figure D-1. Histograms of variables for Bond funds (cont'd on next page)

Figure D-1. Histograms of variables for Bond funds.

Figure D-2. Histograms of variables for Stock Funds (cont'd. on next page).

Figure D-2. Histograms of variables for Stock Funds (cont'd. on next page).

Figure D-2. Histograms of variables for Stock Funds.

Appendix E: Cluster Post-descriptive Summary

Table E-1. Cluster post-descriptive summary of Bond funds

Cluster	No. of funds	Variables	1-y return	3-y return	5-y return	Down-market	Assets (million)	Manager since	Expense ratio	Yield
C1	481	Mean	6.5	6.1	6.2	-2.8	214.5	1995	0.96	4.9
		Median	6.7	6.4	6.4	-2.6	80	1996	0.9	5
		Min	-3	-2.5	-6.8	-19.2	0.4	1935	0	0
		Max	16.2	12.2	12.6	3.6	2977	2002	2.6	9
		Std	1.8	1.6	1.4	2.6	344.8	4.8	0.35	1.1
C2	31	Mean	6.8	6.7	6.7	-2.3	4477	1993	0.6	5.6
		Median	7.4	7.1	6.9	-2	3270	1994	0.6	5.4
		Min	0.4	1.8	4	-8.1	1870	1974	0.2	4
		Max	10.5	8.2	8	1.9	17234	2002	1.1	9
		Std	2.1	1.3	1	2	3733	6.5	0.2	1
C3	21	Mean	7.1	8	4.7	4.8	605.2	1994	1.3	9.5
		Median	6.7	6	4.7	2.7	129.4	1995	1.3	9.7
		Min	-2	-1.1	0.6	-1.2	14.4	1977	0.6	7
		Max	17.1	18	8.4	20.9	3678	2001	2.7	12.3
		Std	5	6.5	2.3	5.8	1014	4.9	0.5	1.3
C4	11	Mean	1.6	3.6	4.3	-0.7	532.4	1991	1.1	7.2
		Median	1.5	4	4.3	-0.9	492.2	1990	1	7.7
		Min	-1.4	1.2	2.8	-3.8	52.4	1983	0.9	4.5
		Max	5.2	5	5.6	1.5	1364	1998	1.4	8.2
		Std	2.4	1.1	0.8	0.2	500.5	5	0.1	1.3
C5	60	Mean	-2.1	-2.4	0.5	1.1	521.5	1997	1.2	9.5
		Median	-1.9	-1.9	0.8	0.7	270.3	1998	1.2	9.9
		Min	-15.9	-11.3	-5.6	-4.7	5.2	1985	0.4	0
		Max	7.4	4.4	4	13.9	3019	2002	2	13.5
		Std	4.8	3.5	2.2	3.3	670.7	4.1	0.3	2.3

Table E-2. Cluster post-descriptive summary of Bond funds

Cluster	No. of funds	Variable	1-y return	3-y return	5-y return	Down-market	Turn-over	Assets (million)	Manager since	E-ratio
C1	24	Mean	-6.4	2.1	9.3	-7.9	41.6	24505	1987	0.65
		median	-6.5	2.1	9.2	-1.5	34.4	19991	1991	0.7
		min	-24.9		5	-51	4	9391	1958	0.2
		max	8.4	15.8	20.5	20.2	141	70094	2002	1
		std	7.9	5.2	3.4	21.61	32.9	15067	11.5	0.2
C2	382	mean	4.2	9.9	10.5	5.4	74.1	635.6	1993	1.3
		median	3.8	10.1	10.1	6.1	61.7	161	1995	1.2
		min	-27.3	-6.4	-1.4	-39.4	0.3	0.7	1931	0.3
		max	34	40.7	31.6	58.9	315	9347	2002	3.6
		std	10.5	6.2	4	17.1	54.4	1214	7.5	0.4
C3	15	mean	92.6	24.2	0.5	-4.3	71.6	177.2	1996	2.1
		median	96.2	25.7	2.3	-7.3	5.2	101.4	1996	2
		min	42.5	5.8	-17.5	-32.1	5	29.4	1991	0.6
		max	138.5	37.2	7.9	75.9	386.4	555.2	2000	5.8
		std	30.3	7.2	6.9	23.9	92.4	172.4	2.5	1.2
C4	8	mean	-9.2	2.4	8.1	-31.1	85	5278	1988	0.65
		median	-6.7	2.3	7.3	-26.4	42	5141	1988	0.65
		min	-26	-3.5	5.8	-53.6	20	3210	1985	0.3
		max	0.3	7.6	12.6	-13.7	284.3	7526	1992	1.1
		std	8.2	4.3	2.4	13.1	90.3	1532	2.1	0.3
C5	34	mean	-13.1	-2.2	7.3	-24.7	57.3	6587	1994	0.85
		median	-13.3	-3.1	7	-24.6	50.6	6125	1994	0.9
		min	-30.7	-9.7	3.2	-55.2	1	92.9	1989	0.2
		max	-3.4	11	12.1	2.3	217	14761	2002	1.7
		std	5.8	4	2.1	13.5	49	3600	3.7	0.3
C6	909	mean	-13.9	-2.5	5.3	-26	85.2	519.9	1997	1.2
		median	-13.2	-2.5	5.4	-26.8	73.8	141.4	1997	1.2
		min	-48.4	-25.6	-8	-81.4	1	0.3	1980	0.1
		max	18	27.3	35.9	38.2	359.6	9315	2002	2.9
		std	9.2	6.4	4	19.3	62.3	1002	3.7	0.3

Table E-2. Cluster post-descriptive summary of Bond funds (contd.)

