

30

8 1 6 8 3

U M I
MICROFILMED 2003

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

**Statistical Physics Based
Heuristic Clustering Algorithms
with an Application to Econophysics**

by

Lucia L. Baldwin

A Dissertation Submitted to the Faculty of
The Charles E. Schmidt College of Science
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Florida Atlantic University

Boca Raton, Florida

May 2003

UMI Number: 3081683

**Copyright 2003 by
Baldwin, Lucia Liliana**

All rights reserved.

UMI[®]

UMI Microform 3081683

**Copyright 2003 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.**

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

Copyright by Lucia L. Baldwin 2003

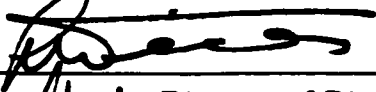
Statistical Physics Based
Heuristic Clustering Algorithms
with an Application to Econophysics


by

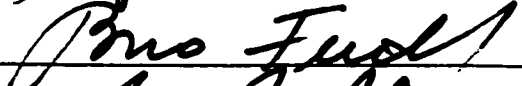
Lucia L. Baldwin


This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Luc T. Wille, Department of Physics, and has been approved by the members of her supervisory committee. It was submitted to the faculty of the Charles E. Schmidt College of Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

SUPERVISORY COMMITTEE:



Director of Dissertation








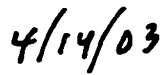
Chairman, Department of Physics



Dean, Charles E. Schmidt College of Science



Division of Research and Graduate Studies



Date

Abstract

Author: Lucia L. Baldwin
Title: Statistical Physics Based
Heuristic Clustering Algorithms
with an Application to Econophysics
Institution: Florida Atlantic University
Dissertation Advisor: Dr. Luc T. Wille
Degree: Doctor of Philosophy
Year: 2003

Three new approaches to the clustering of data sets are presented. They are heuristic methods and represent forms of unsupervised (non-parametric) clustering. Applied to an unknown set of data these methods automatically determine the number of clusters and their location using no a priori assumptions. All are based on analogies with different physical phenomena.

The first technique, named the Percolation Clustering Algorithm, embodies a novel variation on the nearest-neighbor algorithm focusing on the connectivity between sample points. Exploiting the equivalence with a percolation process, this algorithm considers data points to be surrounded by expanding hyperspheres, which bond when they touch each other. Once a sequence of joined spheres spans an entire cluster, percolation occurs and the cluster size remains constant until it merges with a neighboring cluster.

The second procedure, named Nucleation and Growth Clustering, exploits the analogy with nucleation and growth which occurs in island formation during epitaxial growth of solids. The original data points are nucleation centers, around which aggregation will occur. Additional “ad-data” that are introduced into the sample space, interact with the data points and stick if located within a threshold distance. These “ad-data” are used as a tool to facilitate the detection of clusters.

The third method, named Discrete Deposition Clustering Algorithm, constrains deposition to occur on a grid, which has the the advantage of computational efficiency as opposed to the continuous deposition used in the previous method. The original data form the vertexes of a sparse graph and the deposition sites are defined to be the middle points of this graphs edges. Ad-data are introduced on the deposition site and the system is allowed to evolve in a self-organizing regime. This allows the simulation of a phase transition and by monitoring the specific heat capacity of the system one can mark out a “natural” criterion for validating the partition.

All of these techniques are competitive with existing algorithms and offer possible advantages for certain types of data distributions. A practical application is presented using the Percolation Clustering Algorithm to determine the taxonomy of the Dow Jones Industrial Average portfolio. The statistical properties of the correlation coefficients between DJIA components are studied along with the eigenvalues of the correlation matrix between the DJIA components.

Acknowledgements

I am glad to have this opportunity to thank all the professors in the Physics Department. Their kindness and friendly assistance made the time I spent at Florida Atlantic University one of my most pleasant and fulfilling experiences. Dr. F. Medina, Dr. S. Bruenn, Dr. S. Faulkner, Dr. B. Lamborn, and last but not least Dr. R. Jordan marked my memory forever. I hold the same warm and thankful thoughts for the entire staff.

There is one special person I would like to thank above all and that is Dr. Luc Wille. As my advisor he did not spare any time or efforts to help and guide me. His vast knowledge in any area of my interest combined with an infinite patience and the most supportive attitude make him the best advisor anybody could ask for. There are few people in this world with his broad erudition, creative intelligence and such an open mind. My only regrets are that I did not take full advantage of the wisdom he has to offer.

I want to thank the members of my committee, Dr. S. Bruenn, Dr. B. Furht and Dr. V. Jirsa for their kindness in participating in this dissertation defense. I appreciate their time and help.

There are also colleagues and friends I would like to thank for the times of work and leisure we spend together. They all contributed to my success. I want especially to mention Ken DeNisco who, besides his continuously supportive friendship, contributed actively with his great analytic ability to shape some of the thoughts materialized in this dissertation. He went out of his way to provide priceless advise, necessary hardware and software, books

and mainly moral support. I thank Rob Gross for his thoughtful friendship and his help with Mathematica, LaTeX and other software.

Last but not least I want to thank my mother without whom I would not have been able to finish this work and I dedicate this dissertation to her.

Contents

List of Figures	xi
List of Tables	xxiii
1 Introduction	1
1.1 Background	3
1.1.1 Similarity Function	3
1.1.2 Normalization	7
1.1.3 Optimization Criteria	8
1.1.4 Classification of Clustering Techniques	12
1.2 Clustering Techniques Based on Analogies with Physical Phenomena	15
1.3 Data Files	17
2 Percolation Clustering Algorithm	26
2.1 Introduction to Percolation Theory	27
2.2 Description of the Algorithm	36
2.3 Computational Details	40
2.4 Computational Results	43

3	Nucleation and Growth Clustering	57
3.1	Algorithm Description	59
3.2	Computational Details	60
3.3	Computational Results	61
4	Discrete Deposition Clustering Algorithm	68
4.1	Description of the Algorithm	69
4.2	Computational Details	77
4.3	Computational Results	79
5	Application of Clustering to Econophysics	89
5.1	Introduction	89
5.2	Data	91
5.3	Portfolio Taxonomy	99
5.4	Taxonomy of DJIA Portfolio	110
5.5	Statistical Properties of Correlation Coefficients	122
5.6	Properties of Correlation Matrix	135
5.7	The Meaning of the Correlation Matrix Eigenvectors	146
6	Summary and Suggestions for Future Work	162
A	DJIA Components	166
A.1	DJIA Components for 1991	167
A.2	DJIA Components for 2001	168
B	Histograms of Quarterly Correlation Coefficients	169

C Eigenvalue Spectra of Quarterly Correlation Matrices	180
Bibliography	191

List of Figures

1.1	Three two-dimensional clusters illustrating the problem with concavity and interconnectivity.	10
1.2	Data set of 50 two-dimensional sample points grouped in two circular clusters of different densities, set one unit apart. . . .	18
1.3	BWD problem data set consisting of 6000 two-dimensional sample points distributed in three dense regions on a 10 times lower density background.	19
1.4	Histograms of the four attributes for iris flowers: sepal length, sepal width, petal length, and petal width (all measured in cm). 20	
1.5	Projection of the iris data on the plane spanned by petal length and petal width. Open circles correspond to iris setosa, triangles to iris versicolor and filled circles to iris virginica.	21
1.6	Projection of the iris data on the plane spanned by its first two principal components. Open circles correspond to iris setosa, triangles to iris versicolor and filled circles to iris virginica. . .	24

2.1	Seven two-dimensional data points grouped into roughly two clusters of sizes four and three. The first cluster (the left-most one) is completely connected and the second cluster (the rightmost) has not yet linked fully.	37
2.2	<i>Top</i> : Size of the largest (thick line) and second-largest (thin line) cluster as a function of distance between connected points for data set in Figure 1.2. <i>Bottom</i> : Growth rate of largest cluster size as a function of distance between connected points for data set in Figure 1.2.	44
2.3	<i>Top</i> : Size of the largest (continuous thick line), second-largest (thinner line) and third-largest (thinnest dashed line) cluster as a function of distance between connected points for BWD data set. <i>Bottom</i> : Growth rate of largest cluster size as a function of distance between connected points for BWD data set.	46
2.4	<i>Top</i> : Largest cluster size as a function of distance between connected points for the iris problem. Note plateaus and jumps near cluster sizes 50 and 100. <i>Bottom</i> : Rate of variation of largest cluster size as a function of distance between connected points for the iris problem.	49

2.5	<i>Top</i> : Largest cluster size as a function of distance between connected points for the iris versicolor and iris virginica data. Note plateaus in the cluster sizes at 39 and 40 points as well as the jump that follows. <i>Bottom</i> : Rate of variation of largest cluster size as a function of distance between connected points for the iris versicolor and iris virginica data.	50
2.6	Two-dimensional sample points, grouped in two circular clusters of different densities, set two units apart and connected by a bridge.	52
2.7	<i>Top</i> : Largest cluster size as a function of distance between connected points for the two-dimensional set represented in Figure 2.6. <i>Bottom</i> : Rate of variation of largest cluster size as a function of distance between connected points for the data set represented in Figure 2.6.	53
2.8	<i>Top</i> : Largest cluster size as a function of distance between connected points for the two-dimensional set represented in Figure 2.6. <i>Bottom</i> : Rate of variation of largest cluster size as a function of distance between connected points for the data set represented in Figure 2.6.	54
2.9	First three largest cluster sizes as a function of tertiary “distance” between connected points for iris data set.	56

3.1	Snapshots showing the configurations as more and more deposited “ad_data” are introduced, corresponding to the two-dimensional data shown in Figure 1.2.	62
3.2	Size of the largest and second-largest islands as a function of the number of deposited ad_data for the two-dimensional data shown in Figure 3.1.	63
3.3	Size of the largest and second-largest and third-largest clusters as a function of the number of deposited ad_data for the iris data problem.	65
3.4	Size of the largest (solid line) and second-largest (dotted line) clusters as a function of the number of deposited ad_data for the 100 iris versicolor and iris virginica.	67
4.1	The sparse graph between five sample points i, j, k, m, n , represented by black dots, the graph’s edges drawn as solid lines, and the dual sites denoted as, for example, $\{ij\}$ and symbolized as empty circles.	70
4.2	“Snap-shots” of the 50 point two-dimensional data set grouped in two circular clusters and the ad_data points configuration at thermal equilibrium for different temperatures. Notice the number of inter-cluster ad_data points decreasing with decreasing temperature.	82

4.3	Number of <code>ad_data</code> points in cluster 1, (lower thin line), in cluster 2 (upper thick line) and between the clusters (dotted line) for the 50 point two-dimensional toy problem. Notice the relatively constant number of intra-cluster deposition particles up to critical temperature $T_c \simeq 0.18$. The number of inter-cluster <code>ad_data</code> points decreases as the temperature is lowered from $T = 1.5$	83
4.4	Fraction of added (continous line) and deleted (dotted line) <code>ad_data</code> points averaged over 100 attempts before thermal equilibrium for 50 point two-dimensional toy problem.	84
4.5	Fraction of added (continous line) and deleted (dotted line) <code>ad_data</code> points out of 20,000 attempts for 50 two-dimensional points at thermal equilibrium.	85
4.6	Specific heat at constant volume as a function of temperature for the 50 two-dimensional points data set. The specific heat reaches a maximum value of approximately $C_v = 218$ near the critical temperature $T_c \simeq 0.18$ after which it declines abruptly.	86
4.7	“Snap-shots” of the 50 point two-dimensional data set regrouped into two circular clusters with centers placed 3.5 units apart and the <code>ad_data</code> configuration at thermal equilibrium for critical temperature and lowest temperature achived. Notice the low number of inter-cluster <code>ad_data</code> points.	87
4.8	Specific heat at constant volume function of temperature of iris data problem.	88

5.1	DJIA daily closing price during the period 1986-1990. The vertical lines delimit one year from the next.	94
5.2	DJIA daily closing price during the period 1997-2001. The vertical lines delimit one year from the next.	96
5.3	(a) MST and (b) Indexed hierarchical tree obtained during the calendar year 1990 for the portfolio of six companies: CHV, GE, KO, PG, TX and XOM (reproduced after [47]).	108
5.4	Largest and second largest cluster sizes as a function of correlation coefficient during the calendar year 1990 for the portfolio of six companies (CHV, GE, KO, PG, TX and XOM).	109
5.5	The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1986.	111
5.6	The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1987.	112
5.7	The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1988.	113
5.8	The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1989.	114
5.9	The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1990.	115
5.10	The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 1997.	116

5.11	The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 1998.	117
5.12	The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 1999.	118
5.13	The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 2000.	119
5.14	The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 2001.	121
5.15	Histograms of yearly correlation coefficients between 26 major US companies considered representative for the interval 1986-1989.	124
5.16	Histograms of yearly correlation coefficients between the 26 major US companies considered representative for the year 1990.	125
5.17	Histograms of yearly correlation coefficients between the 30 DJIA components for the year 1997.	126
5.18	Histograms of yearly correlation coefficients between the 30 DJIA components for the period 1998-2001.	127

5.19	<i>Top</i> : Histograms of correlation coefficients between shuffled time series of daily logarithmic price variation for DJIA stocks during 2000 and respectively 2001. <i>Bottom</i> : Histograms of correlation coefficients between 30 series of 250 normal distributed random numbers and 100×30 series of 250 normal distributed random numbers.	128
5.20	Average value of quarterly correlation coefficients between 26 major US companies for the interval 1986-1990. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.	130
5.21	Average value of quarterly correlation coefficients between DJIA components for the interval 1997-2001. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.	130
5.22	Standard deviation of quarterly correlation coefficients between 26 major US companies for the interval 1986-1990. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.	131
5.23	Standard deviation of quarterly correlation coefficients between DJIA components for the interval 1997-1999. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.	131

5.24	Kurtosis of quarterly correlation coefficients between 26 major US companies for the interval 1986-1990. On the abscissa each unit represents one year and the data points are placed at the end of each quarter.	133
5.25	Kurtosis of quarterly correlation coefficients between DJIA components for the interval 1997-2001. On the abscissa each unit represents one year and the data points are placed at the end of each quarter.	133
5.26	Eigenvalue spectra (solid lines) of infinite random matrices for the cases: $Q = 2$ (widest), $Q = 8.33$ (middle) and $Q = 41.7$ (narrowest and highest) and normalized eigenvalue histograms (dotted lines) for 10,000 random 30×30 matrices of the same parameters Q	138
5.27	Eigenvalue spectra of the annual correlation matrices for the 26 major US companies studied during the years 1986-1989.	140
5.28	Eigenvalue spectrum of the annual correlation matrix for the 26 major US companies studied during the year 1990.	141
5.29	Eigenvalue spectra of the annual correlation matrices for the 30 DJIA components studied during the years 1997-2000.	142
5.30	<i>Left:</i> Eigenvalue spectrum of the annual correlation matrix for the 30 DJIA components studied during the year 2001. <i>Right:</i> Eigenvalue spectrum of the annual correlation matrix for the 30 DJIA components studied during the year 2002.	143

5.31	<i>Left:</i> Eigenvalue spectrum of the five-year correlation matrix for the 26 major US companies studied during 1986-1990. <i>Right:</i> Eigenvalue spectrum of the five-year correlation matrix for the 30 DJIA components analyzed during 1997-2001. . . .	144
5.32	<i>a:</i> First eigenvector components for two random matrices. <i>b, c, d, e, f:</i> First eigenvector components for yearly correlation matrix during the years 1998 through 2002 as a function of the similar components during the previous years.	147
5.33	Tenth eigenvector components for the years 1998, 2000 and 2002 a function of the similar components during the previous years.	148
5.34	<i>a, b, c, d, e, f:</i> Second eigenvector components for yearly correlation matrix during the years 1998 through 2002 as a function of the similar components during the previous years. .	153
5.35	Second eigenvector projections along the 30 DJIA components for the yearly correlation matrix during the years 1998 (solid line) and 1999 (dotted line). The components for the year 1998 are represented with an inversed sign.	155
5.36	Second eigenvector projections along the 30 DJIA components for the yearly correlation matrix during the years 2000 (solid line) and 2001 (dotted line).	156

5.37	Second eigenvector projections along the 30 DJIA components for the yearly correlation matrix during the years 2001 (solid line) and 2002 (dotted line). The components for the year 2002 are represented with an inversed sign.	157
B.1	Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1986.	170
B.2	Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1987.	171
B.3	Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1988.	172
B.4	Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1989.	173
B.5	Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1990.	174
B.6	Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 1997.	175
B.7	Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 1998.	176
B.8	Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 1999.	177
B.9	Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 2000.	178

B.10	Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 2001.	179
C.1	Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1986.	181
C.2	Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1987.	182
C.3	Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1988.	183
C.4	Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1989.	184
C.5	Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1990.	185
C.6	Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 1997.	186
C.7	Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 1998.	187
C.8	Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 1999.	188
C.9	Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 2000.	189
C.10	Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 2001.	190

List of Tables

1.1	A set of N sample points with an identical number of attributes D in a standard spreadsheet format.	4
1.2	Mean and variance of the four iris data attributes. The total variance is 4.545 cm^2	22
1.3	Eigenvalues and variances of the four zero-mean principal components for iris data. The total variance is 4.545 cm^2	23
2.1	Data distribution and the results of two clustering algorithms for the BWD problem.	48
2.2	Results of different clustering algorithms for iris data. Note the similar partitions produced by Percolation Clustering Algorithm and Superparamagnetic Clustering procedure.	51
4.1	Dual site arrangement and ad_data point repartition at two different temperatures for the 50 point two-dimensional data set.	80

5.1	DJIA daily closing price for the first and last trading day of each quarter, as well as quarterly and annual percentage variation of these prices during the interval 1986-1990.	95
5.2	DJIA daily closing price for the first and last trading day of each quarter, as well as quarterly and annual percentage variation of these prices during the interval 1997-2001.	97
5.3	Number of analyzed days for each quarter and year in the chosen time intervals, 1986-1990 and 1997-2001.	98
5.4	Distance matrix of the six stocks identified by their ticker symbols for the year 1990.	106
5.5	Correlation matrix of the six stocks identified by their ticker symbols for the year 1990.	107
5.6	Ordered correlation coefficients and distances between six stocks designated by their ticker symbols for the year 1990.	107
5.7	Average correlation coefficients over all thirty assets, $\bar{\rho}$, their standard deviations, σ , the largest eigenvalue Λ_1 , calculated according to equation (5.24) and the empirical largest eigenvalue, λ_1 , for the years 1986-1990 and 1997-2002.	151
5.8	Numbers of significant components for the first the second eigenvectors as well as the average over all eigenvectors of the annually correlation matrix during the interval 1997-2002. . .	158

5.9	Second eigenvector inverse participation ratio, I_2 , the average inverse participation associated to each component, I_2^{avg} , and the minimum value of the projection considered significant $ v_{2j} _{min}$ for the interval 1997-2002.	158
5.10	Significant components of the second eigenvector for the interval 1997-2002.	159
A.1	Dow Jones Industrial Average components during 1991 listed in the order of their ticker symbol, together with the primary group they belong to.	167
A.2	Dow Jones Industrial Average components during 2001 listed in the order of their ticker symbol, together with the primary group they belong to.	168

Chapter 1

Introduction

Today's interest in data mining is highly motivated by the explosion of data flooding the world, overwhelming individuals as well as large organizations. All quantitative sciences gather their forces in the quest of leveraging the "Big Bang" of data content into structured information and useful knowledge. Undoubtedly computer development fueled this explosion and at the same time offers the assistance needed to master it.

The challenge is that, although machines outperform the human brain when it comes to simple repetitive operations, enabling artificial systems to process data is not a trivial job. The brain can deal with fuzzy, noisy and even inconsistent information. It is flexible, robust and fault tolerant [1]. Even simple biological creatures perform fundamental tasks such as perception, classification and recognition. Training computers to mirror this biological performance requires addressing the problems in a rigorous, formal manner.

Physics enriches us with an impressive library of concepts and mathe-

mathematical tools whose highly abstract forms make them applicable to a wide variety of problems. The similarity between subjects is also encouraging. For instance, statistical physics studies systems with a very large number of elements where stochastic (thermal) fluctuations generate macroscopic effects. Data mining addresses the problem of revealing the structures and correlations hidden by noise in large data sets. Therefore, in recent years there has been an increased interest in adapting numerical and analytic techniques from statistical physics to different areas of data mining.

The first step in the cognitive process when we have little a priori knowledge about the structure of data or the information we are looking for is to cluster data: divide the data set into a small number of subgroups (clusters) in such a way that the elements within the same subgroup are more similar to each other than to elements from all other different subgroups [2]. Clustering can be considered the archetypical data mining problem, closely related to unsupervised learning and pattern recognition.

Once the clustering problem is defined, a number of fundamental issues arise. We have to describe a measure of similarity (or dissimilarity) between sample points as well as a criterion to choose the best partitioning of the data set. These tasks are inherently related to calculations combining components of data points and involve underlying issues of measurement theory. General aspects of the clustering procedures, their classification, as well as the major directions engaged today are presented in the following sections. The last part of this introductory chapter describes the main data files used throughout the dissertation.

The following chapters have a more definite aim, *i. e.* to introduce three new clustering techniques. The heuristic approach is based on analogies to well-studied physical processes. The algorithms have the advantage of simplicity and computational efficiency over existing methods and provide competitive results.

1.1 Background

Clustering, being an attempt to partition data sets into groups of similar points, becomes a well-defined problem once two functions are chosen: the similarity (or dissimilarity) between two sample points and the criterion (cost function) used to evaluate different partitions [2]. Each of these two functions can be selected in many different ways according to the nature of the data and any a priori knowledge we have about it.

1.1.1 Similarity Function

I will start by discussing the first of these two issues: the measure of similarity between two sample points. The only constraint on this binary similarity function is that it has to be symmetric $s(i, j) = s(j, i)$. Consider a set of N sample points with an identical number of attributes D in a standard spreadsheet set-up, as presented in Table 1.1. All attributes have to be converted into features by encoding them in a numerical format. A data point can be seen as a vector in a D -dimensional space, where the coordinates correspond to its features. Before attempting to combine the vector coordinates, we

Sample Point	Attribute ₁	...	Attribute _k	...	Attribute _D
\mathbf{x}_1	x_{11}	...	x_{1k}	...	x_{1D}
\vdots	\vdots				\vdots
\mathbf{x}_i	x_{i1}	...	x_{ik}	...	x_{iD}
\vdots	\vdots				\vdots
\mathbf{x}_N	x_{N1}	...	x_{Nk}	...	x_{ND}

Table 1.1: A set of N sample points with an identical number of attributes D in a standard spreadsheet format.

should be aware that the space is not necessarily continuous. The *categorical* features (such as color, sex, group membership, etc.), even under numerical format, cannot be ordered. In a spreadsheet format they are represented as numbers using a specified code, but one cannot define a distance between unordered numerical values. In such cases, one can choose a non-metric similarity function between two sample points, \mathbf{x}_i and \mathbf{x}_j as, for example, the normalized inner product of the two vectors:

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

Such a similarity measure, or variations of it, are frequently used in biological taxonomy or information theory [2].

This dissertation focuses on data sets in which all feature values can be ordered, which means that for two different values, a_1 and a_2 , of any attribute a one can always define the operators $a_1 > a_2$ or $a_1 < a_2$. Therefore, a norm,

or distance, can be defined in feature space and can be used as a similarity measure, s . For two sample points, \mathbf{x}_i and \mathbf{x}_j , s becomes

$$s_{\mathbf{x}_i, \mathbf{x}_j} = \|\mathbf{x}_i - \mathbf{x}_j\|. \quad (1.1)$$

The similarity function (distance) between two vectors described by equation (1.1) should be invariant to transformations natural to the problem. Thus the metric has to be selected based on previously known properties of the data. Consider two vectors \mathbf{x}_i and \mathbf{x}_j with attributes x_{ik} and x_{jk} ($k = 1, \dots, D$), respectively, in a uniform isotropic feature space. In this case the binary Euclidean distance:

$$d_{ij}^E = \|\mathbf{x}_i - \mathbf{x}_j\|_E = \sqrt{\sum_{k=1}^D (x_{ik} - x_{jk})^2} \quad (1.2)$$

is a good choice for a similarity measure, since it is invariant to translations and rotations of the coordinates. However, we can not always assume the isotropy of feature space. Imagine, for example, that data points represent individuals whose characteristics are height, income, blood pressure, number of children, etc. A rotation of the original coordinates would generate axes with no real meanings, a dissimilarity along any of these axes does not correspond to any easily interpreted difference between individuals. When we have little a priori knowledge about the properties of sample attributes, an unassuming way of combining vector coordinates could be advantageous. For instance, the Manhattan distance defined as:

$$d_{ij}^M = \|\mathbf{x}_i - \mathbf{x}_j\|_M = \sum_{k=1}^D |x_{ik} - x_{jk}| \quad (1.3)$$

can then be used as a similarity function.

In a low-dimensional sample space both metrics usually generate basically equivalent results for well defined clusters. The differences between these norms, as in fact differences between any other norms, increase with the dimensionality of the sample points.

Since we tested our algorithms on low-dimensional data files, the Manhattan distance is as good as any other norm and for computational efficiency reasons we prefer it since it is faster to evaluate. Throughout the rest of the dissertation we refer to it under the simplified notation d_{ij} , unless otherwise specified. The Euclidean norm was used to study the robustness of some of the partitions obtained with the Manhattan distance.

No matter which metric is used to measure the similarity between two vectors, it involves combining the values of their coordinates. This raises a scaling problem, since features of different magnitudes should have equivalent weight in calculating a binary distance. Some norms, like the Euclidean distance, for instance, are invariant to rotations and translations but sensitive to other linear transformations. The Manhattan distance, on the other hand, is susceptible to any linear transformation of the coordinates. A general distortion of feature space is not a concern unless it is natural to the problem. However, a change in the units of measured values should have no influence on clustering results. This is why in many cases data normalization is required before any clustering procedures are performed.

1.1.2 Normalization

By normalization we understand the scaling of all features to a specified range (usually $[0, 1]$ or $[-1, 1]$), so that all of them have equivalent weights. Several normalization techniques are presented below.

Consider a set of N sample points where the feature x_k has the values: x_{1k}, \dots, x_{Nk} and let $x_{k|max}$, $x_{k|min}$, \bar{x}_k and s_k be the maximum, minimum, mean and standard deviation, respectively, of this feature. The new scaled values x'_{ik} corresponding to x_{ik} can be defined as:

$$x'_{ik} = \frac{x_{ik} - x_{k|min}}{x_{k|max} - x_{k|min}}. \quad (1.4)$$

The transformation expressed by (1.4) is called range normalization and it changes the feature's span from $[x_{k|min}, x_{k|max}]$ to $[0, 1]$. Another option, is to convert all values such that $|x'_{ik}| \leq 1$ by:

$$x'_{ik} = \frac{x_{ik}}{|x_{jk|max}|}, \quad (1.5)$$

where $|x_{jk|max}|$ is the maximum absolute value of attribute k over all sample points. Alternatively, the scaling:

$$x'_{ik} = \frac{x_{ik} - \bar{x}_k}{\bar{x}_k} \quad (1.6)$$

makes all the normalized values have zero mean.

Since the minimum, maximum and average values are drastically affected by outliers, *i. e.* the elements in the tails of the distribution, a better choice is a percentile normalization, as given, for example, by:

$$x'_{ik} = \frac{x_{ik} - x_{k|2.5}}{x_{k|97.5} - x_{k|2.5}}, \quad (1.7)$$

where $x_k|_{2.5}$ and $x_k|_{97.5}$ are the 2.5 and the 97.5 percentiles, respectively. This conversion scales data so that 95% of the values are between 0 and 1.

The most common normalization option is to make all features have zero-mean and unit variance:

$$x'_{ik} = \frac{x_{ik} - \bar{x}_k}{s_k}. \quad (1.8)$$

This way one can easily compare the distribution of the x_k feature to the unit normal distribution.

Normalization, like any data transformation, corrupts to a certain degree the original data configuration. Therefore this procedure has to match the objective of the study. For example formulas (1.4), (1.7) or (1.8) give comparable weight and similar variability to all features and make them suitable for the feature composition used to calculate a norm. Reducing all features' standard deviations to a similar range might diminish the natural difference between groups of sample points[2]. When data variability is under study, we need to preserve it throughout normalization, hence equations such as (1.5) and (1.6) are preferable alternatives.

1.1.3 Optimization Criteria

Once a distance measure is defined, the second important issue of any clustering technique is to select the optimization criterion (cost function) used to evaluate the best partitions of the sample points. This is usually a similarity function between sets of points and is based on a binary similarity function between sample points. Having a metric described by equation (1.1), several

examples of cost functions to be minimized (or maximized) during the clustering procedure are presented below. Given two sets, \mathcal{H}_1 and \mathcal{H}_2 , of vector points $\{\mathbf{x}_1, \dots, \mathbf{x}_{n_1}\}$ and $\{\mathbf{x}'_1, \dots, \mathbf{x}'_{n_2}\}$, respectively, two possible criterion functions are: the average distance

$$d_{avg}(\mathcal{H}_1, \mathcal{H}_2) = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \|\mathbf{x}_i - \mathbf{x}'_j\| \quad (1.9)$$

and the minimum distance between the two sets

$$d_{min}(\mathcal{H}_1, \mathcal{H}_2) = \min_{\mathbf{x} \in \mathcal{H}_1, \mathbf{x}' \in \mathcal{H}_2} \|\mathbf{x} - \mathbf{x}'\|. \quad (1.10)$$

One of the difficulties encountered in clustering problems is that a similarity function (distance) between two points can hardly account simultaneously for the closeness and the interconnectivity among sets of points. Some cost functions emphasize the closeness between groups of points, others stress the connectivity between classes. Optimization criteria based on cumulative distances, such as the one in equation (1.9), reflect well the interconnectivity between sample points, but tend to favor larger data groups and fail for sets which contain large variations in cluster sizes. They also give erroneous results for concave shaped clusters, such as shown in Figure 1.1, *i. e.* when points in a given cluster are closer to points in another cluster. For example, the upper cluster of Figure 1.1, contains the points C, D, and E out of which C and E are farther apart from each other than point A is from D. Therefore a clustering algorithm based on cumulative distances might incorrectly partition the upper and middle clusters. The same statement is true if one considers the points F and H with respect to points B and G with the above

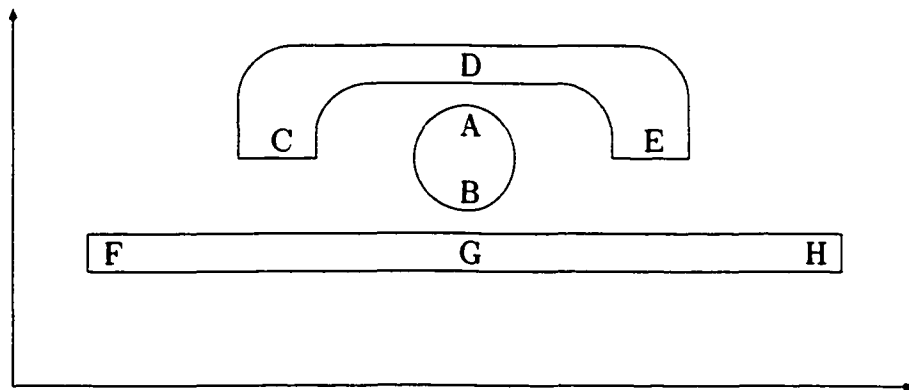


Figure 1.1: Three two-dimensional clusters illustrating the problem with concavity and interconnectivity.

argument. In this type of situation, an optimization criteria which emphasizes connectivity more than closeness would find the correct partitioning of the data.

Methods based on the cost function described by equation (1.10) define the similarity between two clusters as the similarity of the closest pair of points belonging to the different clusters. They underline the closeness between groups and can find clusters of arbitrary shapes and sizes but are highly susceptible to noise and outliers. Improved performance can be achieved when a filtering procedure is performed before clustering. By filtering we understand eliminating the background points and keeping only the data that have an average distance to the first k nearest neighbors smaller than a user defined threshold. Nevertheless these techniques fail to correctly partition the data sets that contain clusters of different densities. Other methods

strive to obtain correct results by normalizing each binary distance via a specific local length whose value is defined based on the local mean k nearest neighbor distance.

Some new techniques define elaborate dynamic similarity functions. For example, Chameleon [5], a hierarchical agglomerative method, uses a similarity function between sets of points that accounts simultaneously for interconnectivity and closeness. The algorithm starts by building a sparse graph between data points in such a way that data items that are far apart are completely disconnected. Each edge of the graph is weighted with the similarity between the two connected data points. An absolute internal interconnectivity, EC , is defined for each group of points as the sum of all edge weights crossing the mid-cut bisection that splits the cluster into two roughly equal parts. Considering two groups of points C_i and C_j , the relative interconnectivity, RI , between them is defined as the absolute interconnectivity normalized by the average internal interconnectivity of the two groups:

$$RI(C_i, C_j) = \frac{|EC(C_i, C_j)|}{\frac{1}{2}(EC(C_i) + EC(C_j))}. \quad (1.11)$$

$EC(C_i, C_j)$ is the absolute interconnectivity between the two clusters, defined as the sum of all the edge weights that connect the two groups, and $EC(C_i)$ and $EC(C_j)$ are the internal interconnectivity of each group respectively.

Also, an absolute closeness, SEC is defined for each group as the average of the edge weights that cross the mid-cut bisection (as opposed to the sum of the edge weights for interconnectivity). A relative closeness, RC , for the

clusters C_i and C_j , is defined as:

$$RC(C_i, C_j) = \frac{SEC(C_i, C_j)}{\frac{|C_i|}{|C_j|+|C_j|}SEC(C_i) + \frac{|C_j|}{|C_i|+|C_j|}SEC(C_j)}, \quad (1.12)$$

where $SEC(C_i, C_j)$ is the absolute closeness described as the average weight of all the vertices that connect the two clusters. $|C_1|$ and $|C_2|$ are the total number of vertices in each cluster, and, correspondingly, $SEC(C_i)$, $SEC(C_j)$ are the internal closeness for the two groups. Using a normalized interconnectivity and closeness between the two merged groups of points accounts for the nature of each individual cluster and makes the similarity function between sets a dynamic one. Chameleon selects a pair of clusters to be linked by maximizing the function $RI(C_1, C_2) \times RC(C_1, C_2)^\alpha$ where α is a user specified value. When $\alpha > 1$ a higher significance is given to the relative closeness and for $\alpha < 1$ the relative interconnectivity is emphasized. Good results have been reported for image processing, except that the dynamic modeling of cluster similarity is applicable only when each cluster contains a large number of items, such that the quantities defined by equations (1.11) and (1.12) can be properly determined.

1.1.4 Classification of Clustering Techniques

In spite of their variety, clustering techniques can be divided in two categories: non-hierarchical and hierarchical. The first type of approach starts by subjectively partitioning the N sample points among a given number of C clusters. The members of the clusters are later redistributed according to an iterative optimization of the appropriately chosen cost function. On

the other hand, hierarchical clustering techniques group the points according to a tree-like scheme: whenever two sample points are assigned to the same cluster at a given level, they will remain together at all parent clustering levels.

An example of a non-hierarchical approach is the k-means method. It was proposed by MacQueen [3] as a simple heuristic method for performing clustering. It is an iterative procedure, which starts with an initial partitioning of the dataset into k clusters either based on a priori information regarding the data or else randomly selected. Next, the *centroids* (average positions) of each cluster are calculated, data points are reassigned to the centroid that is closest to them and a new set of centroids is calculated. Other variations of the method replace the abstract centroid point by the *medoid* which is the data point closest to the center of the cluster. The procedure is repeated until no more reassignments occur. The clustering criterion, which is the function being optimized, is the sum of the distances between each element and the nearest centroid (or medoid). This kind of aggregate cost function is similar to the type described by equation (1.9) and suffers the same limitations. The k-means algorithm is a form of hill-climbing, since it starts with a certain configuration that it systematically improves until no further improvements are possible by small changes. In practice this algorithm performs well for clusters that are hyper-ellipsoidal and have similar sizes, but it cannot find concave shapes or groups of very diverse size. There are two other key drawbacks to this approach. First, the number of clusters must be known in advance, or else the algorithm must be run for different k -values

and a choice between the various clusterings must be made. The second drawback is that there is no guarantee that the algorithm will converge to the global minimum, but instead, like all hill-climbing techniques, it may lead to a local minimum [4], depending on the initial partitioning.

Regarding the hierarchical clustering algorithms there are two methods: divisive and agglomerative. The first category begins by placing all N data points in one cluster and then, following the chosen optimization criterion, splits the cluster in two, three and so on, up to the desired number of clusters. Alternatively, agglomerative methods start with a number of clusters equal to the number of sample points and successively merge the clusters [2]. The hierarchical agglomerative clustering technique that uses equation (1.10) to find the nearest cluster to be merged is called the nearest-neighbor [2] or single link [5] algorithm.

An important observation, regarding all the methods mentioned above, is that they have a parametric modus operandi, in the sense that the number of clusters and even their locations are prior knowledge included as initial parameters in these clustering procedures. Clearly, for an unknown data set, a desirable algorithm is one that provides a “natural”, non-parametric way of partitioning the sample, based solely on the inner structure of the data. There are exhaustive methods that effectively search through the entire solution space and are therefore guaranteed to find the global optimum, but such methods tend to be very time-consuming, since they involve essentially an exponential search [6], and they are consequently only applicable to small data sets.

1.2 Clustering Techniques Based on Analogies with Physical Phenomena

Several techniques have been proposed over the years to avoid the local minimum problem. Some of these are based on the idea of simulated annealing [7, 8] while others use genetic algorithms [9, 10, 11]. All have the core idea of allowing “uphill” moves so that the search procedure may escape from local minima and eventually end up in the global minimum.

Here we are particularly interested in two heuristic methods that perform clustering by exploiting analogies from statistical physics. The first one, which borrows ideas from simulated annealing [7], is the Super-Paramagnetic Clustering method of Domany and collaborators [12, 13, 14, 15], for which they were granted a US patent [16]. The central idea of this method is to mimic, with methods from statistical physics, a cooling process to allow the data points to group themselves in clusters as the temperature is lowered. To this end one associates with each data point a Potts spin [17], a vector that can be pointing in any one of q directions. Here q should be chosen larger than the number of clusters present. A group of adjacent parallel Potts spins forms a cluster. The spins interact with each other like tiny magnets, with the interaction strength increasing with decreasing distance. Thus nearby spins will tend to align with each other to minimize their energy. However, at finite temperature this ordering tendency is offset by entropic effects which will tend to destroy the order. By starting the simulation at high temperature (disordered spins, no clusters) and slowly cooling (using a

Metropolis algorithm [18] one finds that the system undergoes a number of phase transitions. At each phase transition temperature a cluster of spins “snaps” into alignment. By monitoring the number of aligned spins and various statistical fluctuation functions, one is able to detect the number of clusters and their location. This method has been applied to a wide range of test data and performs very well in practice [12, 13, 14, 15]. It improves over the k-means method in that the number of clusters is obtained automatically as it emerges during the cooling process and that trapping in local minima does not occur if the cooling proceeds slowly enough. However, the calculations tend to be quite time consuming and may necessitate some fine-tuning before the optimal parameters (initial temperature, interactions, cooling schedule, etc.) can be determined.

Another recent physical approach is also based on putting physical objects at the data points and exploiting emergent collective behavior. This is the inhomogeneous chaotic map method of Angelini et al. [19]. In this technique, a chaotic map is associated with each data point and short-range interactions between data points are introduced, with coupling strength decreasing with distance. It is known that such chaotic maps, when coupled together, tend to synchronize their behavior. The maps are iterated over time until they reach a stationary regime, which can be shown to be independent of the initial conditions since it is a macroscopic attractor. In this regime one can determine the mutual information between the various maps, which is a measure of the amount of correlation between them. From this mutual information function, the clusters are identified as follows. A graph is constructed by

linking any two data points whose mutual information is above some threshold value. The clusters then correspond to the linked components of this graph, which can be easily determined. By varying the threshold value one can perform hierarchical clustering. This method was tested on a number of problems and was found to perform quite well. Slightly better performance than superparamagnetic clustering was claimed in certain cases. Moreover, the method does not involve a sweep over temperature which means it is considerably faster than techniques based on an annealing scheme. This appears to be quite a promising technique although further test cases will have to be studied to judge its efficacy on a wide range of applications. There is still some fine-tuning of parameters involved, which may be a drawback in complex situations. It is to be noted that in this method too, the number of clusters emerges from the calculation, rather than being fixed in advance and that trapping in local minima does not occur due to the independence of the final state of the initial conditions.

1.3 Data Files

Before presenting the new clustering algorithms which form the core topic of this dissertation, I will take a moment to describe the main data files used to develop and test our techniques. There are several other files employed throughout the report, but the following ones are repeatedly referred to. In order to avoid redundancies, they are presented only once in this section.

The first example is a simple two-dimensional “toy” problem whose pur-

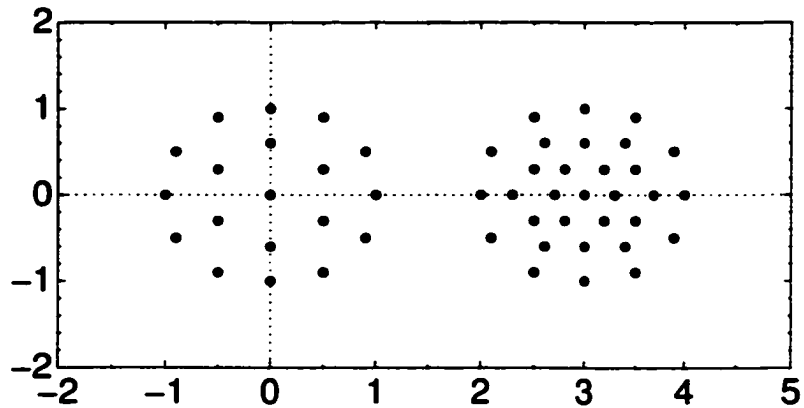


Figure 1.2: Data set of 50 two-dimensional sample points grouped in two circular clusters of different densities, set one unit apart.

pose is to illustrate the concept and general algorithm behavior. The data file consists of 50 sample points grouped into two circles of equal radii ($R = 1$) and different densities. The smaller cluster, centered at the origin, contains 19 points and the larger one 31 points, as shown in Figure 1.2.

To test our algorithm on sensibly complicated examples we used a second set of 6000 points. The original file, obtained courtesy of Dr. Marcelo Blatt, contains two-dimensional sample points distributed in three irregularly shaped dense areas on a diluted background, as presented in Figure 1.3. All dense regions have the same uniform distribution, which is 10 times larger than the density of the background. The data, crafted as a test for the Super-Paramagnetic Clustering procedure, was used for the first time in the article by Blatt et al. [12]. Considering its provenance, I will refer to this

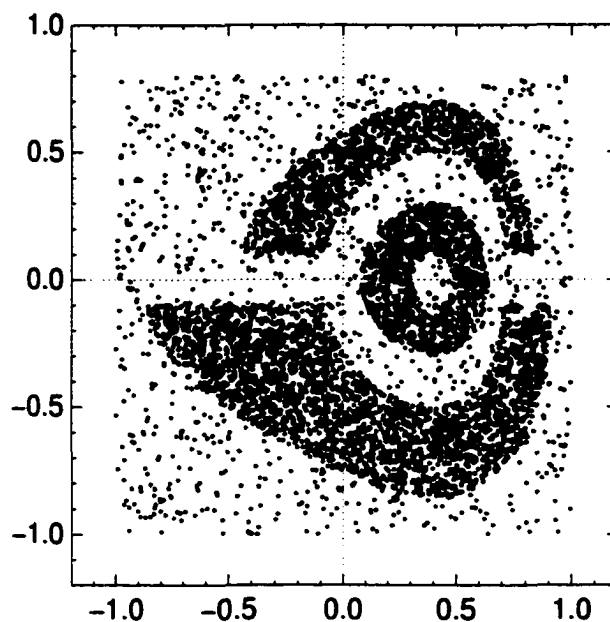


Figure 1.3: BWD problem data set consisting of 6000 two-dimensional sample points distributed in three dense regions on a 10 times lower density background.

data as the BWD problem. The configuration presented in Figure 1.3 is a typical example of concave clusters. The sample points in one group are closer to sample points belonging to other groups than to some in their own category. For example, the points in the upper semi-circle of the inner ring are closer to the lower limit of the upper cluster than to the points on the lower semi-circle of the ring. Our methods, as will be seen, build the clusters based on local configuration of the data, hence the partitions correspond to natural classes.

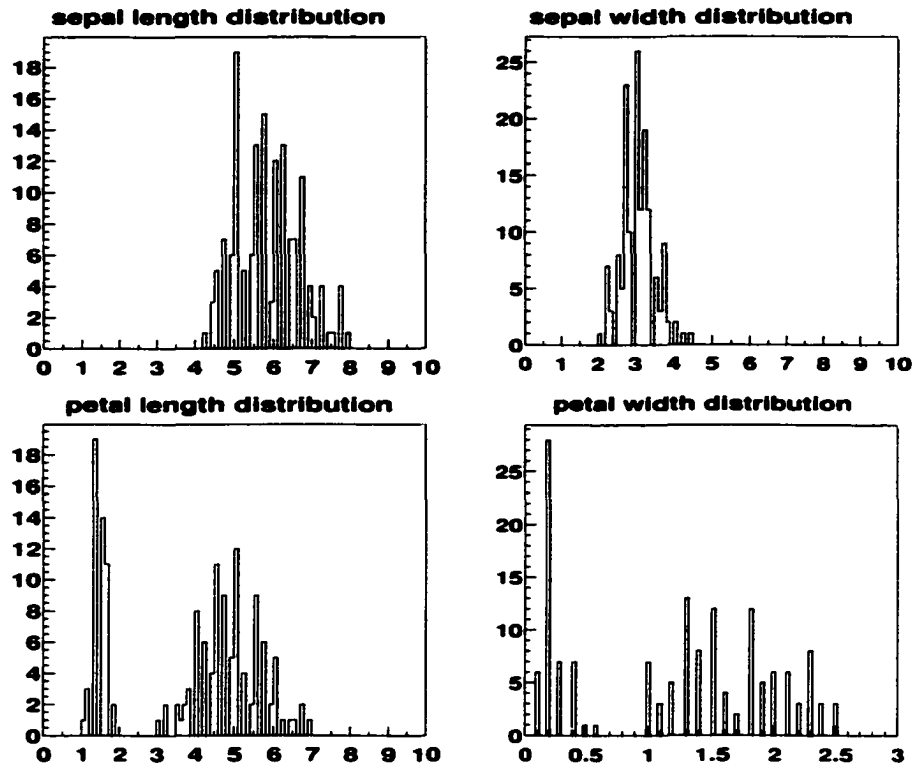


Figure 1.4: Histograms of the four attributes for iris flowers: sepal length, sepal width, petal length, and petal width (all measured in cm).

To confront our algorithms with a “real life” problem, we have studied the well-known iris data problem, perhaps the most famous test case in data mining [20]. This data set was assembled by the statistician R. A. Fisher in the 1930’s. The file is available in the Repository of Machine Learning Databases at the University of California, Irvine, Department of Information and Computer Science[21]. It contains fifty examples each of three types of

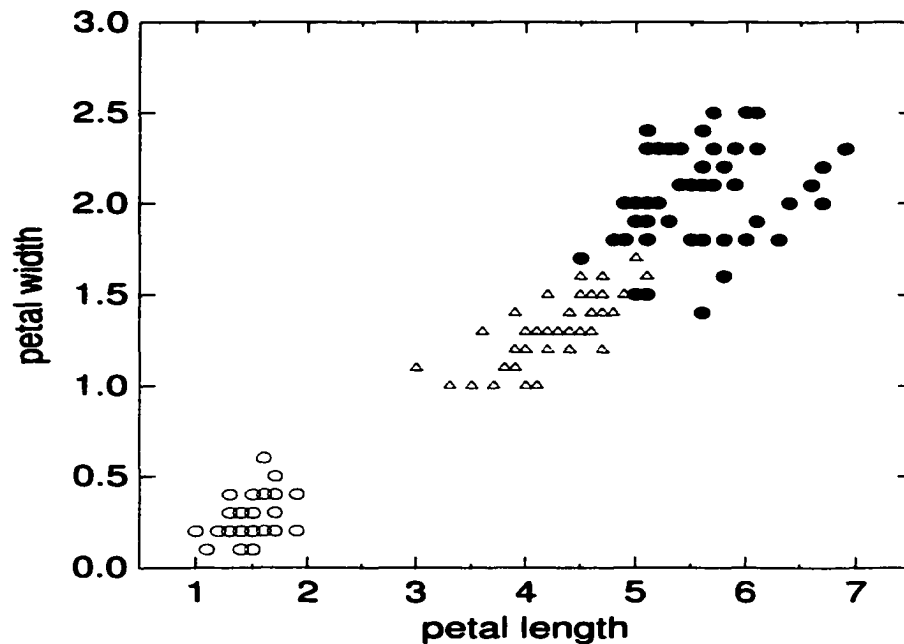


Figure 1.5: Projection of the iris data on the plane spanned by petal length and petal width. Open circles correspond to iris setosa, triangles to iris versicolor and filled circles to iris virginica.

flowers: Iris setosa, Iris versicolor, and Iris virginica, for a total of 150 data points. Each data point is characterized by four attributes, all expressed in centimeters: sepal length, sepal width, petal length, and petal width. Thus this is a four-dimensional clustering (or classification) problem and not so easily visualized as the two-dimensional cases illustrated previously. To better understand the data configurations we start by inspecting the histograms of the four attributes, shown in Figure 1.4. Clearly, the first two variables do not reveal the existence of an easily detectable structure in the

Attribute	Mean (cm)	Variance (cm ²)	fraction of total variance
sepal length	5.84	0.68	15%
sepal width	3.05	0.19	4%
petal length	3.76	3.09	68%
petal width	1.20	0.58	13%

Table 1.2: Mean and variance of the four iris data attributes. The total variance is 4.545 cm²

data, only the last two components, petal length and petal width, determine a visible separation between groups. Therefore a two-dimensional projection of the iris data on the plane spanned by the attributes petal length and petal width, shown in Figure 1.5, is a helpful tool in visualizing the structure of the data set. One can see how the 50 iris setosa points form a well separated group, while the other 100 flowers form two overlapping clusters. While examining the histograms of sample point attributes might be a good idea for low-dimensional data sets, the method becomes ineffective for data points with numerous components. In these situations, a dimensionality reduction technique could be a good start in understanding the data configuration.