Cluster	No. of funds	Variable	1-y return	3-y return	5-y return	Down-market	Turn-over	Assets (million)	Manager since	E-ratio
C7	71	mean	-20.5	-6	3	-38.9	387	409.1	1995	1.6
		median	-21.1	-4.8	3.9	-39.9	319	76.9	1996	1.5
		min	-61.8	-38.8	-21.6	-87.6	70	0.6	1975	0.7
		max	62.4	16.6	23.8	155	1848	7717	2001	2.8
		std	18.4	11.2	7.2	24.3	259	1189	4.6	0.4
C8	27	mean	-31.7	-16.9	-3.5	-63.2	170.8	173.6	1998	1.5
		median	-34.1	-16.8	-3.6	-63.6	191.8	73.9	1999	1.5
		min	-52.8	-33.4	-14.4	-84.1	20	4.1	1992	0.7
		max	-17.2	-7.7	6.4	-40.9	291.4	776.3	2002	2.4
		std	10.6	5.4	5.5	10.9	75.8	207.5	2.6	0.4
C9	8	mean	-30.2	-22.2	-11.4	-59.5	182	5.78	1995	6.4
		median	-30.1	-23.1	-8.6	-66.4	138	2.15	1996	4.8
		min	-50.8	-39	-33.2	-83.9	1	0.3	1990	2.4
		max	-8.9	-7.4	1.3	-25	501.7	21.7	1999	14.7
		std	12.7	11.1	10.6	21.1	1176	7.5	2.9	4.5

Appendix F: Box Plot of Post-clustering Summary

Appendix F: Box Plot of Post-clustering Summary

Figure F-1. Box plot per variable of post-clustering summary of Bond funds, where thick lines represent the mean and standard deviation; horizontal thin lines represent the minimum, median, and maximum respectively. The number in upper right corner denotes cluster number and the number of funds in the cluster.

Appendix F: Box Plot of Post-clustering Summary

1-year return

3-year return

5-year return

Down-market

Appendix F: Box Plot of Post-clustering Summary

Figure F-2. Box plot per variable of post-clustering summary of Stock funds, where thick lines represent the mean and standard deviation; horizontal thin lines represent the minimum, median, and maximum respectively. The number in upper right corner denotes cluster number and the number of funds in the cluster.

Appendix G: MANOVA Results of Bonds and Stocks

Table G-1. MANOVA Results of Discovered Data Structure Focused on Bonds

Source Of Variation	Matrix of Sum of Squares and Cross Products	Degree of Freedoms
Treatment (B)	$\begin{bmatrix} 0.22221 & & & & & & & & \\ 0.18282 & 0.15052 & & & & & & & \\ 0.15033 & 0.1237 & 0.10186 & & & & & & \\ -0.19103 & -0.15684 & -0.12946 & 0.16724 & & & & & \\ -2866.6 & -2459.8 & -1805.6 & 2083.7 & 2.9389\text{e}+008 & & & & \\ 0.16948 & -0.26024 & -0.50297 & -1.2272 & -25450 & 4012.7 & & & \\ -0.009199 & -0.00745 & -0.00630 & 0.00812 & -119.15 & 0.0681 & 0.00063 & & \\ -0.17801 & -0.14586 & -0.12041 & 0.15599 & 1787.6 & -2.7324 & 0.00786 & 0.14669 & \end{bmatrix}$	4
Residual (W)	$\begin{bmatrix} 0.57421 & & & & & & & & \\ 0.38721 & 0.54939 & & & & & & & \\ 0.19057 & 0.2936 & 0.22417 & & & & & & \\ -0.016918 & 0.07131 & -0.03428 & 0.50721 & & & & & \\ 2199.9 & 2769.4 & 2798.1 & -862.97 & 7.6945\text{e}+008 & & & & \\ -2.3791 & -5.6761 & -4.8059 & 6.808 & -2.9269\text{e}+005 & 10768 & & & \\ -0.0046777 & -0.01362 & -0.01585 & -0.00067 & -575.39 & -0.0494 & 0.0078125 & & \\ -0.048838 & -0.02349 & -0.002613 & -0.00551 & 38.188 & 2.7855 & -0.0032559 & 0.10361 & \end{bmatrix}$	599
Total Corrected (T)	$\begin{bmatrix} 0.79642 & & & & & & & & \\ 0.57003 & 0.69991 & & & & & & & \\ 0.3409 & 0.4173 & 0.32603 & & & & & & \\ -0.20795 & -0.08552 & -0.16375 & 0.67445 & & & & & \\ -666.71 & 309.59 & 992.5 & 1220.8 & 1.0633\text{e}+009 & & & & \\ -2.2097 & -5.9363 & -5.3088 & 5.5808 & -5.4727\text{e}+005 & 14781 & & & \\ -0.013876 & -0.021078 & -0.02216 & 0.0075 & -694.54 & 0.01872 & 0.0084 & & \\ -0.22685 & -0.16935 & -0.12302 & 0.15048 & 1825.7 & 0.05313 & 0.0046 & 0.2503 & \end{bmatrix}$	603
$\lambda^* = \frac{ W }{ B+W }$		0.0664
Chi-square $\chi^2_{p(g-1),(.01)}$		53.45
Bartlett's calculation $-(n-1-(p+g)/2)\ln\left(\frac{ W }{ B+W }\right)$		1360.3

Table G-2 MANOVA Results of Centroid Clustering Focused on Bonds

Source Of Variation	Matrix of Sum of Squares and Cross Products	Degree of Freedoms
Treatment (B)	$\begin{bmatrix} 0.17714 & & & & & & & & \\ 0.13293 & 0.10312 & & & & & & & \\ 0.11092 & 0.085526 & 0.071036 & & & & & & \\ -0.17885 & -0.12656 & -0.10679 & 0.19801 & & & & & \\ -1917.3 & -1190.1 & -988.94 & 2518 & 1.1084\text{e}+008 & & & & \\ 14.804 & 11.239 & 9.1897 & -14.714 & -4.3353\text{e}+005 & 2358.5 & & & \\ -0.003468 & -0.00461 & -0.00355 & -0.00107 & -140.5 & -0.25132 & 0.00128 & & \\ -0.14066 & -0.11043 & -0.091316 & 0.13093 & 1289.1 & -12.447 & 0.005617 & 0.1189 & \end{bmatrix}$	3
Residual (W)	$\begin{bmatrix} 0.61927 & & & & & & & & \\ 0.4371 & 0.59679 & & & & & & & \\ 0.22997 & 0.33177 & 0.255 & & & & & & \\ -0.02909 & 0.04103 & -0.05696 & 0.47643 & & & & & \\ 1250.6 & 1499.6 & 1981.4 & -1297.2 & 9.525\text{e}+008 & & & & \\ -17.013 & -17.176 & -14.499 & 20.295 & -1.1374\text{e}+005 & 12422 & & & \\ -0.010408 & -0.01646 & -0.018606 & 0.008533 & -554.05 & 0.27005 & 0.00716 & & \\ -0.086196 & -0.05891 & -0.03170 & 0.019546 & 536.69 & 12.5 & -0.0010 & 0.1313 & \end{bmatrix}$	600
Total Corrected (T)	$\begin{bmatrix} 0.79642 & & & & & & & & \\ 0.57003 & 0.69991 & & & & & & & \\ 0.3409 & 0.4173 & 0.32603 & & & & & & \\ -0.20795 & -0.08552 & -0.16375 & 0.67445 & & & & & \\ -666.71 & 309.59 & 992.5 & 1220.8 & 1.0633\text{e}+009 & & & & \\ -2.2097 & -5.9363 & -5.3088 & 5.5808 & -5.4727\text{e}+005 & 14781 & & & \\ -0.013876 & -0.021078 & -0.02216 & 0.0075 & -694.54 & 0.01872 & 0.0084 & & \\ -0.22685 & -0.16935 & -0.12302 & 0.15048 & 1825.7 & 0.05313 & 0.0046 & 0.2503 & \end{bmatrix}$	603
$\lambda^* = \frac{ W }{ B + W }$		0.0890
Chi-square $\chi^2_{p(g-1),(.01)}$		42.98
Bartlett's calculation $-(n-1-(p+g)/2)\ln\left(\frac{ W }{ B+W }\right)$		1444.8