Using the Principal Component Analysis for the iris data problem, we reached the same conclusion, as stated above, that this 4-dimensional data set has an intrinsic dimensionality of only 2. As is known, the main pur-

Component	Corresponding eigenvalue	Variance (cm ²)	fraction of total variance
principal comp.	0.728	4.22	93%
second comp	0.230	0.24	5.3%
third comp.	0.037	0.08	1.8%
forth comp.	0.005	0.005	0.5%

Table 1.3: Eigenvalues and variances of the four zero-mean principal components for iris data. The total variance is 4.545 cm².

pose of PCA is to define a new coordinate system able to reflect the total variance of the original data set in such a way that each new component accounts for less and less of the total variance [22]. More explicitly, given N vectors, characterized in D -dimensional sample space by the coordinates x_1, x_2, \dots, x_D , the variance (second moment) of each attribute x_k is:

$$Var(x_k) = \frac{1}{N} \sum_{i=1}^N (x_{ik} - \mu_k)^2, \quad (1.13)$$

where μ_k is the mean (first moment) of attribute x_k . The total variance of the data set is the sum of all coordinate variances:

$$Var = \sum_{k=1}^D Var(x_k) = \frac{1}{N} \sum_{k=1}^D \sum_{i=1}^N (x_{ik} - \mu_k)^2. \quad (1.14)$$

This variance represents the dispersion of data points from the center of symmetry of the data set. The new system of coordinates has to preserve this

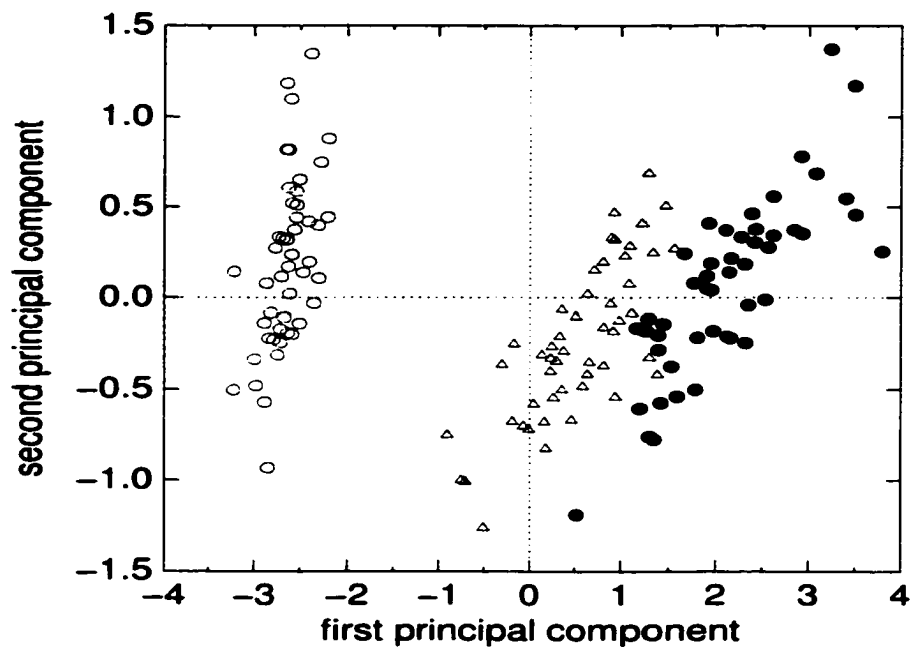


Figure 1.6: Projection of the iris data on the plane spanned by its first two principal components. Open circles correspond to iris setosa, triangles to iris versicolor and filled circles to iris virginica.

distance measure, hence it is obtained merely by a rotation (linear combination) of the original variables. The new coordinates we are looking for are the eigenvectors of the covariance matrices (rotation conserves the trace) and the corresponding eigenvalues are a measure of the magnitude of the variance expressed along each eigenvector. Table 1.3 presents the variance of the iris data along the four original attributes as well as the fraction of total variance expressed by each component. The total variance of 4.54 cm^2 can also be reflected in the principal component representation. Table 1.3 lists the

variance of the iris data set along the four zero-mean principal components.

As one can see the first two eigenvalues are one to two orders of magnitude larger than the others, hence the corresponding eigenvectors (first two principal components) reflect most of the variability of the iris data. Larger variability along one component increases the chance, but does not guarantee, that the component contains a more noticeable differentiation between clusters. In Table 1.3 one can see that sepal length has a larger variance than petal width, while the histograms in Figure 1.4 show a better differentiation between groups along petal width. Projection of the iris data on the plane defined by the first two principal components, presented in Figure 1.6, shows the well defined iris setosa group and the overlapping clusters corresponding to iris versicolor and iris virginica. The iris data file is a benchmark for classification problems more than clustering problems due to the intertwined groups of iris versicolor and iris virginica. Nevertheless, using this data file as a test for a clustering procedure provides a verification of the clustering technique's ability to detect the "natural" number of clusters even if they are entangled.

Using these three data sets, we can cover a wide variety of problems. The first set of 50 two-dimensional data points offers a pedagogical example that allows a good understanding of the models. The second file of 6000 two-dimensional data points represents a verification of the methods' efficiency as well as their robustness to noise and their ability to deal with groups of different shapes and proximities. Finally, the last data set is four-dimensional, therefore an actual problem that is not easily visualized.

Chapter 2

Percolation Clustering

Algorithm

The algorithm represents a heuristic, non-parametric hierarchical technique that emphasizes the similarity with the percolation process. The idea of exploiting the analogy with percolation phenomena in clustering techniques is gaining more and more attention in recent publications. For example the concept was used in the clustering of gene expressions [23]. The preceding method is a non-hierarchical technique in which the data points have multiple probabilistic memberships to different groups, mirroring only vaguely physical percolation. Our simulation focuses on monitoring the largest cluster size, which is one of the core characteristic function of the system in percolation theory. The discontinuities of this function indicate the “natural” number of clusters in the analyzed data set.

2.1 Introduction to Percolation Theory

Percolation models were used for the first time in the early 1940's to describe the gelation of polymers [24, 25]. However, as a mathematical topic, the theory was born the next decade with the contributions of Broadbent and Hammersley [26]. Ever since, it progressed steadily and by the mid 1980's was already established as a broad inter-disciplinary topic [27]. After a quieter period in the first half of the 1990's, the interest for percolation theory rose again due to its multiple applications and the developments in computer technology. The second part of the last decade and the beginning of the current one registered an avalanche of publications on the subject. Mathematicians, physicists, computer scientists, engineers, chemists and biologists have joined interests in the study of this phenomenon and the phase transitions associated with it.

Percolation is the structural modification experienced by thermodynamical systems (systems with a large number of elements) when switching from short-range to long-range connectivity between their components. The configuration of these systems changes suddenly from a set of disconnected parts to one large unitary ensemble. There are multiple natural phenomena that resemble this description such as: the flow of fluids in a porous medium, the spread of diseases in a population, the spread of fires in a forest, stochastic star formation, and so on. The transition has the same characteristics either when independent components suddenly become connected by randomly coupling neighboring elements, or, when the long-range connectivity

of the systems disappears by arbitrarily cutting off some of the short-range bonds. With this specification in mind we can add many more examples that correspond to percolation models, for instance: the conductor-insulator transition in metal films, connected-disconnected networks, para-ferromagnetic transition in diluted magnets, the liquid-gel to polymer transition, etc. [28]

All changes experienced by the physical systems mentioned above conform to the general characteristics of a phase transition. They are sharp, in the sense that the quantities describing the system encounter discontinuities and occur when specific parameters reach critical values. In the proximity of the transition, before reaching the discontinuity, the characteristic functions of the system obey power laws whose exponents are universal constants. I will succinctly describe below the transition parameter and main characteristic functions of percolating systems, focusing on the qualitative behavior in the thermodynamic limit when boundary effects are negligible. The emphasis is on the variation of system features versus percolation parameter rather than on the finite-scaling effects.

Most percolation models deal with connectivity between objects placed at random in a D -dimensional Euclidian space. Particular attention has been given to the cases $D = 2$ and $D = 3$ due to their obvious applicability. There are two main directions in percolation theory: one in which the system components are placed at random on a lattice, called lattice percolation, and the other one where the underlying background has no predetermined structure and objects can be placed arbitrarily anywhere in space, named continuum percolation.

In both cases connectivity remains the main concern and is usually defined based on adjacency. Two points are adjacent whenever they are nearest neighbors on a lattice or are closer than a prescribed distance in a continuum space. Therefore a natural parameter for percolation is one that controls the distances between system components such as a generalized density or a probability of lattice elements occupancy p . The critical value of this parameter, p_c , called the *percolation threshold*, varies wildly depending on system characteristics: its space dimensionality and structure (continuum or different lattice configuration), the shape of the system components (bonds or sites for lattice percolation, spheres or cubes for continuum percolation), definition of adjacency, etc. Despite the particulars of the different systems, their characteristic functions behave similarly during transition. Let \mathcal{F} be such a function, then:

$$\mathcal{F} \sim |p - p_c|^\lambda \quad \text{when} \quad p \uparrow p_c.$$

where the notation $p \uparrow p_c$ means $p \rightarrow p_c$ and $p < p_c$. The exponent λ , called the critical exponent of \mathcal{F} , is specific to the characteristic function and has a universal value related only to the dimensionality of the space. The percolation thresholds as well as the exact values of different critical exponents for a variety of systems are a matter of ongoing research.

Lattice percolation has been studied for a longer time than continuum percolation due to its topologically ordered structure which is easier to monitor. Thus, before referring to continuum percolation, which is closely related to our clustering algorithm, I will discuss the percolation parameter and three

core characteristic functions of lattice percolation. In these models the connecting elements are occupied lattice sites or lattice bonds. The entire space is seen as a grid whose components can take at random two Boolean values: occupied or not. There are two types of lattice percolation: site and bond percolation, both with similar attributes.

The variable that induces the percolation transition, namely the *percolation parameter* p , is defined as the fraction of occupied (or unoccupied) sites (or bonds). In the case of an infinite lattice (thermodynamic limit) this concentration becomes the occupation probability. The critical value of the percolation parameter varies with the lattice structure and shape of connected objects. For example, in the case of site percolation in a two-dimensional Euclidian space, $p_c = 0.50$ is the theoretically established exact value for a triangular lattice and $p_c \simeq 0.5928$ is the computed value for a square lattice[29]. For the bond percolation case, again in two-dimensional Euclidian space, $p_c \simeq 0.34729$ on a triangular lattice and $p_c = 0.5$ on a square one. In a three-dimensional space the critical value for site percolation on a simple cubic lattice is $p_c \simeq 0.3116$. Current research is being done to estimate more precise values of percolation thresholds. Rigorous mathematical models are available only for two dimensions and in many cases their solutions require the use of numerical methods. For higher dimensions we rely on computer simulations [28].

The percolation transition is expressed as the variation of the system characteristic functions versus the percolation parameter. One characteristic function carefully studied is the *mean cluster size*, $S(p)$, defined as the num-

ber of elements per cluster averaged over all existing clusters [28]. At low values of the concentration p , far below the percolation threshold, singletons or small sized clusters fill the system. As p increases, larger and larger clusters appear and we obtain a distribution of cluster sizes. As long as $p < p_c$ this distribution, and therefore the mean cluster size, is dominated by rather small or medium sized clusters. At $p = p_c$ numerous smaller clusters connect in a network that spans the entire lattice and the main contribution to $S(p_c)$ comes from the *maximal* (largest) *cluster size* $M(p_c)$. When the maximal cluster reaches the system limits for the first time some publications call it the *incipient percolation cluster* and the trail that connects the lattice edges is named *first percolation path*[29]. As the percolation parameter approaches the critical value from below, the mean cluster size $S(p)$ starts being dominated by the maximal cluster size, $M(p)$. Both grow rapidly and at $p = p_c$ they encounter a discontinuity. The discontinuity amplifies with increasing system size and develops into a singularity in the thermodynamic limit. This behavior is expressed by a power law:

$$S(p) \sim \frac{1}{(p - p_c)^\gamma} \quad \text{when} \quad p \uparrow p_c. \quad (2.1)$$

The critical exponent γ is independent of the lattice structure and depends only on the system dimensionality D . In the thermodynamic limit, above the percolation threshold the maximal cluster size becomes infinite. hence so does the average cluster size.

The second characteristic function that yields a similar behavior is the *correlation length*, $\xi(p)$, or the *connectivity function*, defined as the average

distance between two components belonging to the same cluster [28]. For simulation purposes an equivalent function can be used: *the mean spanning length*, $l_{avg}(p)$, which represents the spanning length averaged over all existing clusters. The spanning length of a cluster is the maximum distance between any two components:

$$l = \max_{i,j \in cluster} \|\mathbf{r}_i - \mathbf{r}_j\|. \quad (2.2)$$

For an infinite lattice in the proximity of the percolation threshold, the correlation length is dominated by the spanning length of the maximal cluster and behaves according to:

$$\xi \sim \frac{1}{(p - p_c)^\nu} \quad \text{when} \quad p \uparrow p_c. \quad (2.3)$$

Above p_c the connectivity function diverges, and in the thermodynamic limit is infinite for any $p > p_c$. The exponent ν is another universal constant dependent only on the dimensionality of the system.

To uncover yet another general trait of percolation we monitor the relation between the mean cluster size and correlation length in an infinite lattice for the sub-critical, critical and super-critical regimes. Let us consider a hyper-cubic box of large length L , whose volume increases according to the space dimensionality D as $V \sim L^D$. Throughout the first stage, far below the percolation threshold, the size (mass) of the average cluster trapped in the box varies slowly for various large values of L as:

$$S(p, L) \sim \log(L) \quad \text{for} \quad p < p_c. \quad (2.4)$$

Because the space is almost empty, changing the box scale does not drastically change the size of the average cluster. In other words, its mass grows very

slowly with increasing volume, therefore its effective dimensionality is zero. The critical regime begins when the system approaches percolation, for $p < p_c$ but p big enough to create large clusters. At this point the average cluster size starts being dominated by the maximal cluster. The largest cluster, which will become the percolating cluster, does not have a uniform density but rather a fractal structure due to the in-between empty cells. Considering the same box of length L the maximal cluster size (mass) $M(p)$ scales as follows:

$$M(p, L) \sim L^d \quad \text{for} \quad L < \xi(p). \quad (2.5-a)$$

$$M(p, L) \sim L^0 \quad \text{for} \quad L \geq \xi(p), \quad (2.5-b)$$

where $\xi(p)$ is the correlation length for the given concentration p and d is the universal fractal dimension of the incipient percolating cluster. The above equations show that when the box's linear dimension is smaller than the correlation length, the enclosed maximal cluster is seen as a fractal percolating cluster. Due to its many empty cells the mass (size) of this structure increases more slowly than its volume, according to equation (2.5-a). The non-integer dimension d is a universal constant that depends only on the space dimensionality D , but is independent of the lattice particularities. For 2-dimensional space $d \simeq 1.89$ [28]. When the linear dimension of the box becomes of the same order of magnitude or larger than the correlation length, the empty spaces in-between become relevant. The maximal cluster reaches its limits inside the box and its mass remains constant for any box length as expressed in equation (2.5-b). When the concentration $p = p_c$, the correla-

tion length reaches an order of magnitude comparable to the infinite lattice linear dimension and the maximal cluster becomes the percolating cluster. No matter how large the box length, L , the maximal cluster inside has a fractal dimension d . In the super-critical regime, when the parameter p exceeds the critical value p_c , the maximal cluster fills the lattice almost uniformly and its size (mass) increases similar to its volume:

$$M(p, L) \sim L^D \quad \text{for} \quad L \rightarrow \infty. \quad (2.6)$$

For an infinite system many absolute quantities, such as the average cluster size or the correlation length, diverge above the percolation threshold. Therefore, normalized functions would be more appropriate to describe the system's super-critical behavior. Such a characteristic function is *the probability of percolation* $P(p)$ for a given concentration p . It corresponds to the normalized maximal cluster size and is defined as the fraction of the entire system occupied by the spanning cluster. For a two-dimensional square lattice of size L :

$$P(p) = \frac{M(p)}{L^2}. \quad (2.7)$$

In the thermodynamic limit, $P(p)$ is the probability that a randomly selected lattice site (or bond) belongs to the spanning cluster. This quantity is also called *the strength of the infinite cluster*. $P(p)$ plays the role of an order parameter since it is zero for p below p_c when there is no spanning cluster, and becomes a positive value at the percolation threshold ($p = p_c$) when the first percolation path appears. The increase is abrupt and near threshold

$P(p)$ is given by a power law:

$$P(p) \sim (p - p_c)^\beta \quad \text{for} \quad p \downarrow p_c. \quad (2.8)$$

where $p \downarrow p_c$ means that occupation probability approaches the critical value from above ($p \rightarrow p_c$ and $p > p_c$). When all occupied sites (or bonds) are interconnected $P(p) = p$ and when the entire lattice is occupied $P(p) = 1$.

From the discussion above, the maximal (or average) cluster size and correlation length are features that describe the system in the sub-critical regime, while $P(p)$ is a critical and super-critical attribute. The exponents γ , ν , β and the fractal dimension d of the percolation cluster are universal values. Analogies between percolation and thermally induced phase transitions, such as a para-ferromagnet or a liquid-gas transition, reveal the universality of critical phenomena behavior. The concentration p corresponds to temperature T , with the observation that they play an inverse role. Percolating systems become connected above p_c and a thermal system becomes ordered (ferromagnetic or liquid) below T_c . The mean cluster size S corresponds to a susceptibility χ or a compressibility K , since it describes a local order as opposed to the average disordered system. The strength of the infinite cluster (percolation probability) P corresponds to the magnetization M or to the difference between liquid-gas densities, given that all these quantities are zero on one side of the phase transition [28, 30].

The last few years registered an increased interest in continuum percolation models. The objects in this case are equal sized geometrical shapes (spheres, cubes or cylinders) built around Poisson distributed points in a

uniform D -dimensional Euclidian space. The objects can be impenetrable or can overlap. Such a model is the well-known “Swiss cheese” in which equally sized void spheres of radius r appear at random in a system. The spheres can overlap and percolation occurs when the system becomes disconnected. In continuum percolation the role of the occupation probability, p , is taken by the fraction of occupied (or void) spaces ϕ . The percolation transition is described by the variation of system characteristic functions versus ϕ . As in the lattice percolation case the critical value of the percolation parameter ϕ_c depends on the system particularities. These percolation thresholds represent an area of very active research. Recent publications present these values for a two-dimensional and three-dimensional space where the connecting objects are squares or cubes built around Poisson distributed points. For squares aligned with the axes $\phi_c \simeq 0.66$ and for randomly oriented square $\phi_c \simeq 0.62$. In three dimensions for aligned cubes: $\phi_c \simeq 0.28$ and for randomly oriented cubes $\phi_c \simeq 0.22$ (noticeably smaller). The critical volume at percolation threshold for spheres built around Poisson distributed points is $\phi_c \simeq 0.29$ only 4 % larger than the critical value for aligned cubes [31].

2.2 Description of the Algorithm

Our clustering method represents a new approach to the nearest-neighbor algorithm. By exploiting an analogy with the percolation process we add insight into the clustering procedure and transform it from a parametric technique into a non-parametric one. Applied to an unknown set of data the

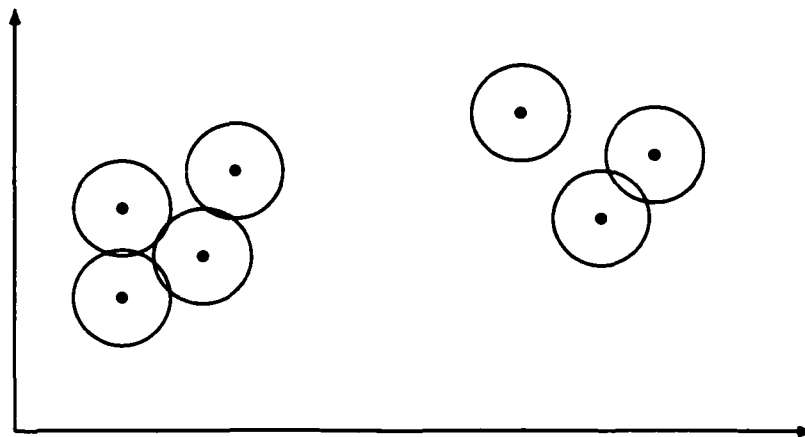


Figure 2.1: Seven two-dimensional data points grouped into roughly two clusters of sizes four and three. The first cluster (the leftmost one) is completely connected and the second cluster (the rightmost) has not yet linked fully.

“Percolation Clustering Algorithm” determines automatically the number of clusters and their location using no a priori assumptions.

The algorithms currently utilized in the study of continuum percolation use equally sized geometrical objects such as squares [31], disks [32], or thin rectangles [33] in two dimensions and spheres, cubes or ellipsoids of revolution in three dimensions [31]. These objects are not placed completely randomly in space, but they are generated such that their centers form a Poisson distribution. In contrast, clustering procedures aim to locate unevenly scattered groups of points, therefore our technique utilizes objects of changing size.

We imagine data points as the centers of initially small hyper-spheres expanding in a D -dimensional space. Each sphere “swells”, increasing its

radius until it touches other spheres and “sticks” to them. This is the growing scheme. A group of mutually connected spheres forms a cluster. A two-dimensional example with seven data points grouped into two clusters is illustrated in Figure 2.1. The quantity we are primarily interested in is the cluster size, defined as the number of sample points in each cluster. At the beginning only the closest points are linked and cloudlike structures start growing in different areas of the sample space. As the spheres keep augmenting their volume, more and more data points connect. Once a “chain” of such spheres spans an entire cluster, percolation occurs and the cluster size remains constant until a group of mutually connected spheres merges with a neighboring cluster. When two clusters connect, a sudden jump in cluster size can be detected. Coalescence continues until all data points are grouped in one large, final cluster. The analogy with the percolation phenomenon allows us to sense the changeover that occurs when a cluster gets “locked in place”. This transition can be detected in two ways. First, by monitoring cluster size as a function of distance between connected points, we notice a plateau once the cluster is completely formed, followed by an abrupt jump when the cluster merges with another cluster. The second way is to examine the discontinuities in the first derivative of cluster size with respect to distance between connected points. In this case, the growth rate of cluster size encounters a sharp peak when two clusters connect or when a large cluster gets “locked in place”.

The successive “swelling” and joining of data points is accomplished by browsing the ascending ordered list of distances d_{ij} between sample points.

For each distance in the list, the two adjacent points are connected. We monitor the size of the largest cluster as a function of distance between sample points. In fact, for a more detailed analysis of the data, the algorithm can output the sizes of the first, second, third or any chosen number of largest clusters.

The binary distances used in our simulations are Manhattan distances defined by equation (1.3) and the Euclidean norm described in equation (1.2). Obviously, the Euclidean distance corresponds to the image of an expanding hyper-sphere while the Manhattan distance defines enlarging hyper-parallelepipeds around data points. In a low-dimensionality sample space (for well-defined clusters) both metrics generate basically equivalent results and for computational efficiency reasons the Manhattan distance is a more convenient choice. A number of partitionings obtained by means of the Manhattan distance were reconfirmed using the Euclidean norm.

The bottleneck of our algorithm is building the ordered list of distances d_{ij} between points. Given a data file of N sample points, the total number of distances between them is $n = \frac{N(N-1)}{2}$, hence the number of elements in the distances list is of order N^2 . Note that all sample points end up being grouped in one final cluster before the expanding spheres around data points reach radii comparable to the largest distances between samples. Once the largest cluster contains all data points the simulation ends. Therefore, a simple way to limit the number of elements in the distance list is to insert only the values smaller than a threshold. The threshold has to be chosen such that by browsing the ascending ordered list of distances, at the end the

largest cluster contains all of the sample points. This value is not known beforehand and has to be tuned for each data set. The cut-off distance is the only adjustable parameter of the Percolation Clustering Algorithm and it has an auxiliary character in that its value has no bearing on the final clustering results. In our calculations the threshold was typically less than one-half of the average distance between sample points.

2.3 Computational Details

Before we finish the general description of the method, it is useful to present the technical details of algorithm implementation. The codes are written in C++ and make use of the Standard Templates Libraries.[35] Developed in early 1990 by Stepanov and Lee at Hewlett Packard Laboratories, these generic (template) containers, iterators and functions became part of standard C++ in 1994. They allow a fast and efficient implementation of different computational techniques as will be the case in our algorithm.

Each sample is represented by a structure, called a point, that contains a label (the sample number in the original data file), an array of attributes (coordinates), a group number (which represents the cluster number to which the point will be assigned) and a link (a pointer to another point) that allows the point to be hooked to different clusters. Since the total number of sample points is known, the points are stored in an array.

The first step of the simulation is the building of an ordered list of distances. In fact the items stored in the list are structures that contain a

distance d_{ij} and the two addresses of the adjacent points. This way, for any ordered distance in the list, the entire information about the edge points is easily available.

As mentioned above, for a set of N sample points, the number of elements in the distance list is proportional to N^2 . As we know, the order of magnitude \mathcal{O} of the time T it takes to sort a list of n items depends on the number of items in the list. Since the computing time required to order the distances increases strongly with the number of elements in the list, the efficiency of the sorting algorithm is of the utmost importance.

For a small number of sample points (up to a couple of hundred) the distances d_{ij} are calculated one by one and inserted into a singly linked list such that the resulting sequence is in ascending order. This simple linear insertion sort, based on a singly linked list requires repetitive comparisons with all of the previously stored values. The total number of comparisons is:

$$2 + 3 + \dots + n = \frac{n(n+1)}{2} - 1 \sim n^2.$$

Thus, for such an algorithm $T(n)$ is $\mathcal{O}(n^2)$ [35]. For N sample points the computational time $T(N)$ needed to build the ordered list of distances is $\mathcal{O}(N^4)$. The linear insertion sorting scheme has the benefit of simplicity and requires very little overhead, which makes it suitable for small data sets. However, larger files call for more efficient sorting algorithms.

For large data sets the Percolation Clustering Algorithm uses heapsort [36], which is an efficient selection sort developed in 1964 by John Williams. A heap is a complete binary tree, hence all its nodes have two descendants

except, possibly, the ones on the last level. If the last level is incomplete the occupied vertices are placed in the leftmost positions. The tree has to satisfy the heap-order condition, which means that data stored in one node is larger or equal to the data stored in its descendants. Any insertion or deletion in a heap structure is done such that it preserves the abovementioned properties. The sorting algorithm based on the heap starts by extracting the largest data from the root, storing it at the bottom of a container and replacing it by the value of the rightmost leaf, which at this point disappears. The next step is to restore the heap properties of the resulting tree by moving down the new root value, until the next largest data surfaces to the root and the algorithm repeats itself. It can be demonstrated that the comparison and interchange scheme described above has a worst-case computing time of order

$$T(n) \text{ is } \mathcal{O}(n \log_2 n), \quad (2.9)$$

hence, for N sample points there are $n \sim N^2$ number of binary distances and:

$$T(N) \text{ is } \mathcal{O}(N^2 \log_2 N^2) = \mathcal{O}(N^2 \log_2 N).$$

The computing time for a large number of sample points still increases as $N^2 \log_2 N$. but it has been theoretically demonstrated that the average complexity of a sorting method cannot be lower than the one expressed in equation (2.9) [35].

The Standard Template Library provides a generic heapsort-based priority queue, that can order any items for which the operator “<” is defined. The sorted data is stored in an auxiliary container of the user’s choice. Since

the Percolation Clustering Algorithm is applied to differently sized files we used a container known as a vector, which is a linear contiguous storage whose capacity can be dynamically increased as needed. The items stored in the priority queue are structures that contain the binary distance, d_{ij} , and pointers to the adjacent points. These structures are sorted in ascending order of their distance members.

The distances are inserted into an ordered list or priority queue as they are calculated while browsing the sample points array. The second step of the algorithm is sweeping the list. Once a distance is extracted from the priority queue, its end points connect and they are linked together in a cluster. The clusters are implemented as numbered, singly-linked lists of points and when a free point connects with a member of a cluster numbered n_1 , the point is added at the end of this list and receives a group number n_1 . All points in a cluster have their group number equal to the cluster number they belong to. Therefore, any time during the simulation we know the cluster arrangements. In case a point belongs to the cluster n_1 and becomes connected to a point attached to cluster n_2 , the two lists n_1 and n_2 are linked head to tail. The order number of the new cluster is $\min(n_1, n_2)$ and the group numbers of all its members are renumbered accordingly.

2.4 Computational Results

We have implemented this method and tested it on a number of cases. The first example is the simple two-dimensional “toy” problem presented in sec-

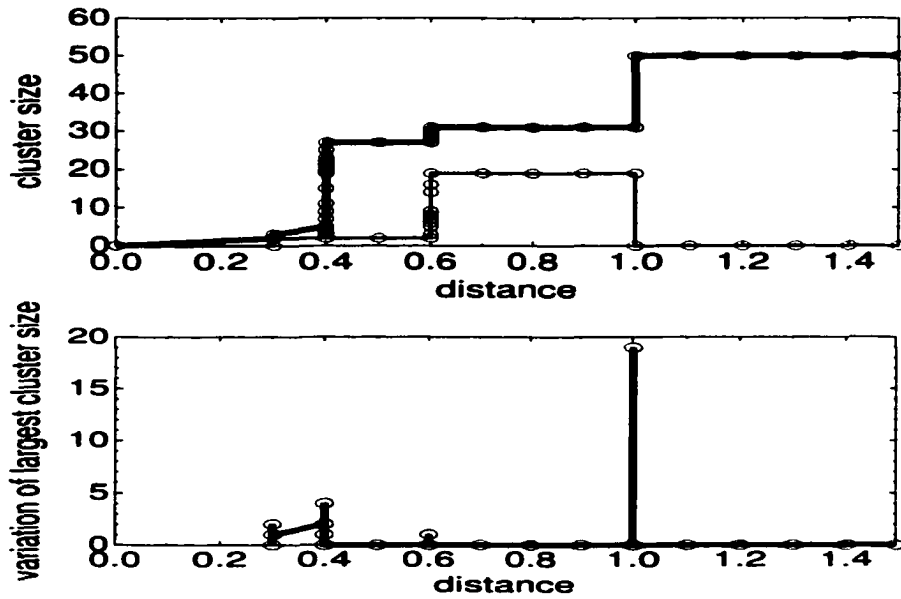


Figure 2.2: *Top*: Size of the largest (thick line) and second-largest (thin line) cluster as a function of distance between connected points for data set in Figure 1.2. *Bottom*: Growth rate of largest cluster size as a function of distance between connected points for data set in Figure 1.2.

tion 1.3. The results of the Percolation Clustering Algorithm for this data set are presented in Figure 2.2. The top chart displays the size of the largest and the second largest cluster as a function of the distance between connected points. One notes how the largest island size increases up to a distance of 0.6, where it reaches a plateau of 31 points corresponding to the rightmost circle. The plateau begins when all the points of this circle are connected, hence when the *largest intra-cluster nearest neighbor distance* is reached. The second cluster (thin line) attains a sustained plateau of 19 points at the same

distance 0.6, which in this case represents the largest nearest neighbor distance within the leftmost circle. At this point the second cluster has also been correctly located, but it is not yet identified by monitoring the largest cluster size. Note that if two clusters have the same largest intra-cluster nearest neighbor distance, such as the two circles in Figure 1.2, they achieve plateaus, thereby connecting all their sample points, at the same point. Clearly the denser cluster (the right circle in Figure 1.2) has a larger growth rate, as the top graph of Figure 2.2 reveals. Finally, at a distance of 1, which corresponds to the *minimum inter-cluster distance* expressed by equation (1), the two islands merge, the largest cluster size has a jump of 31 to 50 points and the second cluster disappears. The 19 point jump of the largest cluster size can also be detected on the bottom graph of Figure 2.2. All data points have been grouped in one single cluster at a distance of 1, which represents less than half of the 2.4 average distance between data points. It is important to note that the apparent jump of the largest cluster size from 5 to 27 points, corresponding to a distance of 0.4, is not due to cluster coalescence, but to the roughly uniform distribution of points in the cluster. Many sample points of the rightmost cluster in Figure 1.2 have their nearest neighbors 0.4 distance away. Therefore this value appears several times in the ordered list of distances, which implies that points are added successively rather than in one piece. This occurrence is clearly reflected by the multiple values that the growth rate of the largest cluster size encounters at the distance 0.4, as one can see on the bottom graph of Figure 2.2. The same remark can be made regarding the apparent leap of the second cluster size at distance 0.6. It is

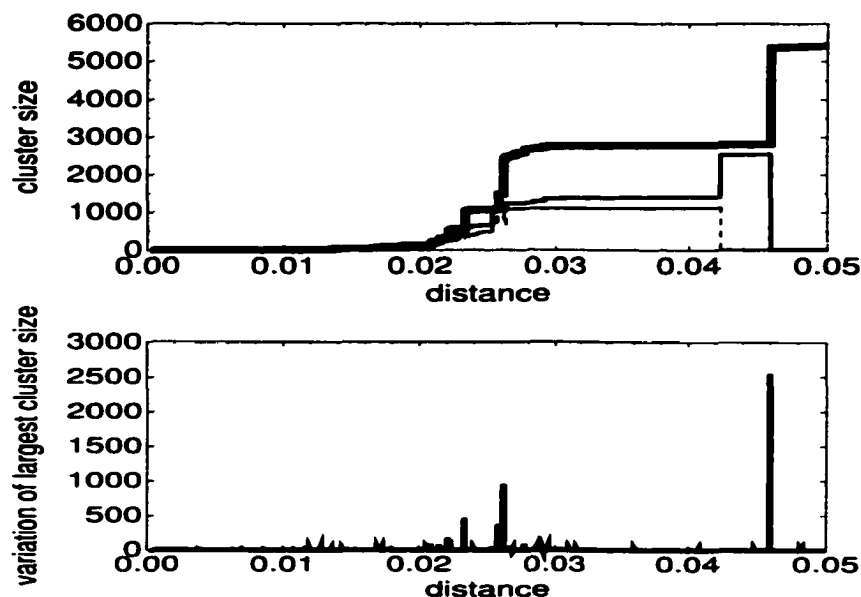


Figure 2.3: *Top*: Size of the largest (continuous thick line), second-largest (thinner line) and third-largest (thinnest dashed line) cluster as a function of distance between connected points for BWD data set. *Bottom*: Growth rate of largest cluster size as a function of distance between connected points for BWD data set.

clearly important to monitor simultaneously the largest cluster size as well as its first derivative.

To compare our algorithm to other existing non-parametric clustering procedures, we analyzed a second two-dimensional data set [12]. The sample distribution is described in section 1.3 and plotted in Figure 1.3. As previously mentioned, the BWD data set represents a cumbersome trial for many clustering algorithms. Figure 2.3 presents the results of the Percola-

tion Clustering Algorithm for the 6000 two-dimensional data set described above. The top chart monitors the size of the three largest clusters as a function of the distance between connected points. The largest cluster size encounters three plateaus that clearly indicate the existence of three clusters, but the presence of the background requires a careful interpretation of the plot. The first plateau of about 1100 samples, which identifies the upper cluster, is shortly followed by a jump of about 1700 points, produced by the coalescence of the upper and center clusters. This jump is larger than the size of the merging cluster (the center ring which has 1400 sample points) due to background points that have been successively connected during the “expansion” of the merging clusters. By the same token, one notices the slow increase of the largest cluster size along the plateaus. Hence, plateaus that are not completely flat indicate the presence of the background and, in this case, the jump sizes do not correspond to the exact size of the joining clusters. The number of clusters can also be correctly identified by examining the first derivative of the largest cluster size as a function of the distance between connected points (bottom part of Figure 2.3). We detect three large jumps (the second one made up of two close steps) which identify the three denser areas. Because the clusters have the same uniform distribution of points, their largest intra-cluster distance is about the same. Thus they will “percolate” at approximately the same point, but, by monitoring only the largest cluster size, they will be recorded later. This method works well for the data set presented in Figure 1.3 as can be seen from Table 2.1, which summarizes the real data distribution as well as partitions obtained using the Percola-

Method	Largest cluster (no. of points)	Middle cluster (no. of points)	Smallest cluster (no. of points)
Data Distribution	2729	1356	1084
Percolation Algorithm	2744	1354	1079
Super-Paramagnetic	2759	1380	1097

Table 2.1: Data distribution and the results of two clustering algorithms for the BWD problem.

tion Clustering Algorithm and the Super-Paramagnetic Clustering technique (SPC).

Several choices are available to better determine the size of each cluster in the presence of the background. One is to use a filtering method that will eliminate the background points before the clustering procedure. The other one is to apply the Percolation Clustering Algorithm iteratively, running the simulation until one cluster has clearly emerged (as signaled by a sustained plateau or a large jump). At that point one interrupts the simulation, locates the cluster in question, removes it from the data set and then restarts the clustering algorithm with the remaining data. Although iterative application of the Percolation Clustering Algorithm is not easily automated, this may be the method of choice for some other complicated problems.

To demonstrate the algorithm's behavior on a more realistic case, we have

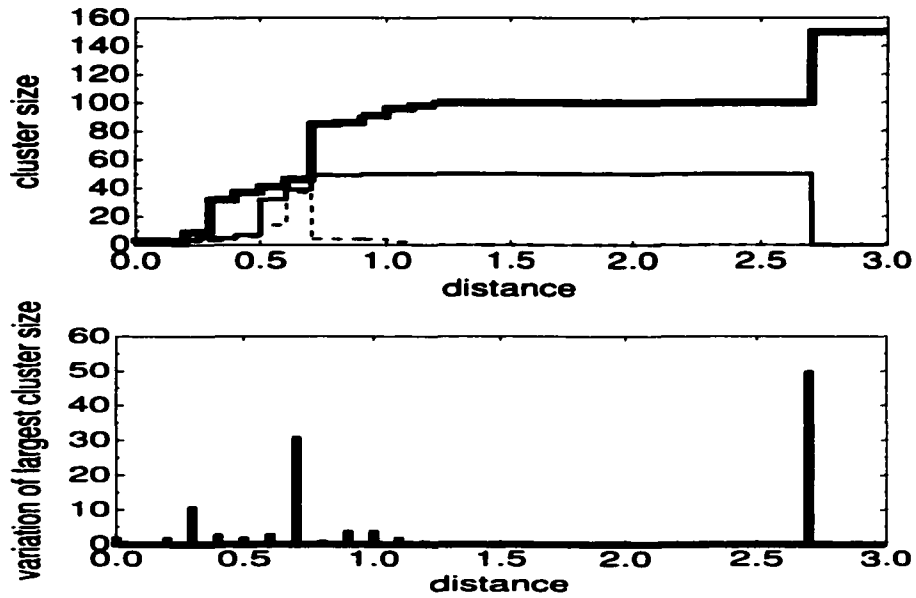


Figure 2.4: *Top*: Largest cluster size as a function of distance between connected points for the iris problem. Note plateaus and jumps near cluster sizes 50 and 100. *Bottom*: Rate of variation of largest cluster size as a function of distance between connected points for the iris problem.

studied the well-known four-dimensional iris data set presented in the introductory chapter, section 1.3. The Percolation Clustering Algorithm generates the results presented in Figure 2.4.

The top plot presents the sizes of the three largest clusters as a function of distance between sample points. One can observe a small plateau at abscissa 0.7 followed by a steep jump. At this point an inversion between the first and second largest cluster occurs and can be explained as follows: at the beginning of the simulation the well-defined group of iris setosa sample points connect.

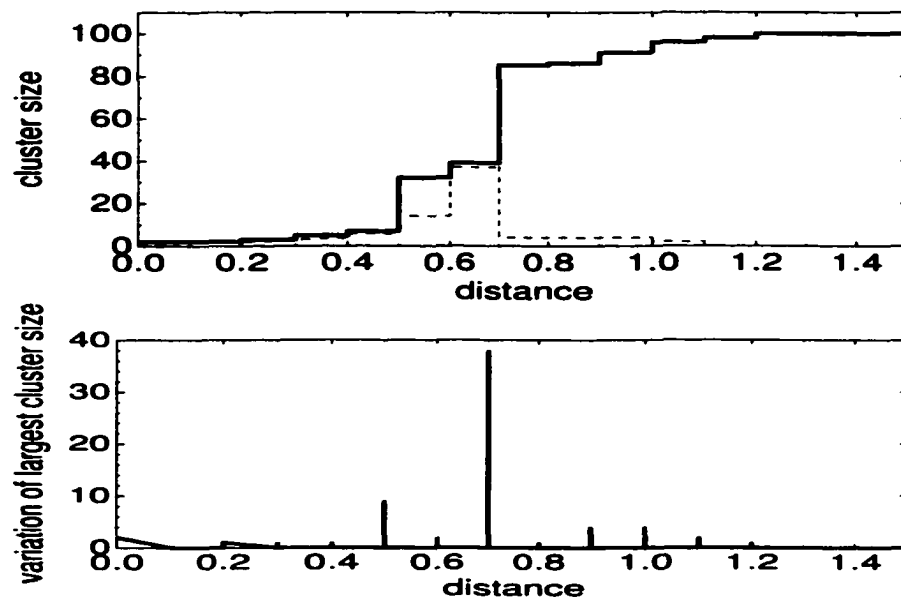


Figure 2.5: *Top*: Largest cluster size as a function of distance between connected points for the iris versicolor and iris virginica data. Note plateaus in the cluster sizes at 39 and 40 points as well as the jump that follows. *Bottom*: Rate of variation of largest cluster size as a function of distance between connected points for the iris versicolor and iris virginica data.

forming the largest cluster. Before all of them are linked, *i. e.* when their largest intra-cluster distance is reached, the size of this cluster is surpassed by the second growing set of flowers iris versicolor and virginica. The first jump is due to the coalescence of these two close groups. The long flat plateau of 100 points indicates the size of the two clustered groups and the existence of no background. The third cluster of 50 flowers is still detached and will merge later at the distance of 2.7. Analyzing the variation rate of the largest cluster

Method	Iris setosa (no. of flowers)	Iris versicolor (no. of flowers)	Iris virginica (no. of flowers)
Data Distribution	50	50	50
Minimal Spanning Tree	50	50	50
Percolation Algorithm	45	40	38
Super-Paramagnetic	45	40	38
Valley Seeking	67	42	37
Complete-Link	81	39	30

Table 2.2: Results of different clustering algorithms for iris data. Note the similar partitions produced by Percolation Clustering Algorithm and Super-paramagnetic Clustering procedure.

size. presented on the bottom of Figure 2.4. one notes the three peaks that correspond to the three groups of flowers. In order to better understand the interplay between the largest and second largest cluster, we consider it helpful to identify and eliminate the well separated iris setosa group and rerun the algorithm for the 100 iris versicolor and iris virginica flowers. The results, similar to the one obtained for the entire file, are presented in Figure 2.5. Note that the inversion present in Figure 2.4 has been eliminated, thereby allowing a clearer monitoring of the cluster sizes.

The classification results of our algorithm as well as the partitions ob-

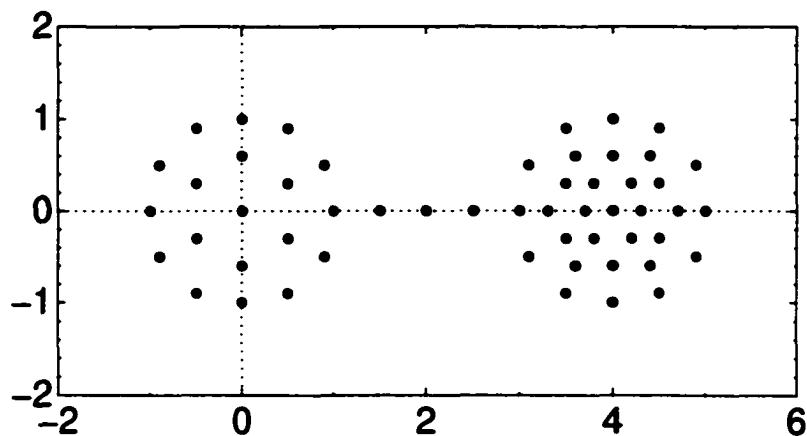


Figure 2.6: Two-dimensional sample points, grouped in two circular clusters of different densities, set two units apart and connected by a bridge.

tained by means of other clustering methods are listed in Table 2.2. Some data points have not been classified, but we note that all prior methods based on a heuristic clustering approach are unable to correctly classify all of the points. Our success rate is as good as that of the superparamagnetic technique, but at a considerable gain in computational efficiency. The only method able to reproduce entirely the original data distribution, is the Minimal Spanning Tree [37], which uses an ultrametric distance function. An ultrametric distance satisfies two properties, the distance between a point and itself is zero ($\hat{d}_{ii} = 0$) and the symmetry property ($\hat{d}_{ij} = \hat{d}_{ji}$), as well as the ultrametric inequality $\hat{d}_{ij} \leq \max\{\hat{d}_{ik}, \hat{d}_{kj}\}$ [38].

In spite of the encouraging results obtained with the Percolation Clustering Algorithm, we are aware that our method, as presented so far, empha-

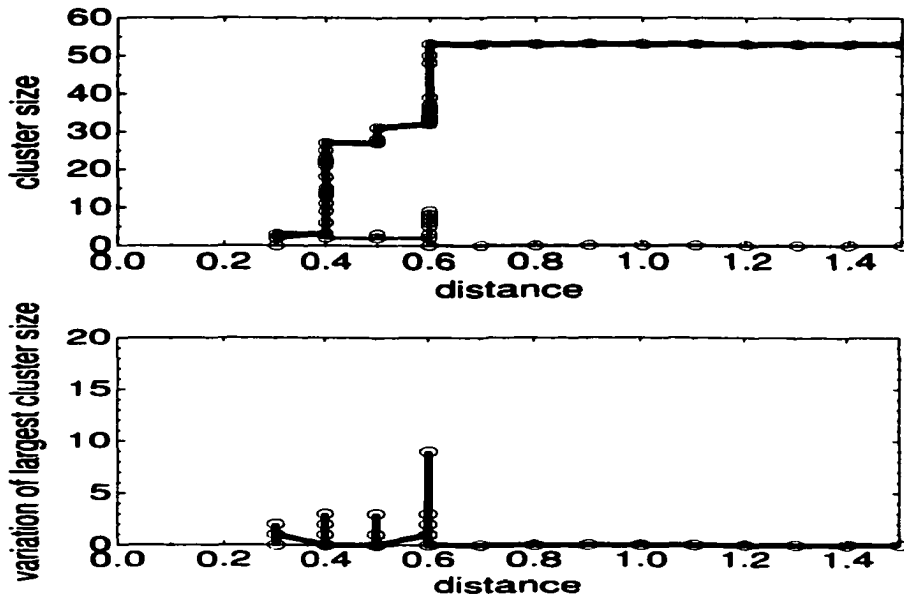


Figure 2.7: *Top*: Largest cluster size as a function of distance between connected points for the two-dimensional set represented in Figure 2.6. *Bottom*: Rate of variation of largest cluster size as a function of distance between connected points for the data set represented in Figure 2.6.

sizes closeness and ignores the connectivity between groups of sample points. Consider, for example, the first simple two-dimensional toy problem whose sample points are grouped in two equal circles ($R = 1$) of different densities. In a new set of data, the clusters are $2R$ apart and the set contains 3 more points which create a bridge between the two islands (see Figure 2.6).

Graphs of the largest cluster size versus the distance between connected sample points and the variation rate of largest cluster size as a function of the distance between connected points are presented in Figure 2.7. The increase

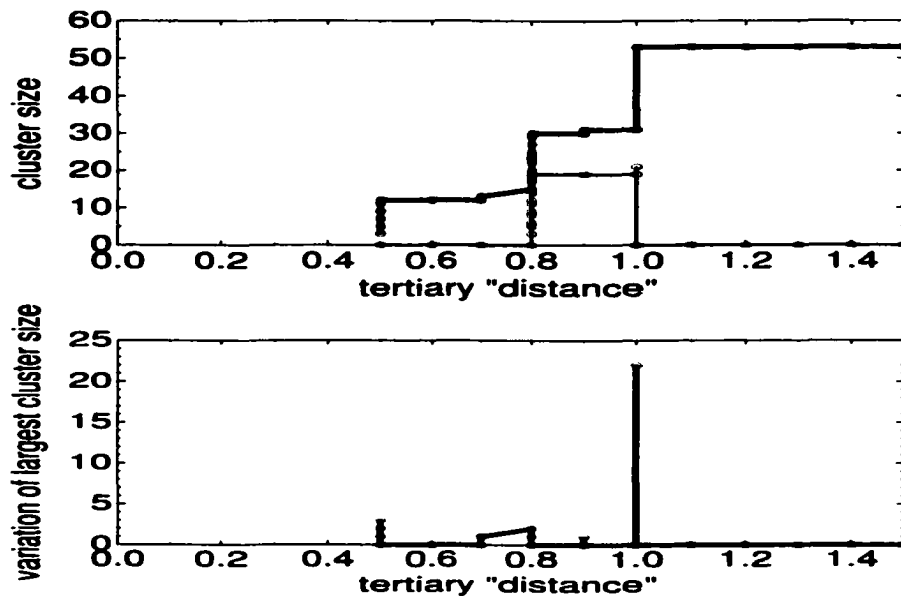


Figure 2.8: *Top*: Largest cluster size as a function of distance between connected points for the two-dimensional set represented in Figure 2.6. *Bottom*: Rate of variation of largest cluster size as a function of distance between connected points for the data set represented in Figure 2.6.

of the largest cluster size is almost continuous and the second cluster cannot be detected. To a certain extent, depending on the definition of a cluster, one can argue that the configuration presented in Figure 2.6 represents only one group of points. Nevertheless, using a more common perception of a cluster, which considers the interconnectivity and compactness between cluster members an important factor, the file in Figure 2.6 contains two clusters of different densities connected by a bridge. In order to express this concept we introduce a new type of similarity function that will express the connec-

tion between multiple sample points. We build a “tertiary distance” between three points as the maximum of the three binary norms d_{ij}, d_{ik}, d_{kj} :

$$d_{ijk} = \max(d_{ij}, d_{ik}, d_{kj}). \quad (2.10)$$

This new function allows one to group three sample points at a time as opposed to only two, thereby emphasizing compactness. The new algorithm using this approach has to create an ordered list of tertiary distances. Note that in the case of an N sample points data file the number of items to be sorted is of order N^3 . For the case in discussion, the results presented in Figure 2.8 are encouraging. The top diagram shows the sizes of the first two largest cluster sizes as a function of the tertiary “distance” defined in the equation 2.10. Note the two plateaus of 19 and 31 points, which correspond to the number of sample points in each circle. The bottom chart displays a jump of 22 points in the largest cluster size which suggests that the leftmost circle simultaneously absorbs all three bridge points and bonds to the largest cluster. Note that the tertiary “distance” at which the two circular islands finally connect is 1, meaning that the middle bridge point concurrently links both clusters.