Table G-3 MANOVA Results of Centroid Clustering Focused on Stocks

Source Of Variation	Matrix of Sum of Squares and Cross Products	Degree of Freedoms
Treatment (B)	$\begin{bmatrix} 1.7593 & & & & & & & & \\ 1.1487 & 0.80433 & & & & & & & \\ 0.2211 & 0.16502 & 0.0579 & & & & & & \\ 0.1400 & -0.02055 & 0.0876 & 7.8816 & & & & & \\ 18.516 & 13.932 & 2.0484 & -36.782 & 431.09 & & & & \\ -35123 & -30721 & -5725.7 & -33996 & -4.413e+005 & 2.8736e+009 & & & \\ 26.683 & 35.949 & 1.262 & -50.896 & 1163.2 & -3.0625e+006 & 14200 & & \\ 0.1093 & 0.07913 & 0.01074 & -0.089346 & 1.8775 & -3866.5 & 5.0134 & 0.0105 \end{bmatrix}$	6
Residual (W)	$\begin{bmatrix} 41.629 & & & & & & & & \\ 16.489 & 12.258 & & & & & & & \\ 4.0146 & 4.9826 & 4.0982 & & & & & & \\ 37.706 & 22.383 & 8.3244 & 81.866 & & & & & \\ -48.983 & -23.926 & -10.301 & -38.35 & 1238.8 & & & & \\ 16905 & 25529 & 33325 & 55517 & -24661 & 1.9145e+010 & & & \\ -124.96 & -103.81 & -42.793 & -212.5 & -274.39 & -4.8281e+006 & 31832 & & \\ -0.0670 & -0.1294 & -0.14149 & -0.162 & -0.63488 & -3131.8 & -2.094 & 0.0535 \end{bmatrix}$	1471
Total Corrected (T)	$\begin{bmatrix} 43.388 & & & & & & & & \\ 17.637 & 13.062 & & & & & & & \\ 4.2358 & 5.1476 & 4.1561 & & & & & & \\ 37.846 & 22.362 & 8.412 & 89.748 & & & & & \\ -30.467 & -9.993 & -8.2524 & -75.139 & 1669.9 & & & & \\ -18218 & -5192 & 27600 & 21521 & -4.65e+005 & 2.2019e+0103 & & & \\ -98.277 & -67.86 & -41.531 & -263.4 & 888.8 & -7.8907e+006 & 46033 & & \\ 0.04231 & -0.050 & -0.13074 & -0.2515 & 1.2426 & -6998.3 & 2.9194 & 0.0641 \end{bmatrix}$	1477
$\lambda^* = \frac{ W }{ B + W }$		0.1127
Chi-square $\chi^2_{p(g-1),(.01)}$		73.65
Bartlett's calculation $-(n-1-(p+g)/2) \ln \left(\frac{ W }{ B+W } \right)$		3208.0

Table G-4 MANOVA Results of Discovered Data Structure Focused on Stocks

Source Of Variation	Matrix of Sum of Squares and Cross Products	Degree of Freedoms
Treatment (B)	$\begin{bmatrix} 1.7865 & & & & & & & & \\ 1.0664 & 1.3282 & & & & & & & \\ 0.3537 & 0.7940 & 0.6355 & & & & & & \\ 3.1973 & 1.0735 & 0.19076 & 7.9774 & & & & & \\ -2.4864 & 5.3032 & 1.3733 & -23.752 & 739.01 & & & & \\ -12694 & -10073 & 9085.1 & 10629 & -3.9556e+005 & 1.33e+010 & & & \\ 61.187 & 58.58 & 8.548 & -10.109 & 741.18 & -4.32e+006 & 12731 & & \\ -0.0557 & -0.0639 & -0.0705 & -0.16585 & 1.273 & -4377 & 5.5315 & 0.02431 \end{bmatrix}$	8
Residual (W)	$\begin{bmatrix} 41.601 & & & & & & & & \\ 16.571 & 11.734 & & & & & & & \\ 3.8821 & 4.3535 & 3.5206 & & & & & & \\ 34.649 & 21.289 & 8.2213 & 81.77 & & & & & \\ -27.981 & -15.297 & -9.6257 & -51.387 & 930.91 & & & & \\ -5523.7 & 4881.4 & 18514 & 10892 & -70406 & 8.702e+009 & & & \\ -159.46 & -126.44 & -50.079 & -253.29 & 147.62 & -3.568e+006 & 33302 & & \\ 0.09809 & 0.013734 & -0.0601 & -0.08572 & -0.03031 & -2621.3 & -2.6122 & 0.03976 \end{bmatrix}$	1469
Total Corrected (T)	$\begin{bmatrix} 43.388 & & & & & & & & \\ 17.637 & 13.062 & & & & & & & \\ 4.2358 & 5.1476 & 4.1561 & & & & & & \\ 37.846 & 22.362 & 8.412 & 89.748 & & & & & \\ -30.467 & -9.993 & -8.2524 & -75.139 & 1669.9 & & & & \\ -18218 & -5192 & 27600 & 21521 & -4.65e+005 & 2.2019e+0103 & & & \\ -98.277 & -67.86 & -41.531 & -263.4 & 888.8 & -7.8907e+006 & 46033 & & \\ 0.04231 & -0.050 & -0.13074 & -0.2515 & 1.2426 & -6998.3 & 2.9194 & 0.0641 \end{bmatrix}$	1477
$\lambda^* = \frac{ W }{ B + W }$		0.0749
Chi-square $\chi^2_{p(g-1),(.01)}$		93.2
Bartlett's calculation $-(n-1-(p+g)/2) \ln \left(\frac{ W }{ B+W } \right)$		3805.2