The idea can be extended to quaternary or any user defined order q of similarity functions, with the observation that for N sample points the number of items to be ordered increases as N^q . This makes the algorithm less and less efficient for larger data sets, and therefore it was not applied to the 6000 points two-dimensional data file.

The method seems to correctly identify the number of clusters and is ro-

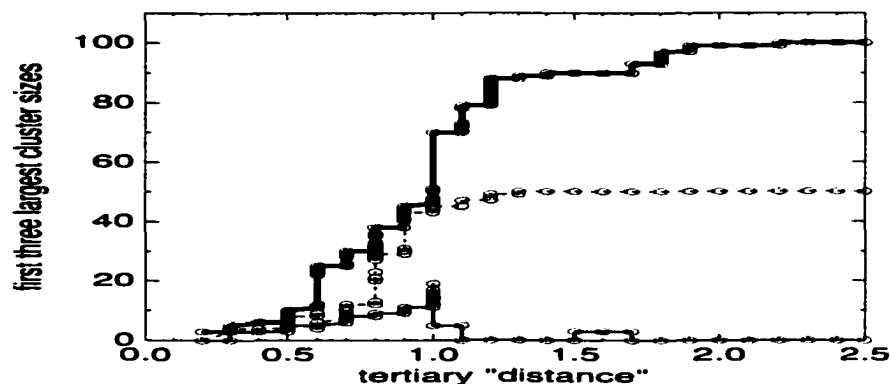


Figure 2.9: First three largest cluster sizes as a function of tertiary “distance” between connected points for iris data set.

bust to noise or outliers, but in the case of overlapping clusters this approach needs to be used with caution. Trying to apply the same procedure for the iris data set, the partitioning worsens. As one can see from Figure 2.9, three clusters can be identified. One cluster is of 50 sample points and is clearly separated from the rest of the file. The other two groups are overlapping clusters of 100 flowers total, containing 20 and 47 compact sample point cores and miscellaneous outliers. The two groups of iris versicolor and virginica are identified by their core, more compact members, while the outliers are collected indiscriminately together. Interestingly enough, similar partition of the 100 iris versicolor and iris virginica flowers is obtained with other clustering algorithms which emphasize the compactness between cluster members, as is the case of the Nucleation and Growth Clustering method presented in the next chapter.

Chapter 3

Nucleation and Growth Clustering

Another heuristic approach to the clustering of data sets is described in this chapter. This novel method is based on an analogy with the process of nucleation and growth that occurs in island formation during epitaxial growth of solids or in other solid-solid reactions. The technique is competitive with existing algorithms and offers some possible advantages for certain types of data distributions. Furthermore some of the results presented below have been previously published [39].

Data mining is rapidly finding acceptance in materials science as a powerful way to detect trends, to optimize manufacturing, and to design or discover new compounds [40]. Techniques used range from neural networks and genetic algorithms to clustering approaches, such as the k-means method. Here we address the reverse problem and exploit ideas from materials science to de-

sign a new data mining algorithm, *i. e.* the clustering of a multi-dimensional data set [41].

In the ensuing discussion we will assume that the data to be clustered resides in a two-dimensional Euclidean space. This is merely a means to aid visualization of the algorithm and does not constrain the range of applicability in any way. Thus, we imagine a group of N data points in the plane and we need some way to divide this group into clusters, depending on the distance between the points. This grouping together reminds us of the nucleation and growth processes that occur in materials as they solidify on the surface of a thin film that is growing by deposition of particles. We therefore propose to exploit this analogy and suggest a novel “Nucleation and Growth Clustering” (NGC) algorithm. The method is able to deal with clusters of varying densities and automatically finds the appropriate number of clusters (as opposed to the k-means technique, which needs to be re-run for different k-values). Moreover, trapping in local minima is less likely in the NGC method since the deposition is random. While the Percolation Clustering Algorithm, described in the previous chapter, emphasizes the connectivity between data points, the current technique considers simultaneously the connectivity as well as the compactness of the samples. In all studied cases, the convergence of this method is very rapid, particularly compared to the other two physically motivated approaches [12, 13, 14, 15, 19]. Thus, the technique combines the speed of heuristics with the convenience of physically motivated approaches.

3.1 Algorithm Description

The core idea of our novel approach is to consider the clustering process as akin to a nucleation and growth phenomenon well-known in materials science and solid state physics [42, 43]. The original data points are called the “seed-data” and are taken as the nucleation centers, around which aggregation will occur. We now imagine a deposition process in which additional data points (“ad_data”) are introduced randomly in the plane. When the ad_data appear within a defined threshold distance, d_t , to a seed-data point, the ad_data will stick to it. This threshold distance is defined as half of the minimum distance between any two seed-data points in the entire data set. If they appear at a distance greater than d_t the ad_data are removed from the space. Although it does not correspond to the physical deposition phenomenon, we perform this removal due to the large empty regions that can occur in clustering problems. Diffusion, while occurring in lattice deposition, would not be an efficient tool in linking the seed-data and would add a large computational overhead to the simulation. The plane, or deposition space, where the ad_data is randomly generated is defined by the maximum and minimum coordinate value of each dimension over the sample set. To these margins we add twice the threshold interaction distance, so that sample points on the edges will be able to have ad_data deposited completely around them as opposed to just on one side.

As more and more ad_data are introduced islands develop just like in epitaxial thin film growth [43]. Thus, as the simulation proceeds and more and more particles have been deposited, one notices the formation of distinct

islands, corresponding to clusters. In a given island, the seed-data form the cluster, while the ad_data are merely a tool to facilitate the detection of the connectivity of that cluster. Eventually, all islands merge together and form one giant cluster. However, before this happens, distinct clusters may be observed. This can best be monitored by looking at the cluster-size as a function of time. Each time step represents a newly generated ad_data that might stick or be removed. Thus, similarly to the Percolation Clustering Algorithm, the output presents the largest cluster sizes versus the number of generated ad_data. As long as a cluster is still growing, its size will increase over time, sometimes in sudden jumps when two clusters merge. Once the cluster has formed completely, its size will remain constant for a relatively long time, until it merges with a neighboring cluster. By detecting these plateaus in the cluster-size curve we can determine the number of clusters and their location.

3.2 Computational Details

Each data point has been implemented as a structure with an integer label, an array of coordinates, and a group index that indicates the cluster it belongs to. Sample points are stored in an array. The randomly generated ad_data are represented by similar structures with the addition that each of them stores the distance they are from the point that they are attached to. All connected added points are stored in a singly linked list, so they can be easily attached or removed.

In order to generate a random point in the deposition space we use the random number generator *rand()*. This function, provided by the Standard Template Library (STL) in C++ or Linux C Library, returns a pseudo-random integer between 0 and `RAND_MAX` (the largest integer offered by the operating system, usually 2^{32}). In order to obtain a repeatable sequence of pseudo-random integers, the *rand()* function needs a seed, which is set by the argument of the function *srand()*. If no seed value is provided, the *rand()* function is automatically seeded with the value 1. In our computation we used a seed value of 2. Since in some old implementations of the *rand()* function the lower order bits of the generated number are less random than the higher ones [44], we produce a random real number $r \in [a, a + b)$, using the formula:

$$r = a + b * (\text{rand()} / (\text{RAND_MAX} + 1.0)). \quad (3.1)$$

where `RAND_MAX` is the maximum value of the integer type

$$\text{RAND_MAX} = 2^{15} = 32,768.$$

3.3 Computational Results

This method has been implemented and tested on several benchmark cases. The first data set on which the algorithm was tested consists of two groups of points, roughly corresponding to circles with different densities, as shown in Figure 1.2. Like presented in Chapter 1, the larger cluster contains 31 points and the smaller one 19 points. To picture how the algorithm progresses

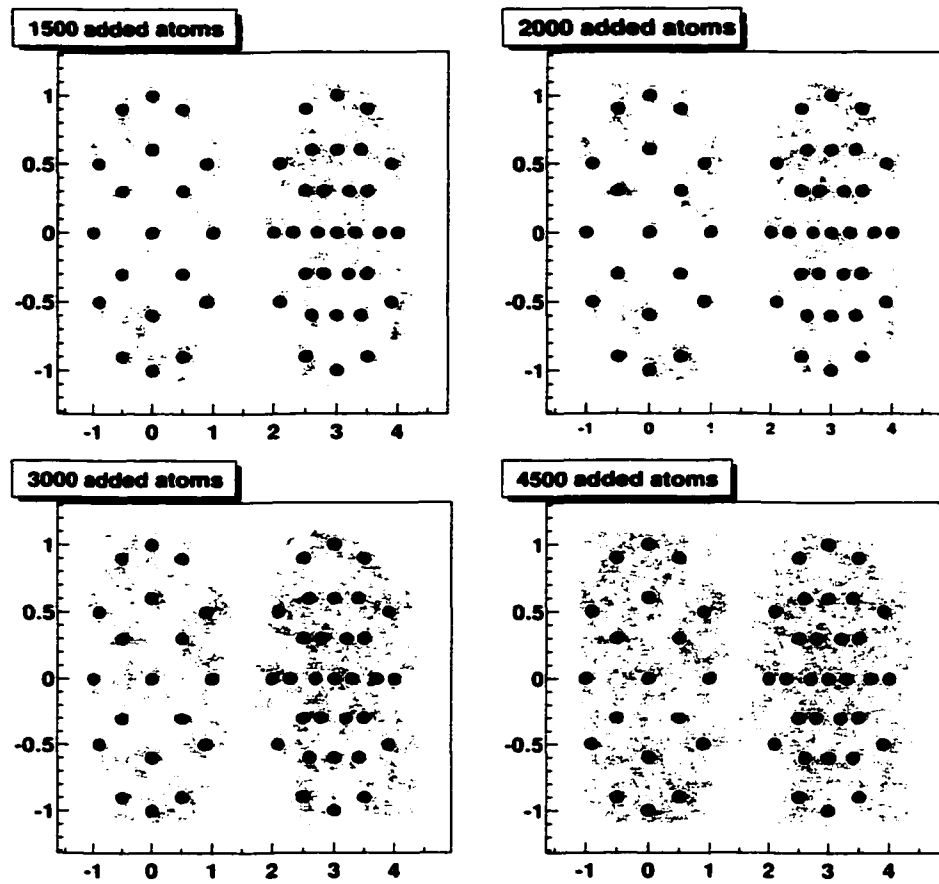


Figure 3.1: Snapshots showing the configurations as more and more deposited “ad_data” are introduced, corresponding to the two-dimensional data shown in Figure 1.2.

the actual configurations are shown as “snapshots” in Figure 3.1, with seed-data in solid symbols and ad_data in open symbols. Note that the two circular clusters have now been distorted into ellipses for ease of graphical

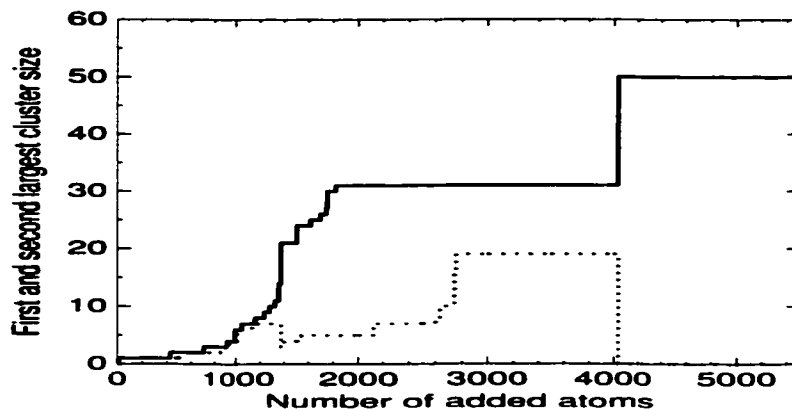


Figure 3.2: Size of the largest and second-largest islands as a function of the number of deposited ad_data for the two-dimensional data shown in Figure 3.1.

presentation. The actual distance functions used were based on the circles as shown in Figure 1.2. After 1500 accepted ad_data, which we will call depositions, have been introduced one can see how the rightmost cluster is almost totally connected, but not quite so. After 2000 ad_data depositions, the rightmost cluster is completely connected, but the leftmost cluster is still disjointed. After adding a further 1000 ad_data the leftmost circle has also been correctly identified. Finally, the bottom right figure shows how the two clusters have just merged after 4500 depositions. The output of the NGC algorithm is presented in Figure 3.2 where the graph shows the size of the largest island and the second largest island as a function of the number of deposited ad_data. One notes how the largest island size increases

monotonically, until it reaches a plateau after about 1800 particles have been deposited. Note that this plateau corresponds to 31 points being clustered, *i. e.* the rightmost circle has been identified. It is logical that this cluster is obtained first, since the density of points there is larger. The second largest cluster reaches a sustained plateau with 19 data points after approximately 2750 ad_data have been introduced. At that point the second cluster has also been correctly located. As more and more ad_data are deposited, finally the two clusters merge, which occurs after roughly 4000 depositions. These results illustrate quite convincingly how the method works on an admittedly simple problem.

We have tested our methodology on several other cases, including problems listed in References [12] and [19], and find good performance in all cases. To demonstrate the algorithm's behavior on a more complex case, we have studied the well-known iris data problem. This data set, described in Chapter 1 of this study, contains fifty examples each of three types of flowers: *Iris setosa*, *versicolor*, and *virginica*, for a total of 150 sample points. Each data point is characterized by four attributes, and a representation of these four-dimensional vectors in a plane spanned by the attributes petal length and petal width is given in Figure 1.5. Inspection of this figure as well as the representation of the data in a plane spanned by the first two principal components, given in Figure 1.6, shows that the *iris setosa* data are well separated from the other two, which are not so easily disentangled.

Figure 3.3 shows the result of the NGC algorithm on the iris data. Plotted are the size of the first three largest clusters as a function of time (number

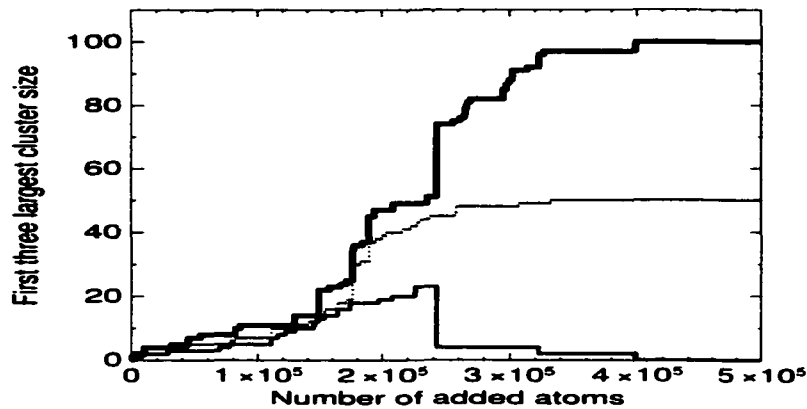


Figure 3.3: Size of the largest and second-largest and third-largest clusters as a function of the number of deposited ad.data for the iris data problem.

of depositions). Although the results are not as clear-cut as for the circles example (Figs. 3.1 and 3.2), one can clearly observe a small plateau. after 2.0×10^5 depositions near cluster size 48 (which then increases to 50), followed by a steep jump. Either a distinct plateau or a noticeable jump signals the presence of a cluster. We also observe a distinct plateau, which first emerges after 3.2×10^5 depositions, corresponding to 98 points. At this point the two clusters of iris versicolor and virginica have merged, with the third one, of iris setosa flowers, still detached.

To better understand the nature of the transitions shown in Figure 3.3. we found it instructive to look at the behavior of the third-largest cluster as a function of the number of deposited ad.data. One notices that the jump in largest cluster size in Figure 3.3 after 2.4×10^5 atoms corresponds to the

third largest cluster merging with the largest cluster. As a consequence, the size of the third largest cluster plummets quickly to zero. Correlating the behavior of several cluster sizes is frequently a good way to identify when a cluster is completely connected. Note that in this particular case, the second-largest cluster corresponds to the iris setosa data, which is easily classified. It shows a distinct plateau at 50 data points, without any further changes until very late in the simulation when all clusters merge. Hence the NGC algorithm reveals the existence of three groups, containing 50 (iris setosa), 46 (iris versicolor) and 23 (iris virginica) flowers, respectively. Some data points have not been classified, but we note that all prior methods based on a strict heuristic clustering approach are unable to correctly classify all points. Notice that the sizes of these last two intertwined groups are almost equal to the sizes found using the tertiary similarity function based Percolation Clustering Algorithm (47 and 20 respectively). This is due to the existence of a more compact group of members in each cluster and a number of entwined outliers.

The interesting behavior in this particular case, is to be found in the interplay of the largest and third-largest clusters. Because it may be infeasible to store too many cluster sizes, we have found it useful on occasion to run a simulation until one cluster had clearly emerged, as signaled by a sustained plateau or a large jump. At that point we would interrupt the simulation, locate the cluster in question, and remove it from the data set. We would then restart the clustering algorithm with the remaining data. This tended to work very well in practice and, although it is not easily automated, may be the method of choice for more complicated problems. For the case of the

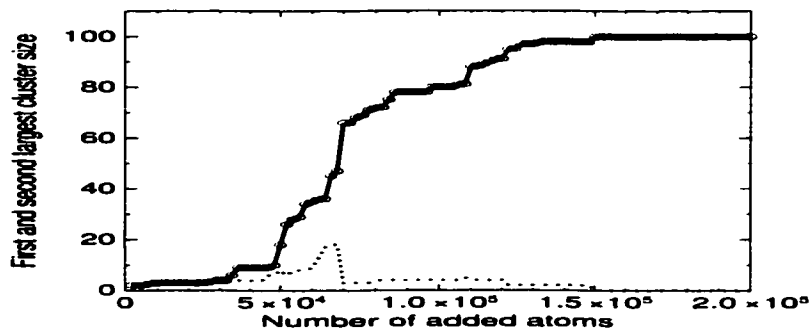


Figure 3.4: Size of the largest (solid line) and second-largest (dotted line) clusters as a function of the number of deposited ad_data for the 100 iris versicolor and iris virginica.

iris data problem, after eliminating the well defined iris setosa cluster, the NGC algorithm generates the results presented in Figure 3.4. It is easy to see that the third cluster increases up to 19 points and then connects with the other group. Due to the randomness of the ad_data deposition, the size of the third cluster is found to be between 22 (when the 150 flower file is analyzed) and 19 (when only the iris versicolor and iris virginica data is examined). To improve the quality of the partition two solutions might be considered. First is to choose a smaller interaction distance between seeds and ad_data, or even a distance dependent interaction. Second is to allow the system to escape from local minima by introducing a temperature dependent diffusion of the ad_data. The necessary computation overhead for such a simulation in a continuous space becomes cumbersome and therefore, in the following chapter we consider a discrete deposition and diffusion.

Chapter 4

Discrete Deposition Clustering Algorithm

The method described in the following sections is a heuristic non-parametric clustering technique inspired by the analogy with the deposition of atoms on a lattice. A system composed of data points and fictitious added particles is allowed to evolve in a self-organizing regime and the thermodynamic quantity specific heat at constant volume is used as a partition validity criterion. The algorithm is robust against the existence of noise and the final results are independent of initial conditions. This procedure differs from the one described in Chapter 3 by changing the deposition structure of the space from a continuum to a discrete lattice-type structure, which brings computational efficiency. In addition, the added particles can diffuse, or be added or removed, with a temperature dependent probability, which allows the system to escape from local minima.

4.1 Description of the Algorithm

The sample data, represented by points in a D -dimensional metric space, are considered as vertices of an undirected graph whose edges connect sample points no further apart than a cutoff distance d_c . Thus vector points further apart than the threshold distance remain unconnected. Each new data set may require a fine tuning of the cutoff distance, but since in general the inter-cluster distances are larger than the intra-cluster ones, a good choice for the cutoff length is the average distance between all sample points. This option implies that it is most likely that the points in the same cluster are connected while the number of inter-cluster links are minimized. Nevertheless, for data sets with a small number of elements, a larger cutoff distance builds a greater number of deposition sites. A larger number of deposition sites in turn allows the system to be better represented by a thermodynamic approximation. The sparse graph spanned by the data points represents the “lattice” on which the deposition takes place. This is, of course, not a regular lattice as in crystallography, but we use the same terminology to emphasize the similarity and to exploit the analogy. The mid points of the graph edges, called “*dual sites*”, represent the deposition locations. Figure 4.1 presents such a graph spanned by five data points i, j, k, m, n , represented by solid circles as the graph’s vertices. Each of the six edges, drawn as lines, contain at the mid point a dual site, symbolized by an empty circle. Dual sites are assigned the indices of the two boundary data points, *i. e.* $\{i, j\}$. As one can notice, there are no direct links between the pairs of data points placed farther apart

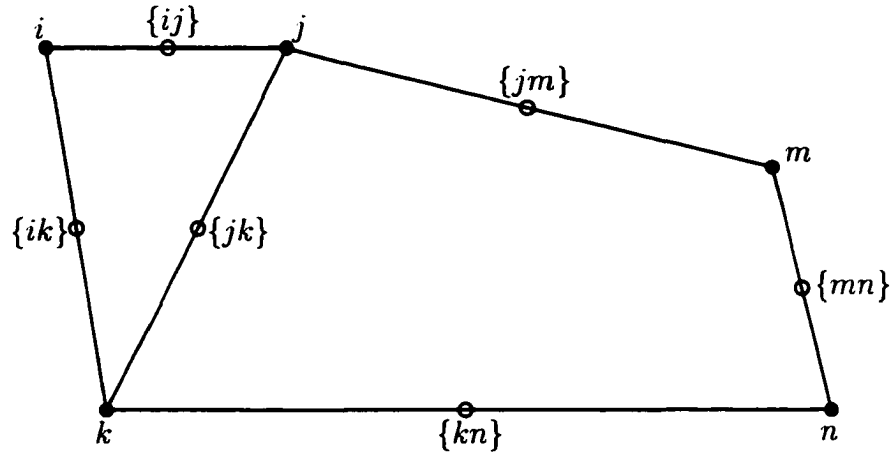


Figure 4.1: The sparse graph between five sample points i, j, k, m, n , represented by black dots, the graph's edges drawn as solid lines, and the dual sites denoted as, for example, $\{ij\}$ and symbolized as empty circles.

than the chosen cutoff distance d_c . Such pairs as i and m or i and n have no deposition sites available between them.

Fictitious points, called “*ad_data points*”, are randomly deposited on the dual sites merely as a way to simulate interaction between sample points. Since there is no way to predetermine the number of initially required deposition particles, we start by depositing a number of *ad_data* points equal to the number of sample points. The total number of *ad_data* points varies during the simulation according to the conditions described below.

Let us consider two data points i and j represented by the D -dimensional vectors \mathbf{x}_i and \mathbf{x}_j respectively, placed at a distance $d_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$. In a

two-dimensional space these vectors can be the vertices i and j of the graph represented in Figure 4.1. The distance is assumed to be a measure of the dissimilarity between the vectors. In our simulations we used a Manhattan distance defined by equation (1.3). If an ad_data point is situated on the dual site $\{ij\}$ between the two sample points i and j (see Figure 4.1), it introduces an attractive interaction, E_{ij} , whose absolute value should be a decreasing function of distance. A possible choice is:

$$E_{ij} = -\frac{1}{d_{ij}}. \quad (4.1)$$

The functional dependence of interaction strength on the dissimilarity (distance) between sample points will not affect the final partition, but rather the convergence of the algorithm and the sharpness of the phase transition. An alternative to the long range interaction described by equation (4.1) is the following expression:

$$E_{ij} = -\frac{1}{d_{ij}^2}. \quad (4.2)$$

Short range interactions such as:

$$E_{ij} = -\exp(-d_{ij}^2), \quad (4.3)$$

or

$$E_{ij} = -\exp\left(-\frac{d_{ij}^2}{a^2}\right). \quad (4.4)$$

where a is a local distance, can also be used. Because the densities of different clusters can be quite diverse, the short range interactions typically need fine tuning and a local specific distance, a , is used to normalize the absolute distance between sample points. This local length a varies from one vector

to another and is usually chosen to be the average distance to the first 3-5 nearest neighboring pairs. As such calculations create a lengthy overhead on the simulation, and since we are mainly interested in the general behavior of the described system in a self-organizing regime, we choose the simple interaction expressed by equation (4.1).