Appendix H: Statistical Summary of Existing Clustering of Bonds and Stocks

Table H-1. Statistical Summary of Bond Funds using Single Linkage

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Yield (%)
C1	157	7.3	6.9	7.1	-4.1	\$1088.4	1995	0.7	5.4
C2	382	6.0	5.9	5.8	-1.6	\$238.9	1995	1.0	5.4
C3	20	5.0	5.8	5.8	0.65	\$107.6	1994	0.8	4.3
C4	45	7.5	6.8	6.9	-4.3	\$134.3	1995	0.95	5.1

Table H-2. Statistical Summary of Bond Funds using Complete Linkage

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Yield (%)
C1	252	7.1	6.8	6.9	-3.3	\$773.5	1993	0.77	5.3
C2	195	6.0	6.5	6.0	-1.1	\$202.4	1997	0.97	5.3
C3	78	6.9	6.5	6.6	-2.7	\$1522.6	1990	0.88	5.5
C4	79	6.9	6.9	6.8	-2.2	\$343.4	1995	0.61	5.1

Table H-3. Statistical Summary of Bond Funds using Centroid

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Yield (%)
C1	107	7.5	7.0	7.3	-4.9	\$149.4	1994	0.81	5.3
C2	416	6.1	6.1	6.0	-1.6	\$586.1	1995	0.96	5.4
C3	36	4.7	5.2	5.5	-0.8	\$1037.2	1991	0.96	6.1
C4	45	7.6	6.9	6.9	-3.7	\$83.5	1995	0.9	5.2

Appendix H: Statistical Summary of Existing Clustering of Bonds and Stocks

Table H-4. Statistical Summary of Stock Funds using Single Linkage

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Turnover (%)
C1	67	-10.0	-0.4	8.1	-20	\$12917.7	1991	0.68	50
C2	64	-11.0	-2.8	5.6	-22	\$476.5	1996	0.55	20
C3	446	-18.0	-6.3	3.5	-35	\$570.0	1996	1.2	81
C4	51	-26.0	-9.0	2.2	-51	\$817.1	1998	1.9	150
C5	471	-18.0	-6.2	3.5	-35	\$727.4	1996	1.2	82
C6	182	-24.0	-8.7	1.8	-45	\$624.4	1994	1.6	160
C7	197	-17.0	-6.3	3.1	-33	\$336.5	1996	1.4	68

Table H-5. Statistical Summary of Stock Funds using Centroid

Cluster	No. of funds	1-y return (%)	3-y return (%)	5-y return (%)	Down-Market (%)	Assets (million)	Manage Since	Expense Ratio (%)	Turnover (%)
C1	206	-6.0	2.3	8.2	-7.1	\$4911.0	1988	0.96	55
C2	72	-9.1	-1.5	5.6	-18.0	\$509.8	1995	0.64	30
C3	174	-5.0	2.4	7.8	-4.7	\$908.8	1988	0.94	48
C4	350	-14.0	-3.6	4.8	-25.0	\$408.2	1993	1.2	54
C5	225	-18.0	-6.3	3.5	-34.0	\$361.8	1995	1.3	61
C6	256	-15.0	-4.4	4.5	-27.0	\$378.3	1993	1.2	53
C7	195	-14.0	-4.1	4.4	-26.0	\$334.1	1993	1.3	56

Appendix I: Symbols of Mutual Funds in Cluster of Bond Funds

Cluster 1:

TRUSX,WTLTX,RYGBX,BTTRX,BTTTX,SEGSX,CMPIX,GSBOX,OGGAX,PAGVX,AOBDX,GUIQX,APBDX,
CFSAX,PCDFX,ATRRX,BFIFX,FGOIX,ADFSX,JHGX,VUGVX,AIGRX,EUSAX,THOPX,ONIX,DPIAX,
KIFIX,NWINX,FOFIX,TSWFX,PCSIX,LETMB,MUUA,BCBAX,CMIAX,CTYLX,GAHQX,WEIBX,GALBX,
NVGAX,UMMGX,SIUSX,TAUSX,STATB,HTBAX,SAMBX,AQFIX,EKSAX,CMPGX,FPFTX,BNIGX,PFIX,
CSBNX,SRINX,PRPIX,AAINX,RIBDX,FRBMX,OUSGX,PFCCX,FIUAX,FAGVX,ACGVX,GOITX,LGVX,
FIGVX,PINCX,PRFIX,VGVIX,PBLGX,PUSGX,HBFAX,NGVAX,FMEQX,SFIUX,BABIX,NSFAX,SSIGX,
ROGVX,TINAX,PHBAX,CFIFX,MFBFX,AEBAX,AKIRX,BHIII,FIIGX,ACCBX,NOFIX,STBDX,MUCAX,
LEXNX,SCUSX,EKIX,HFIIX,HMGCX,PHGBX,RFUGX,PRBDX,LZBOX,TPINX,OIBAX,HUMSX,PNGBX,
BJBGX,ACUAX,QUFIX,MRUAX,DEGGX,PBGPX,GNMBX,BGVAX,CUSGX,FDLBX,HBDIX,NEFUX,GIGVX,
MBCTX,SSGTX,MBHQX,ITBAX,FSGSX,NEFLX,PRPGX,COFAX,LSGBR,PGTRX,FBHBX,AFIFD,MBFSX,
SHMGX,FBYBX,PIMGB,ABUSX,PZGAX,BUSGX,ONEFI,MCSGX,HGVVX,IDITX,NGTVX,FTIIX,SGTBX,
MGBX,VRPFX,MFIBX,TIBPX,IISBX,LMGGX,MBGOX,PAGIX,CPFAX,MBSIX,DSILX,DRGIX,FTGSX,
TCFIX,IOGSX,TMBUX,PGBOX,SIBOX,TATBX,MTIOX,OGBAX,ELINX,CHINX,GVITX,INISX,HMRBX,
SUSAX,MBDAX,CCIX,IISX,APBRX,ASGMX,SBCGX,FOIBX,CIGAX,HIGAX,BTRAX,TOBAX,VBDFX,
PCTAX,DRGMX,PCGTX,FMSAX,TGMBX,FGSSX,TWRGX,MGMTX,CGIAX,WTIBX,BBDSX,EFIX,NOUGX,
GGIFX,SIMIX,MRGIX,FTBRX,DTRIX,NSIGX,AMGEX,BBNDX,WBRRX,MUMAX,ATGAX,PCIFX,PSGNX,
AGNRX,SUSGX,TWTBX,PMNIX,KNLIX,GASTX,EQUAX,CSIBX,BSGAX,STGIX,BIAX,FBDPX,SBGLX,
ARMGX,SGSIX,BEGBX,CWBFX,IGBFX,MIBAX,BBIFX,PAGSX,ICSTX,TGLMX,PYOPX,SBIFX,FBDFX,
WPGVX,PRITX,MAABX,NIIBX,DBIRX,FTRFX,SSBMX,CPTNX,IUTIX,CCITB,UIMIX,MCMFX,FTARX,
PYGFX,FGFSX,FSMIX,PRSUX,DPIGX,SSINX,MCMNX,DIGFX,KATRX,GVSBX,RSDIX,AFIRX,IOFIX,
OTABX,DRITX,PRITX,STBFX,ANSTX,STAAR,UMGVX,PFSGX,NLMBX,MLOIX,HASDX,MWLDX,GSSDX,
PTLAX,MSLDX,SRHQX,OPGVX,SNGSX,CFICX,FRGVX,SNSDX,NSTRX,MQLFX,MAIBX,CSSTX,DWSHX,
NISBX,STINX,MSINX,DSTIX,SBSTX,IFINX,BLDAX,FASFX,PFSFX,FPLGX,PLTBX,STSFIX,STSBX,
NVSAX,DRBDX,DSINX,MBAJX,ILGAX,MMPIX,GITMX,LTDFX,CGFIX,TTRBX,NEFRX,PDBAX,PSHAX,
TPBIX,FLXBX,MASAX,UNHHX,CLINX,OPIGX,AMMSX,WAIMX,RPIBX,LGIX,TIFUX,PCGLX,PGGIX,
NOIFX,GTRAX,VKMGX,MCFIX,PRGMX,SPGVX,USGNX,PRCIX,FSTGX,SWLBX,VFISX,MWTRX,USAIX,
BGNMX,VBISX,PYUSX,ACITX,APSTX,RIMFX,RFBSX,PRWBX,TSDBG,TWUSX,SWBDX,MNTSX,ECUIX,
SLTMX,SFSIX,HLRGX,OLGVX,FAIIX,AINRX,AALPX,SHTIX,FALTIX,DSIGX,ONUAX,FSGIX,TGSMX,
TSDOX,AOLMX,AFTRX,CTYSX,LALDX,WNTBX,HSTGX,FLDSX,MGLFX,ASTTX,PBSMX,JASBX,GSAMX,
FSIX,ISBSX,HOSGX,FTRDX,SSYPX,MIBFX,PRVBX,RUSGX,USBX,EAVPR,SCAAX,TAIBX,TGCFX,
FIGIX,RMSX,CHTBX,CUIGX,MGIDX,FUSGX,TIUX,SBGIX,FITSX,IVFTX,UNBDX,UNGVX,SNGVX,
ARSAX,VALBX,AGOVX,BRSFX,THIFX,KNFVX,WHICX,FLINX,WEFIX,SIBPX,APFBX,ACFMX,BTSFX,
HOSBX,HIMBX,MGSDX,NEFAX,FISAX,NDIAX,FOSIX,CUFIX,CFXIX,LSBRX,FSIAX,NARAX,FRSTX,
CMHYX,FPNIX,HABDX,CFISX,AMUSX,ASMTX,JAFIX,MFGSX,SRBFX,ASCPX,ASITX,ADMSX,MGFIX,
MUBX,UMBBX,ARUGX,PIOBX,LTUSX,SNBDX,SSGIX,EVGX,SFUSX,PRGVX,WCIFX,CUGGX,FAFIX,
LGINX,WGOBX,JHFIX,JHNBX,EKDMX,LMIGX,INSEX,WTRBX,ENGX,ISHIX,NOAMX,MGSIX,NIBDX

Cluster 2:

VFIX,VBIIX,FSIBX,PGSIX,FGOVX,KUSAX,AGNMX,USGBX,VSGBX,RPSIX,FSHBX,SNIDX,INBNX,DODIX,VFI
TX,PTTAX,VWESX,VFIIX,FTHRX,FBNDX,FKUSX,FGMNX,VBMFX,VFSTX,ABNDX,VWEHX,STVSX,AIBAX,OPSI
X,ASARX,STADX

Cluster 3:

MBAMX,IHIX,AGDAX,PREMX,FAEMX,FNMIX,PEMAX,NEFZX,EMDFX,SSTAX,EVSGX,LHIRX,ANAGX,NCIN
X,SAHYX,NAMFX,PRHYX,TGHYX,NTHIX,LBNDX,AHITX

Cluster 4:

PDINX,ATSAX,JAHYX,UNHIX,ENHYX,PRTBX,BUFHX,CBFAH,SSTBX,IHIYX,STCBX

Cluster 5:

HRIDX,MFIOX,SLDSX,PHDAX,MTGVX,DISAX,KSTAX,SINCX,COSIX,EKHBX,SIHAX,AGAIX,AMIFX,STH
IX,LZSOX,MHYAX,OPCHX,SSHAX,DINBX,DHOAX,FIFIX,AAHYX,GSIX,RYJUX,TSHYX,KHYAX,MKHCX,
FHIIX,AGEFX,MBWIX,EVIBX,MHITX,VAGIX,FHYTX,DHIYX,LMHYX,AHYAX,DETWX,PBHYX,SPHIX,OPP
HX,INEAX,MSWAX,DPLTX,MBHIX,EVHIX,COLHX,FAHYX,SHIBX,SHNAX,PHYLX,PHCHX,SAFHX,AMHY
X,STHYX,ACHYX,VKHYX,FAGIX,NEFHX,SHYBX

Appendix J: Flexibility check of proposed method

Figure J-1. Clustering results from proposed explicit clustering method for Bonds by using (b) the average value and (c) minimum value of the cluster when applying the step 3 of the algorithm. (a) is the original U-matrix.

Figure J-2. Clustering results from proposed explicit clustering method for Stocks by using (b) the average value and (c) minimum value of the cluster when applying the step 3 of the algorithm. (a) is the original U-matrix.