After the initial ad_data point deposition the system, considered to be in contact with a heat bath, is allowed to evolve toward thermal equilibrium. A Monte Carlo simulation [45, 46] generates configurations according to a Boltzmann distribution in the grand-canonical ensemble (*i. e.* with variable particle number). The deposited particles can diffuse, be added or be deleted according to the conditions described below. Ad_data points can move around to adjacent unoccupied dual sites. Adjacent dual sites are defined as the mid-points of the edges with a common vertex. As shown in Figure 4.1 the dual sites $\{i, j\}$ and $\{i, k\}$ are considered adjacent since they share the common data point i . Consider the previously mentioned ad_data point placed on the dual site $\{ij\}$. A jump to a neighboring site $\{ik\}$, $\{jk\}$ or $\{jm\}$, is always accepted if the newly occupied dual site introduces a stronger interaction:

$$E_{ik} = -\frac{1}{d_{ik}} < E_{ij} = -\frac{1}{d_{ij}}, \quad (4.5)$$

or correspondingly $E_{jk} < E_{ij}$, etc. So the system Hamiltonian (energy)

$$H = \sum_{i=1}^{\tilde{N}} \sum_{j=1, (j < i)}^{\tilde{N}} E_{ij}, \quad (4.6)$$

where $i, j = 1, \dots, \tilde{N}$ and \tilde{N} is the total number of ad_data points, decreases

as shown:

$$\Delta H = E_{ik} - E_{ij} < 0. \quad (4.7)$$

This condition obviously corresponds to the situation when $d_{ij} > d_{ik}$ (or $d_{ij} > d_{jk}$). Therefore the ad_data point will tend to introduce attraction between less dissimilar sample points. The contact with a heat bath at temperature T allows the system to escape from local minima, offering ad_data points a free energy that permits “uphill” moves. Therefore the transition from a dual site $\{ij\}$ to another deposition site $\{ik\}$ is also possible even if $\Delta H = E_{ik} - E_{ij} > 0$ with a probability p :

$$p = \exp\left(-\frac{\Delta H}{k_B T}\right), \quad (4.8)$$

where k_B is the Boltzmann constant, henceforth set equal to unity. These acceptance criteria are known as the Metropolis algorithm [18].

Given that the initial number of deposition particles is not strictly determined with respect to the total number of ad_data points necessary to group the sample points into clusters, we found it useful to work in the grand-canonical ensemble and vary the number of ad_data points. Utilizing the chemical potential $\mu > 0$, a deposition on a randomly selected empty site $\{kl\}$ generates a variation of the system’s Hamiltonian:

$$\Delta H = -\frac{1}{d_{kl}} + \mu, \quad (4.9)$$

while a deletion of a randomly chosen ad_data point creates a difference:

$$\Delta H = \frac{1}{d_{kl}} - \mu. \quad (4.10)$$

The additions and deletions are always permitted if they lower the Hamiltonian. In case they generate a variation $\Delta H > 0$, they are accepted with the probability described by equation (4.8).

The chemical potential is a very sensitive parameter whose value can produce a complete deletion of all ad_data points if μ is too large or a total occupation of the dual sites if μ is too small. Consequently it must be carefully adjusted by trial and error for each run of the simulation. The larger the number of deposition sites in the system, the less sensitive it is to the value of the chemical potential. Therefore, in the case of small data files it is easier to choose a larger value for the cutoff distance, hence a larger number of dual sites, than to fine tune the chemical potential.

In the physical phenomenon of atoms depositing on a lattice, diffusion is much more likely than addition or deletion. Therefore, during our simulation we set the diffusion of ad_data points to be one hundred times more likely than either addition or deletion. Thus every one hundred attempts to move an ad_data point is followed by one addition and one deletion attempt. Since there are no relevant criteria, a number of ad_data points equal to the number of sample points is initially deposited randomly on the dual sites. The system is then allowed to evolve toward thermal equilibrium by repeating attempts to move, add and delete ad_data points until the staggered averages of the system's Hamiltonian are equal.

Thus the thermal equilibrium criterion is fulfilled when the system's Hamiltonian averaged over the odd-number steps (H_o) equals the average calculated over the even-number steps (H_e) up to a tolerance of one-tenth of

one percent of their average as shown:

$$H_o - H_e \leq 0.001 \times \frac{H_o + H_e}{2}. \quad (4.11)$$

Once thermal equilibrium is reached, the simulation generates representative configurations for the statistical ensemble, but since it is impossible to cover the entire configuration space we replace it with a statistical sample of size M , as is common in the Monte Carlo method [45]. The number of steps needed for the system to reach thermal equilibrium as well as the size of the statistical sample is, obviously, larger for larger data sets. After thermal equilibrium is reached, each accepted jump, addition or deletion creates a new configuration which is considered for the statistical sample average. Various observables from statistical physics [30] can be computed at this stage. The quantity we are primarily interested in is the variance of the system's Hamiltonian over the statistical sample normalized by the temperature squared, in other words the specific heat at constant volume:

$$C_v = \frac{\overline{H^2} - \overline{H}^2}{T^2}, \quad (4.12)$$

where

$$\overline{H^2} = \frac{1}{M} \sum_{k=1}^M H_k^2 \quad (4.13)$$

and

$$\overline{H}^2 = \left(\frac{1}{M} \sum_{k=1}^M H_k \right)^2. \quad (4.14)$$

Such a model exhibits two phases. At high temperature the system is disordered and the fluctuations of the system's Hamiltonian are large, but the

specific heat remains rather small due to the temperature squared in the denominator of equation (4.12). At low temperatures the system is in an ordered state and the variance of the system's energy is very small even compared to the the temperature, therefore the specific heat is small. In an intermediate regime, reached as we sweep through temperatures, there is a certain point at which the system undergoes a phase transition. The compact groups of sample points become relatively strongly coupled, the intra-cluster deposition sites are filled and any change in the positions of ad_data points induces large variations of the Hamiltonian. At this point the variance of the Hamiltonian is large relative to the temperature and the specific heat encounters a peak. Thus the temperature where the peak of specific heat occurs indicates when some of the internal structure of the system emerges. This is an analog of the phase transition encountered in statistical physics [30], similar to a magnet becoming magnetized or a liquid freezing. In principle one can have a sequence of several such transition as the clusters split into smaller ones. Such a situation indicates a hierarchical structure of the data set.

One of the intricacies of the non-parametric technique is finding a cluster validity criterion, a parameter that could provide the most "natural" partition. Our technique uses the thermodynamic quantity specific heat as such a parameter and identifies the clusters at a temperature slightly lower than the critical temperature. We start the simulation at high temperature where the system's memory is insignificant and apply a simulated annealing procedure. The fictitious temperature is lowered, the system is allowed to reach thermal

equilibrium and then the Monte Carlo simulation generates a large number of configurations for the statistical sample. The specific heat is averaged over this sample according to equations (4.12), (4.13) and (4.14).

4.2 Computational Details

The main entities of the model are the data points, the deposition sites and the `ad_data` points. Each data point is represented by a structure that contains the array of coordinates, a label, a group index which indicates the cluster it belongs to (initialized to -1 for unattached sample points), a pointer link that allows the addition to a cluster-list, and a vector of neighbors which contains the labels of vertices the point is connected to. The elements of the dual sites structure are the two labels of adjacent vertices, the distance between them, and a flag to indicate if the site is occupied or not. An `ad_data` point is represented by one integer that is the index of the dual site vector it occupies.

Arrays are employed to store the two sets of elements with known size: the number, N , of sample points and the D coordinates for each of them. The cutoff distance is a parameter that can be changed from one run to another, therefore the number of graph edges and the number of dual sites as well as the degree of any vertex varies. Similarly the number of `ad_data` points fluctuates during the simulation. Thus from the rich variety of containers provided by the Standard Template Library (STL) of C++, we chose vectors to store the deposition sites and the `ad_data` point sets. As linear contiguous

storage, vectors are similar to arrays but have the capacity to expand their sizes at run time [35]. Vectors are also used to load the labels of all points connected to a certain vertex.

The first function of the program initializes the array of sample points and their coordinates. The next function calculates the distances between all sample points and connects them according to the condition $d_{ij} < d_c$. The function also builds the dual sites vector and the vector of neighbors for each point. The simulation continues by randomly filling a chosen number, nr , of dual_sites and creating the ad_data points vector.

The main tasks of the algorithm are performed by three additional functions. The *move* function arbitrarily picks an ad_data point to move and shifts it if there is any available empty dual site and the conditions (4.7) or (4.8) are met. The *add* function randomly selects an empty deposition site and, if the conditions (4.9) or (4.8) are satisfied, fills it with a binding particle. The *deletion* function arbitrarily chooses an_data and removes it from the system in the conditions of equations (4.10) or (4.8). The preceding probability condition, (4.8), is fulfilled by a calculating $\exp(-\Delta H/T)$ at each temperature and by randomly generating a number p between zero and one. If $p \leq \exp(-\Delta H/T)$ the change is accepted.

For each *add* and *delete* function, the *move* function is called one hundred times. This cycle is repeated a number of times dependent upon the data set size, until the system reaches thermal equilibrium as determined from the staggered average of the system's Hamiltonian. At this point, the algorithm is repeated M times, M depending on the size of the data set, to build a

statistical sample set. From this statistical sample set, the specific heat at constant volume, C_v , is determined for each temperature.

Some auxiliary functions are also used to visualize the process and analyze the validity of the simulation. Thus the number of ad_data points, both inter-cluster and intra-cluster, as well as the fraction of accepted moves, additions and deletions are output.

4.3 Computational Results

The way the method works will be illustrated on two of the data files already presented in the first chapter, respectively the two-dimensional toy problem consisting of 50 points grouped in two circles and the four-dimensional iris data problem.

The description of the simulation applied to the first two-dimensional data set allows visualization and a good understanding of the model. The distance between the centers of the two circular clusters, of radius $R = 1$, is 3 and the average distance between the 50 sample points is 2.46. Opting for a cutoff length approximately equal to the average distance between data points, $d_c = 2.5$, generates 342 deposition sites which can easily be filled by or depleted of ad_data points during the simulation. Therefore we chose a threshold of $d_c = 3.5$, which creates 892 dual sites and prevents the complete deposition or deletion of ad_data points, making the simulation more stable with respect to different values of μ . By trial and error we select a chemical potential $\mu = 0.8$. The simulation starts at the temperature $T = 2$ by

	Nr.Dual sites	Initial nr of ad_data T=2	Nr.ad_data at equilibrium T=2	Nr. ad_data at equilibrium T=0.01
Cluster 1	171	8	63	68
Cluster 2	456	27	195	236
Inter-clusters	256	15	84	3
Totat nr.	892	50	342	307

Table 4.1: Dual site arrangement and ad_data point repartition at two different temperatures for the 50 point two-dimensional data set.

depositing 50 ad_data points and applying a simulated annealing in steps of $\Delta T = 0.01$. For each temperature the system is allowed to reach thermal equilibrium by performing 20,000 attempts to add or delete ad_data points and $100 \times 20,000$ attempts to move a deposition particle. Table 4.1 lists the number of dual sites belonging to each cluster and to the inter-cluster space. The 50 initially deposited ad_data points are placed on the deposition sites according to a uniform distribution. Allowing the system to reach thermal equilibrium at $T = 2$, the number of ad_data points increases as shown in the table. Also illustrated is the arrangement of the ad_data points at thermal equilibrium for the lowest temperature attained. Notice that at thermal equilibrium at temperature $T = 2$ the proportionality between dual sites and deposited ad_data points still exists, which means that the distribution

of ad_data points is arbitrary. However, this is not the case at thermal equilibrium for temperature $T = 0.01$ when the deposition particles occupy mainly the intra-cluster sites.

The “snap-shots” of the system arrangements for different temperatures are shown in Figure 4.2. As the temperature is lowered the deposition particles move onto the lower energy dual sites, the intra-cluster ones. Therefore, as mentioned above, at low temperature the ad_data point does not follow the proportionality of the dual sites. The interplay between inter-cluster and intra-cluster ad_data is also monitored in the graph in Figure 4.3. The number of deposited particles for different temperatures is recorded at thermal equilibrium. As one can see, the two groups of intra-cluster ad_data points have almost a constant size, up to the critical temperature. The inter-cluster depositions decrease starting from $T = 1.5$ and become almost zero below the critical temperature.

A way to check if thermal equilibrium is reached is to monitor the fraction of accepted additions and deletions for each temperature. Figure 4.4 shows these fractions for different temperatures before thermal equilibrium is reached. Notice that the number of additions of ad_data points at the beginning of the simulations is larger than the deletions. Approaching the critical temperature T_c , the number of deletions exceeds the number of additions, due to the elimination of inter-cluster ad_data points. Once the system reaches equilibrium and the algorithm generates representative configurations for the statistical sample the number of ad_data points is almost constant, therefore the additions approximately equal the deletions. Figure 4.5 shows

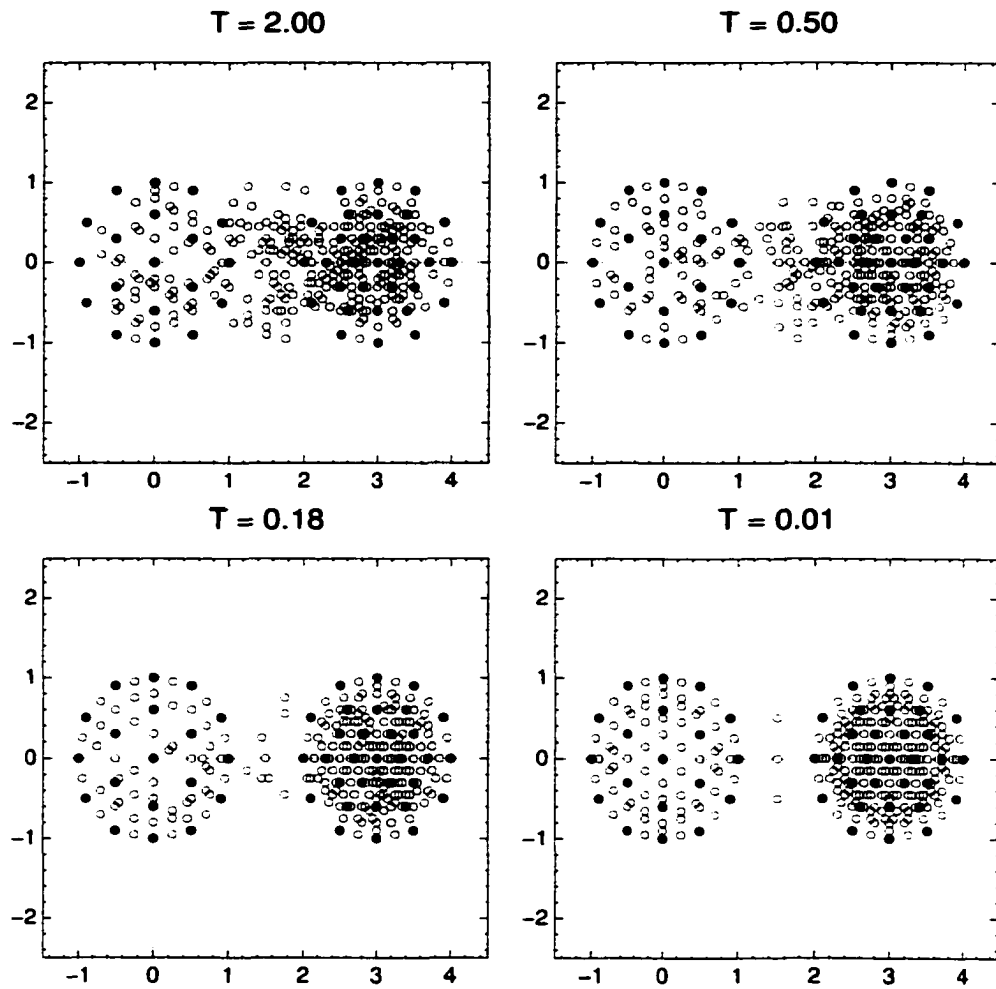


Figure 4.2: “Snap-shots” of the 50 point two-dimensional data set grouped in two circular clusters and the ad_data points configuration at thermal equilibrium for different temperatures. Notice the number of inter-cluster ad_data points decreasing with decreasing temperature.

the fraction of additions and deletions out of 20,000 attempts for temperatures $T > 1.5$; for lower temperatures this difference is insignificant.

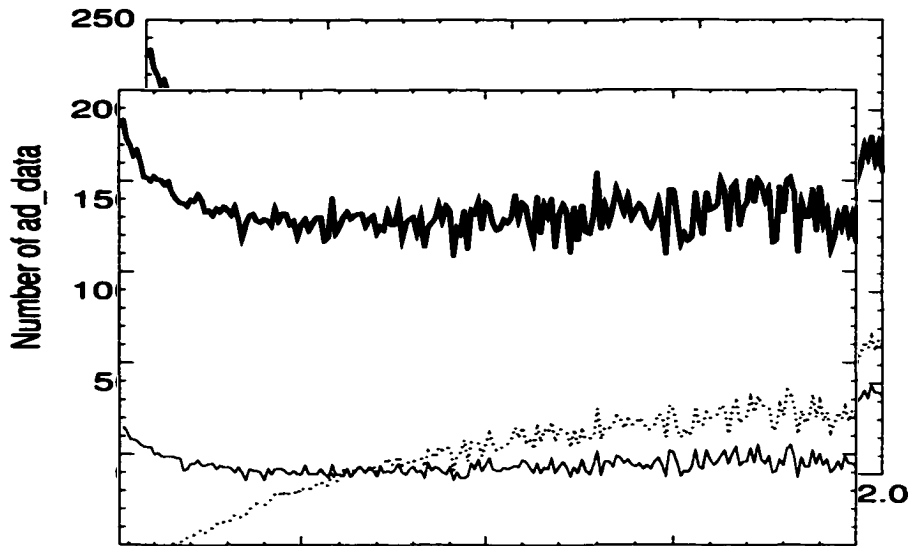


Figure 4.3: Number of ad_data points in cluster 1, (lower thin line), in cluster 2 (upper thick line) and between the clusters (dotted line) for the 50 point two-dimensional toy problem. Notice the relatively constant number of intra-cluster deposition particles up to critical temperature $T_c \simeq 0.18$. The number of inter-cluster ad_data points decreases as the temperature is lowered from $T = 1.5$.

Once we have a feeling for the system behavior, the variation of the specific heat at constant volume as a function of temperature, presented in Figure 4.6, becomes a relevant criterion for a “natural” partition of the data. From this figure the critical point can be seen to be near the temperature $T_c \simeq 0.18$ when the specific heat reaches its maximum value of $C_v = 218$. The image of the system configuration near the critical point (Figure 4.2) shows how the intra-cluster deposition sites are almost filled and the inter-

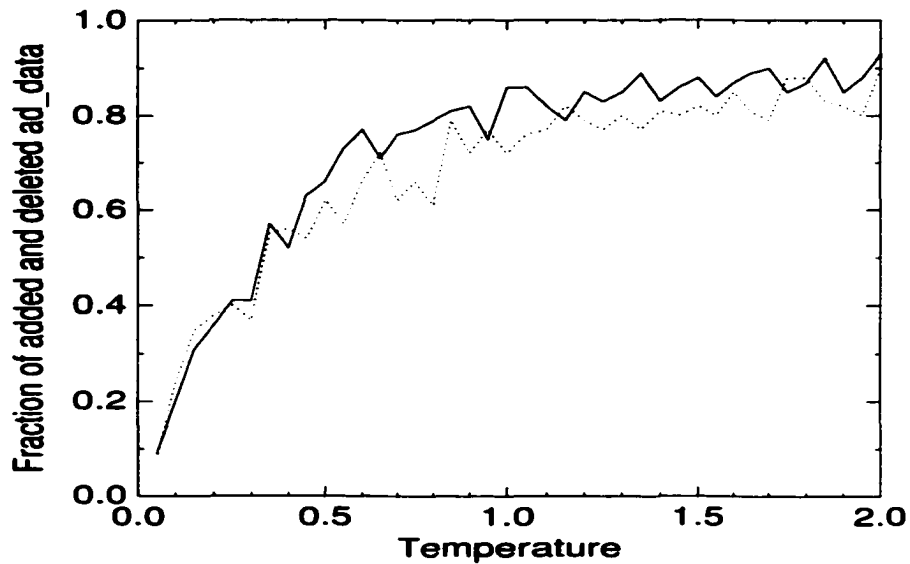


Figure 4.4: Fraction of added (continuous line) and deleted (dotted line) ad_data points averaged over 100 attempts before thermal equilibrium for 50 point two-dimensional toy problem.

cluster ad_data points occupies already similar positions to the ones attained at the lowest monitored temperature $T = 0.01$.

For similar situations to the one under discussion, when the system encounters one phase transition, the clustering is done at the lowest temperature achieved. This particular data file has a high symmetry, which usually raises problems of local minima for many clustering algorithms. As we can see from the configuration of the system illustrated in Figure 4.2, even at $T = 0.01$ there are three inter-cluster deposited ad_data points. This is due to the fact that the centers of each circle are at the same distance from the

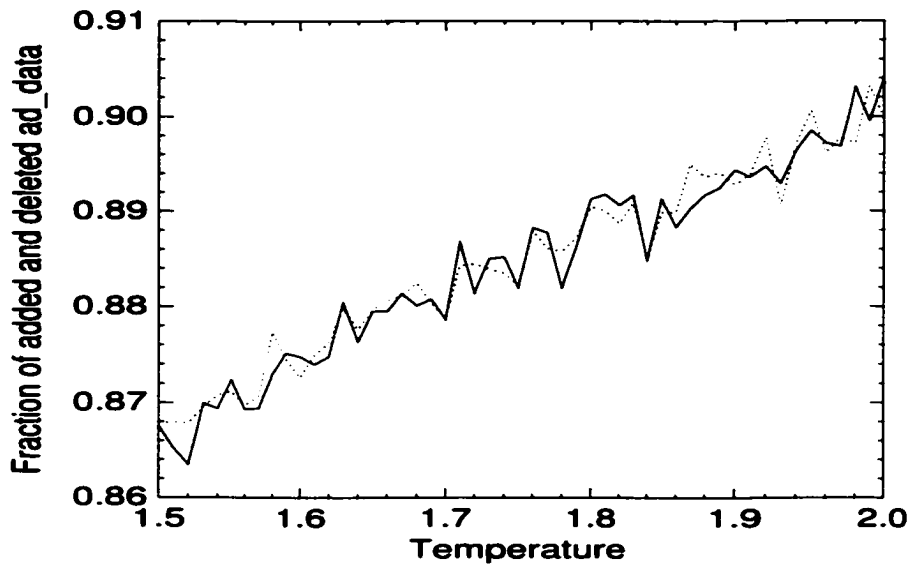


Figure 4.5: Fraction of added (continuous line) and deleted (dotted line) ad_data points out of 20,000 attempts for 50 two-dimensional points at thermal equilibrium.

points on the circumference as the distances between points at the extreme right of the leftmost circle and the extreme left of the rightmost one. To avoid such a lock in, the clustering is done by considering for each sample only a desired number of closest ad_data points or, similarly, eliminating a chosen number of furthestmost ones. For the 50 point two-dimensional toy problem we eliminate two of the furthestmost ad_data points for each sample and the partition obtained is identical to the original one presented in Figure 1.2.

Let us consider for now the same circular clusters of radius $R = 1$ placed this time with the centers a distance of 3.5 units apart. Since the inter-cluster

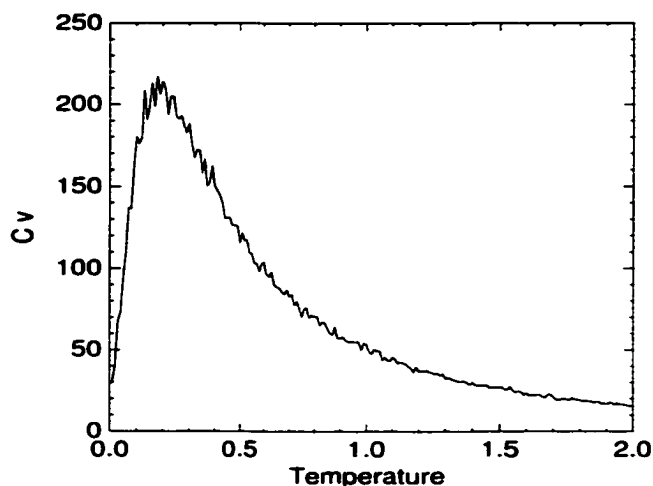


Figure 4.6: Specific heat at constant volume as a function of temperature for the 50 two-dimensional points data set. The specific heat reaches a maximum value of approximately $C_v = 218$ near the critical temperature $T_c \simeq 0.18$ after which it declines abruptly.

distance is larger in this case, the simulation provides a more clear-cut result. The critical temperature is $T_c \simeq 0.16$ and the configurations of the system at critical temperature as well as at the lowest reached temperature are illustrated in Figure 4.7. Notice the total absence of inter-cluster ad_data points. In this situation it is not necessary to eliminate any deposited particles. Nevertheless even by disregarding the two furthestmost ad_data points for each sample point we obtain the correct partition of the data. This approach seems to work well in more general cases and eliminates some of the possible problems related to local minima.