Appendix K: Source Code

1. Data Extracting Code

```
*****
'* This program is used to extract data from downloaded HTML file
'* as well as to Add company symbol and to change delimiter from
'* "," to space for format purpose
'* All Rights are reserved by Danyu Xu, 2002
*****

Option Explicit

Dim fileName As String
Dim fileRoot As String
Dim fileNameStep2 As String
Dim fileNameStep3 As String
Dim fileNameFinal As String
Private Declare Function DeleteFile Lib "kernel32" Alias "DeleteFileA" (ByVal lpFileName As String) As Long

Private Sub cmdAddCompanySymbol_Click()

    Dim fileNum1 As Integer
    Dim fileNum2 As Integer
    Dim tmpStr As String
    Dim subStr
    Dim symbol As String
    Dim iLeft As Integer
    Dim iRight As Integer
    Dim itemNum As Integer
    Dim newLine As String
    Dim i%

    ' simple error handling
    On Error GoTo errHnd

    ' set hourGlass for the mouse pointer
    Screen.MousePointer = vbHourglass
    fileNum1 = FreeFile

    ' open file for input
    Open fileNameStep2 For Input As #fileNum1
    fileNum2 = FreeFile

    ' open file for output
    Open fileNameStep3 For Output As #fileNum2
    Do While Not EOF(fileNum1)
        Line Input #fileNum1, tmpStr
        subStr = Split(tmpStr, ",")
        itemNum = UBound(subStr)
        iLeft = InStr(subStr(0), "(")
        iRight = InStr(subStr(0), ")")
        If iRight > iLeft Then
```


Appendix K: Source Code

```
        symbol = Mid$(subStr(0), iLeft + 1, iRight - iLeft - 1)
    Else
        symbol = ""
    End If
    symbol = Chr$(34) + symbol + Chr$(34)
    newLine = subStr(0) + "," + symbol
    For i = 1 To itemNum - 1 ' 2
        newLine = newLine + "," + subStr(i)
    Next i

    ' save to new file
    Print #fileNum2, newLine
Loop

' close files
Close (fileNum1)
Close (fileNum2)

' set mouse pointer back to normal
Screen.MousePointer = vbNormal

' enable the button for the next step
cmdChangeDelimiter.Enabled = True

errHnd:

End Sub
'*****
'* To get rid of some extra space before or after "," in
'* the data file, using Tab and Space delimiter instead of
'* ", " when import the file to Excel for further process
'*****
Private Sub cmdChangeDelimiter_Click()

    Dim fileN As Integer
    Dim fileN2 As Integer
    Dim tmpStr$
    Dim filename2 As String
    Dim tmpstr2$

    On Error GoTo errhandler

    Screen.MousePointer = vbHourglass

    fileN = FreeFile
    Open fileNameStep3 For Input As #fileN
    fileN2 = FreeFile
    Open fileNameFinal For Output As #fileN2
    Do While Not EOF(fileN)
        Line Input #fileN, tmpStr

        ' replace the ", " with space
        tmpstr2 = Replace(tmpStr, ", ", " ")
        Print #fileN2, tmpstr2
    Loop

    ' close the files
```

Appendix K: Source Code

```
Close fileN2
Close fileN

Screen.MousePointer = vbNormal

errhandler:

End Sub

Private Sub cmdExtractAll_Click()

    Dim i%

    ' hard coded to 80 files
    For i = 1 To 80
        fileName = fileRoot + Trim(Str(i)) + ".htm"
        cmdExtract_Click
    Next i
End Sub

Private Sub cmdInputFile_Click()

    On Error GoTo errhandler

    CommonDialog1.CancelError = True

    ' show open dialog
    CommonDialog1.ShowOpen
    fileName = CommonDialog1.fileName
    fileRoot = Left$(fileName, Len(fileName) - 5)

    ' enable the buttons for the next step
    cmdExtract.Enabled = True
    cmdExtractAll.Enabled = True

errhandler:

End Sub

*****
'*   The main code for extract data from HTML file
*****
Private Sub cmdExtract_Click()

    Dim startF As Boolean
    Dim stopF As Boolean
    Dim mark As Integer ' 1 as start (>), -1 as end (<)
    Dim fSum As Integer ' sum of < and >, < +1, > -1
    Dim fnum As Integer
    Dim strLine As String
    Dim lineLen As Integer
    Dim i%
    Dim char As String
    Dim strNeed As String
    Dim count As Integer
    Dim data(26) As String
```

Appendix K: Source Code

```
Dim outf As Integer
Dim lineCount As Integer
Dim recordCount As Integer

On Error GoTo errhandler

Screen.MousePointer = vbHourglass
fnum = FreeFile

' open file for input
Open fileName For Input As #fnum
outf = FreeFile
' open file for output as append
Open fileNameStep2 For Append As #outf
Do While startF = False And EOF(fnum) = False
    Line Input #fnum, strLine
    lineCount = lineCount + 1

    ' use the "Notes" to indicate the start of the data
    If InStr(strLine, "Notes") Then
        startF = True
    End If
Loop
' get the data
Do While (Not EOF(fnum)) And (recordCount < 50)

    ' read a line
    Line Input #fnum, strLine
    Dim strLine2 As String
    strLine2 = Trim(strLine)
    lineCount = lineCount + 1
    lineLen = Len(strLine2)
    For i = 1 To lineLen
        char = Mid$(strLine2, i, 1)
        If char = ">" Then
            fSum = fSum - 1
        ElseIf char = "<" Then
            If Trim(strNeed) <> "" And fSum = 0 Then
                count = count + 1
                data(count) = strNeed
                strNeed = ""
            End If
            fSum = fSum + 1
        Else
            If fSum = 0 Then
                strNeed = strNeed + char
            End If
        End If
    Next i
    If strNeed <> "" Then
        strNeed = strNeed + " "
    End If

    If count >= 25 Then
        ' If InStr(data(count), "*") Or InStr(data(count), "$") Then
        ' the above condition will cause problem for company name such
        ' as "E*Trade...."
```