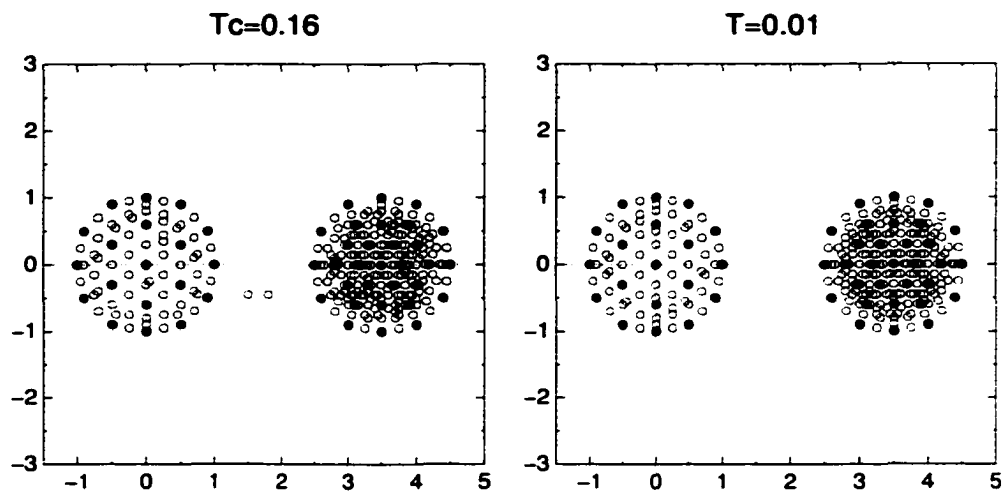


Figure 4.7: “Snap-shots” of the 50 point two-dimensional data set regrouped into two circular clusters with centers placed 3.5 units apart and the `ad_data` configuration at thermal equilibrium for critical temperature and lowest temperature achieved. Notice the low number of inter-cluster `ad_data` points.

The second file, chosen to test the Discrete Deposition Clustering Algorithm, the iris data problem is a more complex one. Not only are the sample points four-dimensional, but also two of the three clusters are interconnected. The specific heat as a function of temperature for the iris data problem is presented in Figure 4.8. Clustering performed at the lowest temperature achieved shows one cluster of 50 flowers, corresponding to iris virginica, and a second one of 100 flowers, representing the iris setosa and versicolor groups. In order to obtain three separate groups, the clustering has to be done immediately after the critical temperature, but before the two intertwined groups connect. Partitioning done at a temperature of $T = 0.15$ reveals the existence

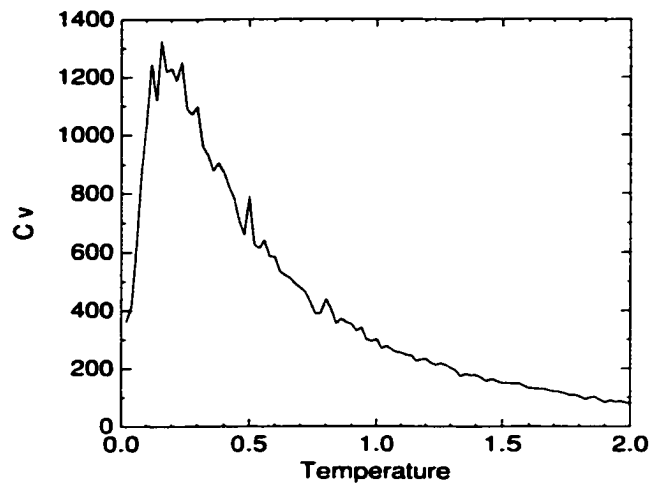


Figure 4.8: Specific heat at constant volume function of temperature of iris data problem.

of three clusters, containing 50, 42, and 54 flowers respectively, corresponding to the three iris groups. This result is comparable to ones obtained using other heuristic techniques and has the advantage of computational efficiency and a clear clustering criterion, *i. e.* a well defined phase transition.

Chapter 5

Application of Clustering to Econophysics

In this chapter we apply some of the previously described data mining procedures to financial data. The Percolation Clustering Algorithm, presented in Chapter 2, is used to identify “natural” classes of stocks and examine portfolio taxonomy. Once the clusters of stocks are determined, the logical follow up is probing their time stability. By examining the stocks’ correlations during different time intervals we detect significant changes occurring in the assets cross-correlation matrix during volatile market conditions.

5.1 Introduction

Data mining algorithms are indispensable to financial data analysis. Present-day technology produced not only a major rise in the market participation.

but also an increased data availability. Due to fast execution and low commissions, the volume of shares traded daily on the New York Stock Exchange (NYSE), the American Stock Exchange (AMSE) and the National Market System (NASDAQ) increased by a factor of 100 times in the last 25 years. Simultaneously, large amounts of more and more detailed information about market transactions are collected daily. For instance, starting from 1993 all transactions made on any of the world's major exchanges are recorded "tick-by-tick" (down to the bid and ask prices). As a result of the continuous diversification of the traded assets (stocks, indices, mutual funds, annuities, etc.) as well as of their derivatives (futures, options), it is not only the amount of data that increases but also its complexity. Since the nature of the randomness and interactions that move the markets are not completely known, there is no stipulated method of analyzing the complexity of all this data.

Among different directions of research dealing with financial data, an emerging discipline developed in the 1990s: *Econophysics*. Emphasizing the empirical analysis of the large amount of available economic data, physicists exploit similarities between statistical laws in physics and in financial markets. Concepts such as scaling, renormalization, self-organized systems, critical phenomena, etc. have been conveyed as new tools in modeling financial and economic data [47].

In order to describe and anticipate the market behavior numerous studies examine the time series of the assets' price fluctuations. We direct our attention to the correlations between stock price variations as an indica-

tor of market conditions. Specifically, by applying the previously described Percolation Clustering Algorithm, we try to determine “natural” classes of correlated stocks and their stability in different market conditions.

5.2 Data

We investigate two five-year-long intervals: 1986-1990 and 1997-2001. In the first interval, twenty-six major US company stocks are tracked, and in the second period thirty Dow Jones Industrial Average (DJIA) components are examined. The index components change over time. For instance, Citigroup (C) replaced Travelers (TRV) after the merger between Citicorp and Travelers in October 1998. Additionally, on November 1, 1999, Intel (INTC), Microsoft (MSFT), Home Depot, Inc. (HD), and SBC Communications, Inc. (SBC) were added to the Dow Industrials. Intel and Microsoft were the first NASDAQ companies to ever be included in the Dow. In the same year, the following companies were deleted from the Dow Industrials: Chevron Corp. (CHV), Goodyear Tire & Rubber Co. (GT), Sears, Roebuck & Co. (S), and Union Carbide Corp. (UK). The thirty stocks tracked by the DJIA in 2001, listed in Table A.2 in Appendix A, were considered representative for the period 1997-2001.

One realizes the major improvement encountered lately in data availability when confronted with the study of past intervals. For instance, during the period 1986-1990 there are good records of the DJIA daily variations but insufficient ones regarding its constituents. During this interval the com-

ponents of the DJIA did not change; however, many of these companies, nowadays, folded or merged with other companies. For them there are few or no historical records available that have been updated, *i. e.* accounted for splits or reverse splits. Nevertheless, the index tracks large cap corporations with long records of consistent good performance. therefore the components of the index in 1991, listed in Table A.1 in Appendix A together with their ticker symbol and the primary group they belong to, are considered representative for the interval 1986-1990. Twenty-six of the DJIA components for the year 1991 have good updated records for the interval 1986-1990 and have been analyzed during this period. They are written in a regular font in the Table A.1. The other four constituents, listed in italic, Allied Signal (ALD), Bethlehem Steel (BS), Union Carbide (UK) and Westinghouse (WX) have short or nonexistent records and have been ignored.

An interesting remark is that, regardless of the differences in DJIA constituents, the statistical properties of the correlation coefficients between its elements remain similar during analogous market conditions. This behavior reflects an underlying characteristic of the studied assets, independent of the industry they belong to: they are the “blue chips”, the largest publicly traded companies in U.S. Their market value represents between 15% and 20% of the total worth associated to more than 2000 securities listed on the NYSE. We restrict our attention to these particular stocks due to their similarly large market caps, which confers to them a relative stability. An asset’s volatility is defined as the standard deviation of the asset’s returns time series. It has been established that a stock’s volatility decreases with

its increasing market capitalisation (total value of all its outstanding shares) [48], therefore shorter time series are needed to obtain statistically significant information for large cap items. To investigate the interaction between assets we tracked their price variation time series for fixed time intervals chosen long enough to provide statistical relevance, but sufficiently short to accommodate similar market conditions. Moving toward a common ground with the economists' points of view, we analyze the DJIA components quarterly and annually for the two intervals, 1986-1990 and 1997-2001. During these periods the index encountered some of the largest declines in history. What makes these changes even more interesting, besides their current interest, is the variety of causes that induced them. Some were triggered by internal market dynamics and had no immediate connection with political or social events. This is the case of the 508 point decline encountered by the DJIA index on October 19th 1987, the largest daily plunge ever. Another example is the 15% fall of index value during the first two months of the year 2000, a decline that started the "bear" market we are currently in. Other market declines encountered during these periods are due specifically to extraordinary events such as the unexpected terrorist attack on September 11, 2001.

The daily closing prices of the DJIA during the interval 1986-1990 are represented graphically in Figure 5.1. As one can see the period starts with strong market growth sustained for more than a year and a half, followed by a sharp decline during mid October 1987. The plunge lasts only several days after which the market builds a new base and starts rising again for the next two years (1988 and 1989). Out of this five year interval, the market ends

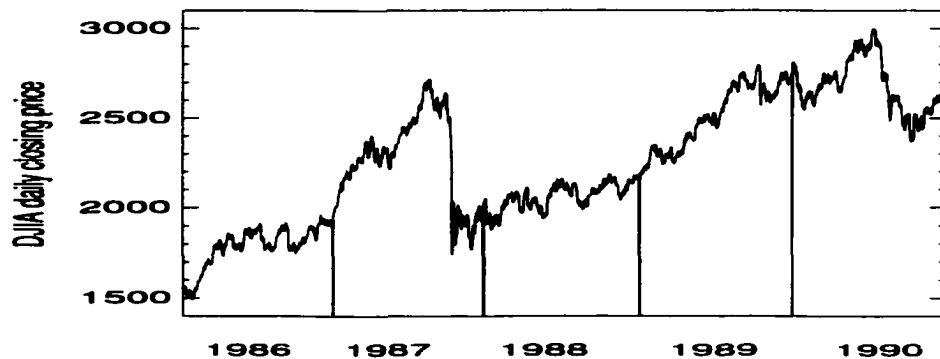


Figure 5.1: DJIA daily closing price during the period 1986-1990. The vertical lines delimit one year from the next.

lower than it started only during 1990, which is a year of high volatility.

A summary of market performance throughout the period 1986-1990 is presented in Table 5.1, which contains the DJIA daily closing price for the first and the last trading day of each quarter, in addition to the quarterly and annual percentage variation of the index.

The second interval, 1997-2001, contains the transition from the longest “bull” market in history to a “bear” market. The general market behavior during this period is illustrated by the daily closing prices of the index, presented in Figure 5.2. With small exceptions the period 1997 through 1999 brought a significant increase in the market. In contrast, starting in the beginning of the year 2000 the DJIA’s value declines and continues to wane for the rest of the remaining interval. Thus, the first three years are particularly “bullish”, as often happens before a market crash, while the last

Year	Quarter 1	Quarter 2	Quarter 3	Quarter 4	Annual (%)
1986	1538	1790	1903	1783	+23.2%
	1819	1893	1768	1896	
	+18.3%	+5.7%	-7.1%	+6.3%	
1987	1927	2316	2410	2639	+0.6%
	2305	2418	2596	1939	
	+19.6%	+4.4%	+7.7%	-26.5%	
1988	2015	1981	2132	2105	+7.6%
	1988	2142	2113	2168	
	-1.3%	+8.1	-0.9%	+3.0%	
1989	2145	2305	2453	2714	+28.3%
	2294	2440	2693	2753	
	+6.9%	+5.8%	+9.8%	+1.4%	
1990	2810	2700	2899	2516	-6.3%
	2707	2881	2452	2634	
	-3.7%	+6.7%	-15.4%	+4.7%	

Table 5.1: DJIA daily closing price for the first and last trading day of each quarter, as well as quarterly and annual percentage variation of these prices during the interval 1986-1990.

two are “bearish” years. This classification is encapsulated in Table 5.2, which presents the DJIA daily closing price for the first and the last trading day of each quarter as well as the quarterly and yearly variation of the index for the period 1997-2001.

In order to operate with homogeneous data, we had to discard several trading days during the interval 1997-2001. Out of the 30 DJIA components, GE in 2000 required the deletion of Sept 5th through 8th; and HD, JNJ, and JPM necessitated the elimination of March 1st. Similarly, in 2001 UTX, WMT, and XOM needed the deletion of May 23rd and 24th. Table 5.3 lists

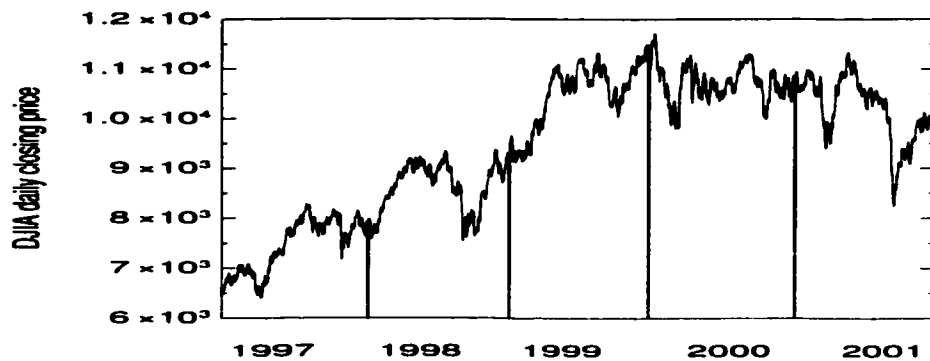


Figure 5.2: DJIA daily closing price during the period 1997-2001. The vertical lines delimit one year from the next.

the number of trading days considered during each quarter and each year for the intervals 1986-1990 and 1997-2001.

To discover classes of similarly performing stocks we analyze the correlations between simultaneous variations of assets' prices. The quantity we are primarily interested in is the asset *price change*:

$$Z(t) = Y(t) - Y(t - \Delta t), \quad (5.1)$$

where $Y(t)$ is the asset price at time t and $Y(t - \Delta t)$ is the same quantity at a time Δt before. Since the price is expressed in dollars (or the currency of the country where the financial asset is traded) its unit varies in time. Price changes encountered at different periods or the price of the assets traded in different currencies become incomparable. A way to eliminate the scaling

Year	Quarter 1	Quarter 2	Quarter 3	Quarter 4	Annual (%)
1997	6448	6611	7722	8015	
	6583	7673	7945	7908	
	+2.1%	+16.1%	+2.9%	-1.3%	
1998	7965	8868	9049	7632	
	8800	8952	7843	9181	
	+10.5%	+0.9%	-13.3%	+20.3%	
1999	9184	9832	11066	10273	
	9786	10971	10337	11497	
	+6.6%	+11.6%	-6.6%	+11.9%	
2000	11358	10922	10561	10700	
	10922	10448	10651	10787	
	-3.8%	-4.3%	+0.8%	+0.8%	
2001	10646	9778	10594	8837	
	9879	10502	8848	10022	
	-7.2%	+7.4%	-16.5%	+13.4%	

Table 5.2: DJIA daily closing price for the first and last trading day of each quarter, as well as quarterly and annual percentage variation of these prices during the interval 1997-2001.

effect is to study the asset *return* defined as:

$$R(t) = \frac{Y(t) - Y(t - \Delta)}{Y(t - \Delta)} = \frac{Z(t)}{Y(t - \Delta)}, \quad (5.2)$$

where Δt is a time interval short enough such that the currency fluctuations are negligible. Previous studies of portfolio and risk management have shown that stocks with similar market caps interact mainly as equally weighted, rather than value weighted, assets [49]. Since we monitor the interaction between similarly large companies, the thirty largest in U.S., it is the absolute variation of their prices that is relevant and not the relative variation provided

year	1986	1987	1988	1989	1990
1st quarter	61	62	63	62	62
2nd quarter	64	63	63	64	62
3rd quarter	64	64	64	63	62
4th quarter	64	64	63	63	63
Total	253	253	253	252	249
year	1997	1998	1999	2000	2001
1st quarter	61	61	61	62	62
2nd quarter	64	63	63	63	61
3rd quarter	64	64	64	59	59
4th quarter	64	64	64	63	64
Total	253	252	252	247	246

Table 5.3: Number of analyzed days for each quarter and year in the chosen time intervals, 1986-1990 and 1997-2001.

by the return. To avoid discontinuities due to currency changes the variable chosen to be analyzed is the successive difference of the natural logarithm of prices:

$$S(t) = \ln Y(t) - \ln Y(t - \Delta t) = \ln \frac{Y(t)}{Y(t - \Delta t)}. \quad (5.3)$$

As previously mentioned Δt has to be an interval during which the the currency variations are insignificant, thus we choose it to be the time between successive trading days. For each one of the DJIA components we monitor, the daily logarithmic variation of the closing prices is given by

$$S(t) = \ln Y(t) - \ln Y(t - 1), \quad (5.4)$$

where $Y(t)$ is the asset closing price in one day and $Y(t - 1)$ is the same quantity at the end of the previous trading day. The logarithm difference has the property that it transforms an absolute variation into a fraction.

which is scale invariant, but it has the drawback that, being a nonlinear transformation, it generally changes the statistical properties of the data.

However some parameters, in particular the cross-correlation coefficients of simultaneous price changes, remain unaffected. The correlation coefficient between the time series $Z_i(t)$ and $Z_j(t)$ is the same as the correlation coefficient between the series $S_i(t)$ and $S_j(t)$.

It is interesting to note that when the price changes are very small with respect to the asset price, $Z(t) \ll Y(t)$, the change in the logarithm of prices $S(t)$ approximately equals the return $R(t)$. Knowing that $Y(t) = Y(t - \Delta t) + Z(t)$, equation (5.4) becomes:

$$S(t) = \ln[Y(t-\Delta t)+Z(t)] - \ln Y(t-\Delta t) = \ln\left[1 + \frac{Z(t)}{Y(t-\Delta t)}\right] \approx \frac{Z(t)}{Y(t-\Delta t)} = R(t).$$

The previous condition is usually fulfilled for high frequency data when Δt is small and in the absence of major market changes.

5.3 Portfolio Taxonomy

Determining classes of stocks with similar or opposite behavior is essential for risk management. Generally, in the case of major market moves, all assets in the market are positively correlated and follow a similar pattern, that is when some go up, or down, the others follow. However, the actual increase or decrease of assets' market value is different from one stock to another. Various factors can influence various sectors of the market differently. There are cyclical industries and in the absence of major global trends, there are

even groups of stocks that move in opposite directions. Assembling a portfolio of stocks that belong to different anti-correlated or weakly correlated groups can substantially minimize the risk.

To determine the similarity in the synchronous time behavior between two stocks i and j , during a chosen time interval T , we monitor the time series of logarithmic price changes S_i and S_j , respectively, defined by equation (5.4) and calculate the correlation coefficient of the two time series as

$$\rho_{ij} = \frac{\langle S_i S_j - \langle S_i \rangle \langle S_j \rangle \rangle}{\sqrt{(\langle S_i^2 \rangle - \langle S_i \rangle^2) (\langle S_j^2 \rangle - \langle S_j \rangle^2)}}, \quad (5.5)$$

where the symbol $\langle \rangle$ means the average over the T time records described as:

$$\langle S \rangle = \frac{1}{T} \sum_{t=1}^T S(t) \equiv \frac{1}{T} \sum_{t=1}^T S_t. \quad (5.6)$$

The above equality introduces the notation S_t as an alternative to $S(t)$ to underline the discrete character of the parameter t and of the time series' values. This notation will be used from now on throughout the entire chapter. Note that the symbol $\langle S \rangle$ denotes the average over time as opposed to the average over the sample ensemble defined as:

$$\bar{S} = \frac{1}{N} \sum_{i=1}^N S_i,$$

where N is the number of studied assets.

The definition (5.5) guarantees that for any i and j , the correlation coefficient has the following properties:

$$\rho_{ii} = 1 \quad , \quad \rho_{ij} = \rho_{ji} \quad \text{and} \quad |\rho_{ij}| \leq 1. \quad (5.7)$$

Current algorithms [47, 50] used to determine a portfolio taxonomy consider the time series of each stock as a vector in a T -dimensional space, where T is the length of the time series. The coordinates of such a vector are the logarithm price variations normalized to zero mean and unit variance. Hence stock i is represented by the vector $\tilde{\mathbf{S}}_i$ with the components S_{it} , given by:

$$\tilde{S}_{it} = \frac{1}{\sqrt{T}} \frac{S_{it} - \langle S_i \rangle}{\sqrt{\langle S_i^2 \rangle - \langle S_i \rangle^2}}, \quad (5.8)$$

where $S_{it} \equiv S_i(t)$ is the change in the logarithm of the price for asset i at time step t , described by equation (5.4) and $\langle S_i \rangle$ is the time average of this quantity over all trading days in the investigated interval T . Notice that even though each trading day represents a new dimension, the coordinate along this new direction depends on all other coordinates through the temporal average $\langle S \rangle$ and standard deviation $\sigma = \sqrt{\langle S^2 \rangle - \langle S \rangle^2}$. All coordinates are sensitive to the chosen investigated interval and change when adding or omitting time records. Another important remark is that, by definition, such vectors have unit magnitude:

$$|\tilde{\mathbf{S}}|^2 = \sum_{t=1}^T \tilde{S}_t^2 = \frac{1}{T} \frac{\sum_{t=1}^T (S_t - \langle S \rangle)^2}{\langle S^2 \rangle - \langle S \rangle^2} = 1. \quad (5.9)$$

An Euclidian distance between two vectors \mathbf{S}_i and \mathbf{S}_j is defined as:

$$d_{ij}^2 = \|\mathbf{S}_i - \mathbf{S}_j\|^2 = \sum_{t=1}^T (\tilde{S}_{it} - \tilde{S}_{jt})^2 = \sum_{t=1}^T (\tilde{S}_{it}^2 + \tilde{S}_{jt}^2 - 2\tilde{S}_{it}\tilde{S}_{jt}). \quad (5.10)$$

According to the property defined in (5.9), the above equation can be written as:

$$d_{ij}^2 = 2 - 2 \sum_{t=1}^T \tilde{S}_{it}\tilde{S}_{jt},$$

and, considering the definition (5.5) of the correlation coefficient ρ_{ij} , we obtain

$$\rho_{ij} = \sum_{t=1}^T \tilde{S}_{it} \tilde{S}_{jt}.$$

Therefore, the distance between the two stocks i and j becomes:

$$d_{ij} = \sqrt{2(1 - \rho_{ij})}. \quad (5.11)$$

As long as the two assets are positively correlated ($\rho_{ij} > 0$), it can be shown that the quantity defined by equation (5.11) has all the characteristics of a metric [47]:

$$d_{ij} = 0 \Leftrightarrow i = j \quad , \quad d_{ij} = d_{ji} \quad \text{and} \quad d_{ij} \leq d_{ik} + d_{kj}. \quad (5.12)$$

The first two properties are easily understandable based on the correlation coefficients properties (5.7). Squaring both sides, the third property can be written as:

$$1 - \rho_{ij} \leq 2 - \rho_{ik} - \rho_{kj} \quad \text{or} \quad \rho_{ik} + \rho_{kj} \leq 1 + \rho_{ij}.$$

The last inequality is satisfied only for positive correlation coefficients ($0 \leq \rho \leq 1$). Therefore an Euclidean distance described by equation (5.10) cannot be defined between anti-correlated stocks ($\rho < 0$).

To find the taxonomy of a portfolio containing N positively correlated assets, $\frac{N(N-1)}{2}$ Euclidean distances are defined and a symmetric distance matrix is built. From this point on, the classification can be achieved by applying any one of the many clustering procedures that use distance as a similarity function.