Appendix K: Source Code

```
' so
If (Left(Trim(data(count)), 1) = "*" Or Left(Trim(data(count)), 1) = "$") And Mid$(Trim(data(count)),
3, 1) <> "." Then
    ' also the last line of each file ---**--close...
    data(24) = data(count)
    count = 0
    ' write to file
    For i = 1 To 23
        Write #outf, data(i);
    Next i
    Write #outf, data(24)
Else
    data(24) = ""
    ' write to file
    For i = 1 To 23
        Write #outf, data(i);
    Next i
    Write #outf, data(24)
    data(1) = data(count)
    count = 1
End If
recordCount = recordCount + 1
End If
Loop

Close #outf ' close the files
Close #fnum
Screen.MousePointer = vbNormal
' enable the next step
cmdAddCompanySymbol.Enabled = True
errhandler:

End Sub

Private Sub cmdExit_Click()
    ' unload the program
    Unload Me
End Sub

Private Sub Form_Load()

    ' for simplicity, fix the output data file names
    fileNameStep2 = App.Path + "\data-step2.txt"
    fileNameStep3 = App.Path + "\data-step3.txt"
    fileNameFinal = App.Path + "\data-final.txt"
    ' delete the files if exist
    DeleteFile (fileNameStep2)
    DeleteFile (fileNameStep3)
    DeleteFile (fileNameFinal)
    ' disable the buttons
    cmdExtract.Enabled = False
    cmdExtractAll.Enabled = False
    cmdAddCompanySymbol.Enabled = False
    cmdChangeDelimiter.Enabled = False

End Sub
```

2. SOM visualization and analysis code

```
%=====
%
%      Mutual Fund data VISUALIZATION and ANALYSIS USING THE SOM
%      Copyright (c):Danyu Xu, 2003, Carleton University
%=====
%

% reset
clf reset;
f0 = gcf;
echo on
clc

% run parameters
fundsType = 2 % 1 for Bonds, 2 for Stock
methodType = 1 % 1 for existing method (centroid), 2 for improved method

% read data from file
if fundsType==1,
    D = som_read_data('mtBondSelect_8_604.txt');
    D.comp_names = {'I_Y return' '3_Y return' '5_Y return' 'Down market' 'Assets' 'Manager' 'E_ratio' 'Yield'};
    sD = som_normalize(D,'var'); % the data is normalized
    sM = som_make(sD,'msize',[9 6] , 'training',[20 80]) % training for bonds
else
    D = som_read_data('mtStockFinal_8.txt');
    D.comp_names = {'I_Y return' '3_Y return' '5_Y return' 'Down market' 'Turn over' 'Assets' 'Manager'
'E_ratio'};
    sD = som_normalize(D,'var'); % the data is normalized
    sM = som_make(sD,'msize',[13 10] , 'training',[20 80]) % training for stocks
end;
sM = som_autolabel(sM,sD,'add'); % map is labeled

%=====
%      Visualization of explicit clustering based on new clustering function
%=====
%Nov. 26
% [base,seed] = som_dmatchclusters(sM,'centroid','N1')

if methodType==1
    [base,seed] = som_dmatchclusters(sM,'centroid','N1');
    Unode=base;
else

    Umat = som_umat(sM);
    % take the average (median) only
    UmatM= Umat(1:2:size(Umat,1),1:2:size(Umat,2));
    if fundsType==1,
        [umatrix, zls] = Umatrix_cluster(UmatM,0.3);
    else
        [umatrix, zls] = Umatrix_cluster(UmatM,0.2);
    end;
    Ubase = umatrix;
```

Appendix K: Source Code

```

    Unode = Ubase(:,1);
    for i = 2:length(Ubase(1,:)),
        tw = Ubase(:,i);
        Unode = cat(1,Unode,tw);
    end;
    base = Unode;
end;

% Colour code the clusters
C=som_clustercolour(sM, base, 'rgb2');
% Visualize
f2=figure;
    som_show(sM,'colour',C);

% Colour code the clusters
C=som_clustercolour(sM, base, 'rgb1');
som_show(sM,'colour',C);

% get the company label for each cluster
company_label={} ; % labels for each cluster as cell
for i=1:zls, %cluster
    tmpStr='';
    for j=1:length(Unode), % node
        if Unode(j)==i,
            for k=1:size(sM.labels, 2), % element
                if strcmp(sM.labels(j,k),'')==0, % not ''
                    % company_label(i,:)=strcat(company_label(i,:),sM.labels(j,k));
                    tmpStr=strcat(tmpStr, sM.labels(j,k),',');
                end;
            end;
        end;
    end;
    % put the final tmpStr to the company_label
    company_label{i}=tmpStr;
end;

% DISPLAY
for i=1:zls,
    char(company_label{i}) % using char to make all displayed
end;

char

%=====
%          Post-clustering summary for discovery clusters
%=====
% the Best Matching units
[Bmus,Qerrors] = som_bmus(sM, sD);
sD = som_denormalize(sD);
% add BMU to the data vector and sort it according to BMU
sDB = [sD.data, Bmus];
[dlen, dim] = size(sDB);
sDB = sortrows(sDB,dim);

% Go through all BMU to get the map units matching with the map (columns)
%extracting BMU column
% no. of fund for each cluster

```

Appendix K: Source Code

```
fundNum=[];
sDBmus =sDB(:,dim);
for i=1:size(sDBmus,2),
    % Get unique class labels in ith classification
    c=unique(sDBmus(:,i)); % row vector for loop indexing
    % Go through all class in ith classification
    sBase=[];
    for j=c;
        index=(sDBmus(:,i)==j);
        N=sum(index);
        for k = 1:N;
            sBase = [sBase, base(j)];
        end;
    end;
end;
% transpose the vector
sBase = sBase';
% add Base classification to the data vector and sort it according to its classification
sDBB = [sDB, sBase];
[dlenB, dimB] = size(sDBB);
sDBB = sortrows(sDBB,dimB);

sD.comp_names{15} = 'BMUS';
sD.comp_names{16} = 'Base';

sD.data = sDBB;
n=0;
%extracting base column
sDD={};
cs = [] % cluster statistic matrix
ccs = [];
cs_cell = {};
ssBase =sDBB(:,dimB);
for i=1:size(ssBase,2),
    % Get unique class labels in ith classification
    c1=unique(ssBase(:,i)); % row vector for loop indexing
    % Go through all class in ith classification
    for j=c1;
        index=(ssBase(:,i)==j);
        N=sum(index);
        sDD{j}=sD.data((1+n):(N+n), 1:dimB-2);
        n = n + N;
        sD.data = [];
        sD2=[];
        sD2 = sDD{j};
        sD.data = sD2;
        ccs = [];
        som_info(sD,3);
        % record the fund number of the cluster
        fundNum(j)=size(sD.data, 1);
        csS = som_stats_danyu(sD,'nosort');
        for m = 1:(dimB-2),
            fprintf(1,' csS [%6.4f, %6.4f, %6.4f,
%6.4f,%6.4f]\n',csS{m}.min,csS{m}.max,csS{m}.mean,csS{m}.std,csS{m}.median);
            cs_min =csS{m}.min;
            cs_max = csS{m}.max;
            cs_mean = csS{m}.mean;
```

Appendix K: Source Code

```

        cs_std = csS{m}.std;
        cs_median = csS{m}.median;
        cs = [cs_min, cs_max, cs_mean, cs_std, cs_median];
        cs = cs';
        ccs = [ccs,cs];
    end
    fprintf(1, '\n');
    cs_cell{j} = ccs;
    fprintf(1, 'cs_cell{%%6.4f,%%6.4f,%%6.4f,%%6.4f}',cs_cell{j});
    fprintf(1, '\n');
    sD.data = sDBB;
end;
end;

%%=====
%%      MANOVA Test
%%=====
%% log scale
manageStartYear=1931;
a1=log( sDBB(:,5));
a2=log( sDBB(:,6));
a3= (sDBB(:,7)-manageStartYear);
a4= sDBB(:,8)

A=[a1,a2,a3,a4];
B=sDBB(:,9:10);
C=[A,B];
aDataVector = sDBB(:,1:4);
aDataVector = [aDataVector,C];

X=aDataVector(:,1:8); % variables
Y=aDataVector(:,10); % groups
[d,p,stats] = manova1(X,Y)

%%=====
%%      Boxplot of post-clustering summary
%%=====
%% plot
setDimRow = 0;
setDimCol=0;
clusterNum=size(cs_cell, 2);
setDimRow=size(cs_cell{1},1);
setDimCol=size(cs_cell{1},2);
subFigRow=ceil(clusterNum/3);
if clusterNum<3,
    subFigCol=clusterNum;
else
    subFigCol=3;
end;

for i=1:setDimCol, %variant, i.e. the number of graphs
    %new graph
    % April 22, 2003
    % added same axis scale y for the same variant, instead of autoscale
    % theMaxY, theMinY
    theMinY=99999
    theMaxY=-.99999

```


Appendix K: Source Code

```

for q=1:clusterNum,
    if cs_cell{q}(1,i)<theMinY,
        theMinY=cs_cell{q}(1,i);
    end;
    if cs_cell{q}(2,i)>theMaxY,
        theMaxY=cs_cell{q}(2,i);
    end;
end;
% round to .01
theMinY100=theMinY*100 +0.5;
theMaxY100=theMaxY*100 + 0.5;
theMinY=ceil(theMinY100)/100;
theMaxY=ceil(theMaxY100)/100;
% end of change

F(i)=figure;
for j=1:clusterNum, % the number of subplot on one graph
    subplot(subFigRow, subFigCol, j);
    myStatPlot(cs_cell{j}(1,i), cs_cell{j}(2,i),cs_cell{j}(3,i),cs_cell{j}(4,i),cs_cell{j}(5,i), fundNum(j), j,
theMinY, theMaxY);
    % rotate the graph
    view(90,-90);
end;
end;

%*****
%* This function is used to cluster the SOM visual results based on the colour values.
%* The implementation is based on the following algorithm:
%* 1. Each node is compared with its nearby neighbours.
%* 2. If no neighbour is classified then this node will be assigned to a new cluster.
%* 3. Assign the node to the same cluster as the neighbour node that has the smallest
%*    difference with the node, if the relative difference of the node to the average
%*    value of the cluster is within a given criterion (which determines the fineness
%*    of the clustering level).
%* 4. The node will be assigned to a new cluster if the above condition is not met.
%* Copyright (c): Danyu Xu, 2002, Carleton University
%*****

function [clsInfo, clsNum] = Umatrix_cluster(Umat,threshold)
% the threshold is from 0 to 1, i.e. the diff criteria for the same cluster
% if the relative diff of two nodes (to max value of all nodes) is within threshold, then they in
% the same cluster

% first find the max distance of Umat
maxD=max(Umat(:)) %get the max value
seed=0
clust_num=0
clsInfo=[] % 2 dim array holding cluster number of each node
clsMean=[] % 1 dim array holding the mean value of the cluster
clsNodeNum=[] % 1 dim array holding the node number of the cluster

for i=1:size(Umat,1),
    for j=1:size(Umat, 2),
        disDiff1=99999;
        disDiff2=99999;
        % comp Dij with the average value of the cluster(i,j-1)
        if j-1>=1,

```

Appendix K: Source Code

```
    disDiff1=abs(Umat(i,j)-clsMean(clsInfo(i,j-1)));
end
if i-1>=1,
    disDiff2=abs(Umat(i,j)-clsMean(clsInfo(i-1,j)));
end

if disDiff1==99999 & disDiff2==99999, % i.e. none, so a new cluster
    clsInfo(i,j)=clust_num+1;
    % overall cluster number increase 1
    clust_num=clust_num+1;
    % add to the new cluster
    clsMean(clust_num)=Umat(i,j);
    % the init node number
    clsNodeNum(clust_num)=1;
else
    if disDiff1<=disDiff2,
        if disDiff1/maxD <=threshold,
            % belong to the same cluster as node (i,j-1)
            tmp=clsInfo(i,j-1);
            clsInfo(i,j)=tmp;
            clsNodeNum(tmp)=clsNodeNum(tmp)+1;
            % calculate the new mean value for the cluster
            clsMean(tmp)= (clsMean(tmp)*clsNodeNum(tmp)+Umat(i,j))/(clsNodeNum(tmp)+1);
            % increase the node number by 1
            clsNodeNum(tmp)=clsNodeNum(tmp)+1;
        else
            % a new cluster
            clsInfo(i,j)=clust_num+1;
            clust_num=clust_num+1;
            clsMean(clust_num)=Umat(i,j);
            clsNodeNum(clust_num)=1; % one node in the cluster
        end
    else
        if disDiff2/maxD <=threshold,
            % belong to the same cluster as node (i-1, j)
            tmp=clsInfo(i-1,j);
            clsInfo(i,j)=tmp;
            clsNodeNum(tmp)=clsNodeNum(tmp)+1;
            % calculate the new mean value for the cluster
            clsMean(tmp)= (clsMean(tmp)*clsNodeNum(tmp)+Umat(i,j))/(clsNodeNum(tmp)+1);
            % increase the node number by 1
            clsNodeNum(tmp)=clsNodeNum(tmp)+1;
        else
            % a new cluster
            clsInfo(i,j)=clust_num+1;
            clust_num=clust_num+1;
            clsMean(clust_num)=Umat(i,j);
            clsNodeNum(clust_num)=1; % one node in the cluster
        end
    end
end
end
end
clsNum=clust_num;
return;
```