At this point it is important to underline the specifics of the stock classification problem as opposed to other clustering problems. The number of stocks is somewhat limited, and in the end all assets have to be partitioned. This is unlike, say, the pixels in image processing or other clustering problems that deal with large numbers of sample points, out of which some are part of the background and are considered noise. Therefore the main approach in contemporary research is to use a deterministic procedure to classify the financial assets: the minimal spanning tree (MST), known also as the shortest link algorithm (SL) [47, 50, 51]. The algorithm selects the shortest distance between successive vectors, creating an associated ultrametric hierarchical tree and an associated hierarchical classification (dendogram). The method proves its efficiency by producing economically meaningful taxonomies [47], [51], but the depiction becomes more and more cumbersome as the size of the analyzed portfolio increases. It is hard to identify “natural” classes of stocks and their time evolution. Another drawback of the procedure is that the distance can only be defined between positively correlated stocks.

In an attempt to overcome this difficulty and cluster simultaneously correlated as well as anti-correlated assets, another method has been used [52]: the Super-Paramagnetic Clustering algorithm (SPM) mentioned previously in Chapter 1. In this new context, the procedure is generalized by introducing two types of interactions: an attraction that tends to align the spins and a repulsion that favors different spin orientations. Out of q Potts spin values a random one is assigned to each asset and a ferromagnetic interaction is defined between positively correlated stocks, while an anti-ferromagnetic

interaction is considered for anti-correlated ones. Both types of binary interactions are defined based on the correlation coefficient between the assets, given by:

$$J_{ij} = \text{sgn}(\rho_{ij}) \left(1 - \exp \left\{ -\frac{n-1}{n} \left[\frac{\rho_{ij}}{a} \right]^n \right\} \right). \quad (5.13)$$

Performing a simulated annealing procedure, strongly correlated companies within the same group tend to align their spins while anti-correlated stocks “repel”, opting to point in different “Potts directions”. By monitoring the susceptibility of such a system as a function of temperature, one can detect peaks indicating super-paramagnetic transitions. During the super-paramagnetic phase the stocks within the same cluster have identical spin values, while separate clusters are in different spin states. Thus the clusters are identified by means of a spin-spin correlation function. Unlike the MST, Super-Paramagnetic Clustering is a heuristic approach. Since the clusters are identified by means of a spin-spin correlation function, large fluctuations are expected for small size systems and sometimes the obtained partition is not unambiguous [51]. The method also requires fine tuning of the three parameters q , n and a as well as burdensome calculations.

We propose to use the Percolation Clustering Algorithm to determine a portfolio taxonomy. This new approach necessitates two changes: a different similarity function and a new way to visualize the results. As mentioned in Chapter 1, the similarity function between two samples does not have to be a metric, but can be any monotonous symmetrical binary function [2]. Hence, instead of the distance or the correlation-based-interaction, we use

the correlation coefficient between the two T long times series, defined by equation (5.5) as a measure of the similarity between two items. Considering the vectors $\tilde{\mathbf{S}}_i$ and $\tilde{\mathbf{S}}_j$, with the components described by equation (5.8), the correlation coefficient can be seen as the normalized inner product of these vectors or, equivalently, the cosine of the angle between them. Thus the similarity function $s(\tilde{\mathbf{S}}_i, \tilde{\mathbf{S}}_j)$ is defined as:

$$s(\tilde{\mathbf{S}}_i, \tilde{\mathbf{S}}_j) = \frac{\tilde{\mathbf{S}}_i \tilde{\mathbf{S}}_j}{|\tilde{\mathbf{S}}_i| |\tilde{\mathbf{S}}_j|} = \sum_{t=1}^T \tilde{S}_{it} \tilde{S}_{jt} = \rho_{ij}. \quad (5.14)$$

In writing the above equality we used the property (5.9) and the definition (5.5). The angle between two vectors represents a meaningful measure of their similarity, especially when the vector's length is invariably unity.

Choosing the correlation coefficient as a similarity function minimizes the assumptions imposed on the data and avoids altogether embedding it in a vector space. Obviously the clustering procedure has to be able to operate with a non-metric similarity function, which is not the case for MST, SPM or many other techniques. Out of the three new clustering algorithms described in the previous chapters, only the Percolation Clustering Algorithm fulfills this requirement, since due to its simplicity it involves only ordering the values of the similarity function. In fact any arbitrary similarity function generates a similarity matrix that defines a similarity graph. The procedure is analogous to the MST in the sense that it selects only the highest similarity values to successively connect the items. It is the representation of the results that makes the clustering obvious.

The Percolation Clustering Algorithm proceeds, as described

	CHV	GE	KO	PG	TX	XOM
CHV	0	1.15	1.18	1.15	0.84	0.89
GE		0	0.86	0.89	1.26	1.16
KO			0	0.74	1.27	1.11
PG				0	1.26	1.10
TX					0	0.94
XOM						0

Table 5.4: Distance matrix of the six stocks identified by their ticker symbols for the year 1990.

in Chapter 2, by building an ordered list of the similarity function's values. The difference is that the more similar two stocks are the larger their correlation coefficient, while the more alike two sample points are the smaller the distance between them. Therefore, using correlation coefficients as a measure of similarity, the algorithm starts by ordering these coefficients in a descending rather than an ascending order. The next step is to sweep through this list and, once a value is encountered, group together the two stocks with the respective correlation coefficient. The output is a graphical representation of largest cluster size as a function of correlation coefficients. The plateaus, where the cluster size remains unchanged for while, indicate the completion of a class, while an increase larger than one stock at a time in the monitored cluster size points out the addition of a new subgroup.

To demonstrate the proof of concept we analyze comparatively for the year 1990 a six stock portfolio using the MST and the Percolation Clustering Algorithm. The considered assets are: Chevron (CHV), General Electric (GE), Coca Cola (KO), Proctor & Gamble (PG), Texaco (TX) and Exxon

	CHV	GE	KO	PG	TX	XOM
CHV	1	0.34	0.30	0.34	0.65	0.60
GE		1	0.63	0.60	0.21	0.33
KO			1	0.73	0.19	0.38
PG				1	0.21	0.39
TX					1	0.56
XOM						1

Table 5.5: Correlation matrix of the six stocks identified by their ticker symbols for the year 1990.

Ticker Symbols	ρ	d
KO - PG	0.73	0.74
CHV - TX	0.65	0.84
KO - GE	0.63	0.86
GE - PG	0.60	0.89
CHV - XOM	0.60	0.89
TX - XOM	0.56	0.94
PG - XOM	0.39	1.10
.....

Table 5.6: Ordered correlation coefficients and distances between six stocks designated by their ticker symbols for the year 1990.

(XOM). This portfolio was also used by R. N. Mantegna and H. E. Stanley [47, 50] as an example of hierarchal taxonomy based on an ultrametric space. Between the 6 items there are $\frac{6(6-1)}{2} = 15$ independent values of the binary similarity function (correlation coefficient or distance). Table 5.4 presents the distance matrix for the given portfolio during the year 1990, reproduced from [47], while Table 5.5 contains the correlation matrix among them during the same interval.

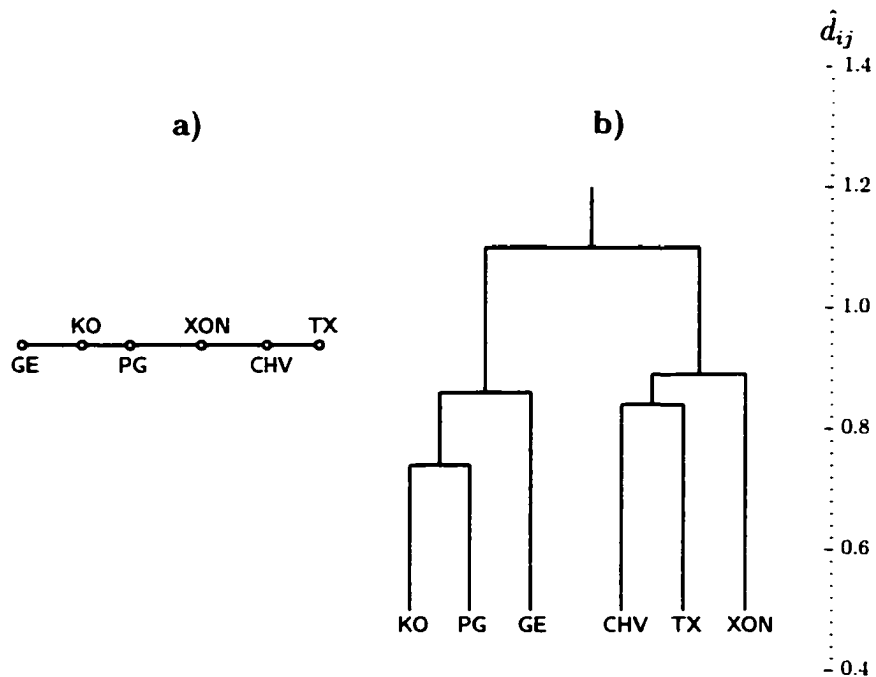


Figure 5.3: (a) MST and (b) Indexed hierarchical tree obtained during the calendar year 1990 for the portfolio of six companies: CHV, GE, KO, PG, TX and XOM (reproduced after [47]).

Note that, as expected, our calculated correlation coefficients and the distances, satisfy the relation (5.11). The highlighted values are the ones used to build the dendrogram and the clusters while the other parameters are somewhat redundant, connecting assets already grouped in the same category. The maximum number of distances (or correlation coefficients) needed to join $N = 6$ assets is $N - 1 = 5$. Based on the above two matrices, Table 5.6 lists, in descending order, the correlation coefficients between the

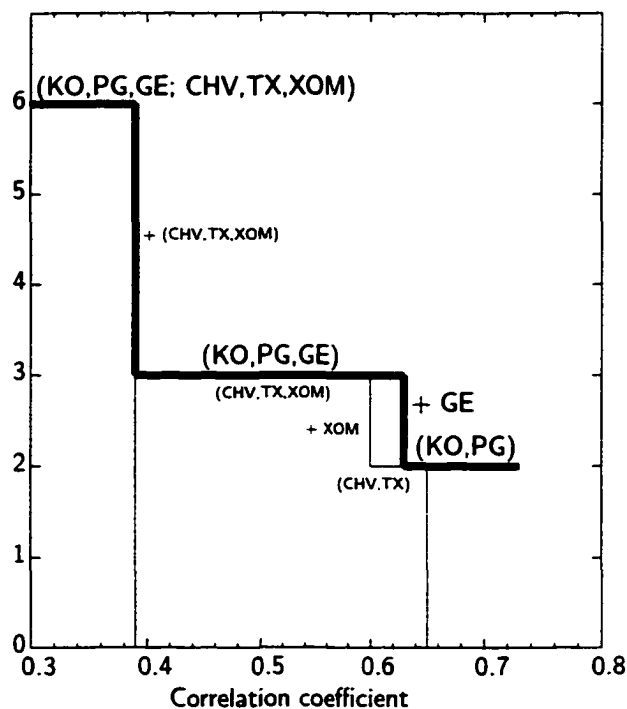


Figure 5.4: Largest and second largest cluster sizes as a function of correlation coefficient during the calendar year 1990 for the portfolio of six companies (CHV, GE, KO, PG, TX and XOM).

six stocks for the year 1990, as well as, in ascending order, the distances among them during the same period. The total number of similarity measures between the items is 15, but only the first seven values are explicitly included in the ordered list, since at this point all assets are connected.

The results of the Percolation Clustering Algorithm are represented in Figure 5.4. Monitoring the largest cluster size function of correlation coeffi-

cients, listed in Table 5.6, one notices a plateau of two stocks that identify the consumer nondurable goods group: KO and PG. Before the correlation coefficient decreases to the value 0.6, this cluster is enlarged by the addition of the durable goods producer GE. When the correlation coefficient reaches a much lower value of about 0.4, the second largest cluster, representing the energy class and formed by CHV, TX and XOM, is added. Using the ordered list of distances in Table 5.6, the MST method generates the hierarchical indexed tree (dendrogram) presented in Figure 5.4.

5.4 Taxonomy of DJIA Portfolio

For the examined intervals 1986-1990 and 1997-2001, the quarterly and annual correlation coefficients between the studied assets are mostly positive. The negative values are generally small and within the noise level. The application of the Percolation Clustering Algorithm to this data set illustrates how the graphical presentation of the results facilitates the identification of economically relevant classes of stocks.

For each of the ten studied years we present below the graphical representation of the first three largest cluster sizes as a function of the correlation coefficient. The largest cluster is represented with a thickest line and data points, the second largest one is drawn in a thin line and the third in a thicker one. For consistency and ease of comparison, the graphical representations have the same interval on the abscises, $[-0.1, 0.9]$ for the years 1986 through 1990 and $[-0.2, 0.8]$ for the period 1997-2001, respectively. Before analyzing

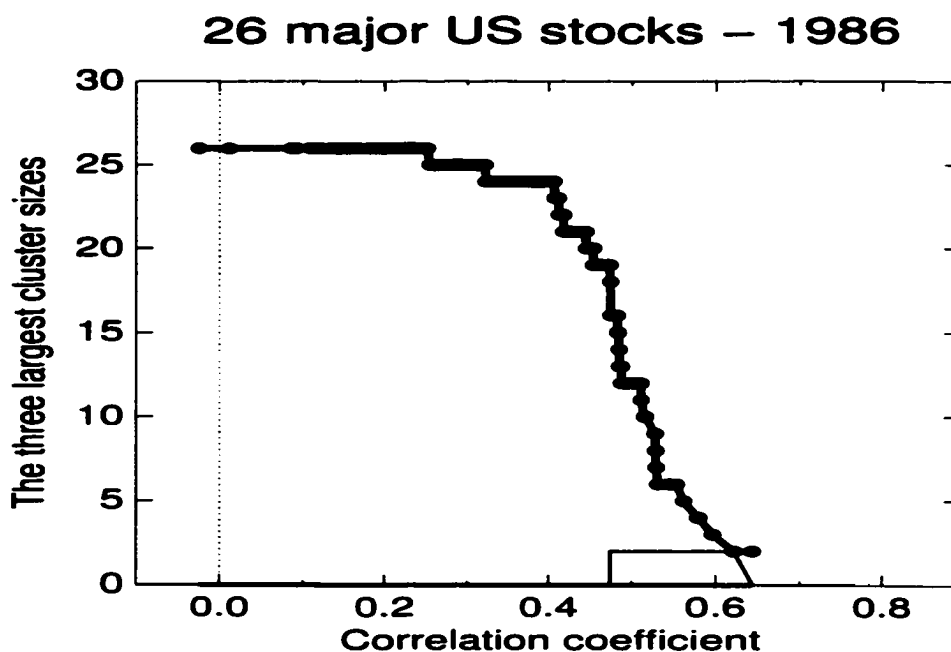


Figure 5.5: The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1986.

the first interval, 1986-1990, a reminder is needed. The portfolio studied during this period is built of major US companies tracked by the DJIA in 1991. Therefore, during the interval 1986-1990, the performance of the portfolio items can be compared with DJIA variations, presented in Figure 5.1 only *qualitatively*.

The year 1986 is a good performing interval for the market and the correlation between stocks is rather strong, as seen in Figure 5.5. The separation between primary groups is not clear. Nevertheless, notice the second largest

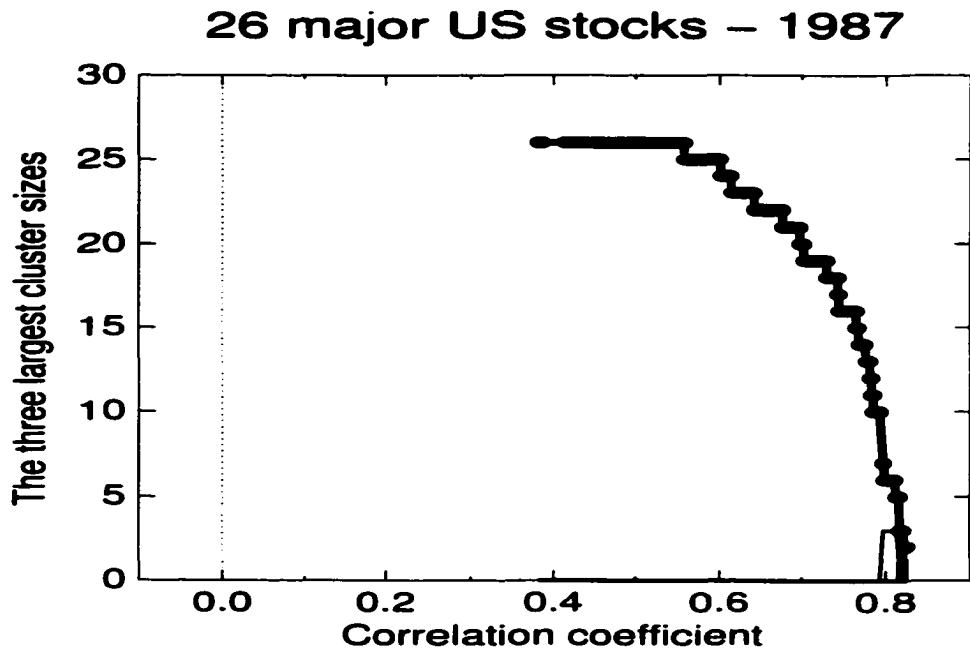


Figure 5.6: The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1987.

cluster composed of two major oil companies, XOM and CHV, as a stable group. As usually happens during major market moves, most of the large cap stocks behave similarly. The largest cluster of six stocks is a mixture of heavy industries (MMM and DD), combined with consumer goods manufacturers (PG, KO, GE) and even retailers (S). This cluster continues to grow by the addition of several diverse companies (MCD, GM, AXP, MO, MRK, and T).

The strongest correlation in the interval from 1986-1990 is seen in 1987.

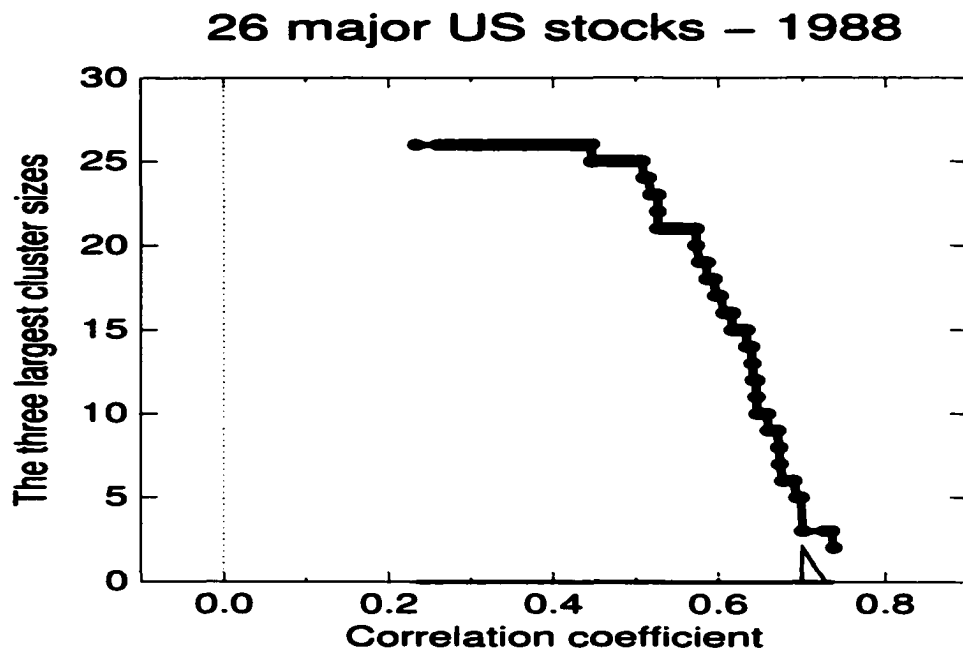


Figure 5.7: The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1988.

as illustrated in Figure 5.6. This is due to the continuation of the bull market started in the previous year, followed by the abrupt drop registered during the fourth quarter of 1987. During major and abrupt market shifts, the structure of the portfolio is not clearly defined. All items perform as a unitary group due to the psychological component of the investment process. However, notice the existence of two slightly separated clusters, the second largest with three assets (S, GE, DIS) and the third largest composed of EK and IBM, which nowadays would correspond to the technology group.

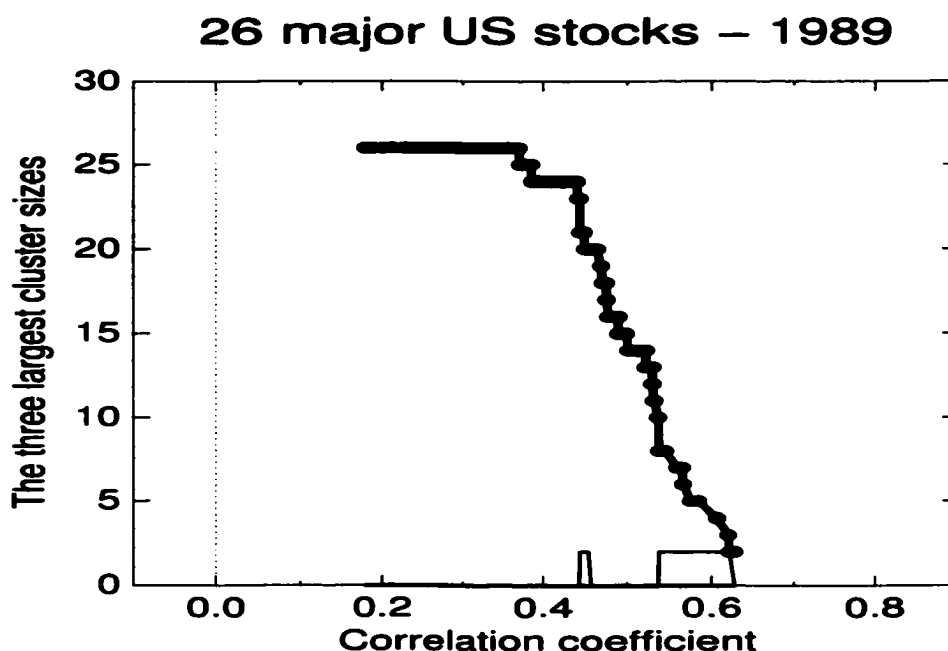


Figure 5.8: The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1989.

In 1988 we again see a high correlation between stocks, as shown in Figure 5.7, though not at the level seen in 1987. This still causes a poor resolution between different sectors. The two slightly distinguishable clusters, the larger composed of three companies (XOM, KO, and GE) and the second one formed by MMM and IBM, have no meaningful economic relevance.

The year 1989, shown in Figure 5.8, is previous to the weak year in 1990, and thus reveals a smaller average correlation between assets. The structure is more discernable and we notice the existence of two groups. S and AXP

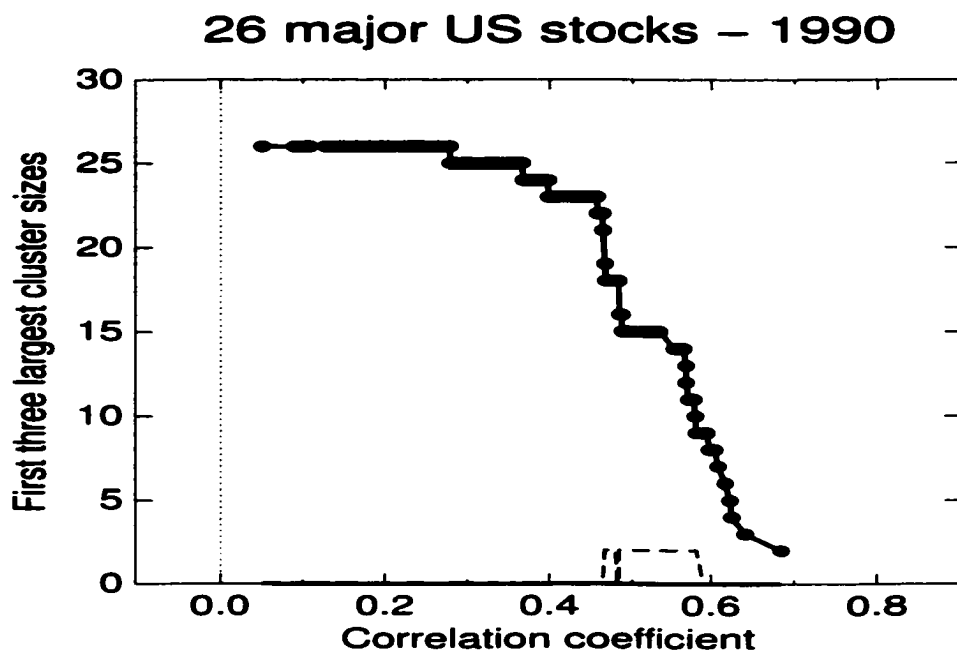


Figure 5.9: The three largest cluster sizes as a function of correlation coefficient for 26 major US companies during the year 1990.

form a fairly stable cluster related to retailers and financial groups. The second set of two items has a rather small intra-cluster correlation, less than 0.5, and contains the companies Z and BS.

The only year that the market ended lower than it started in the first interval was 1990. Notice a more visible structure of the portfolio components. The plateaux encountered by the largest cluster size, shown in Figure 5.9, are larger as compared to the previous years. Again, CHV and XOM form a stable separate cluster. The stocks group into a large cluster of 15 elements.

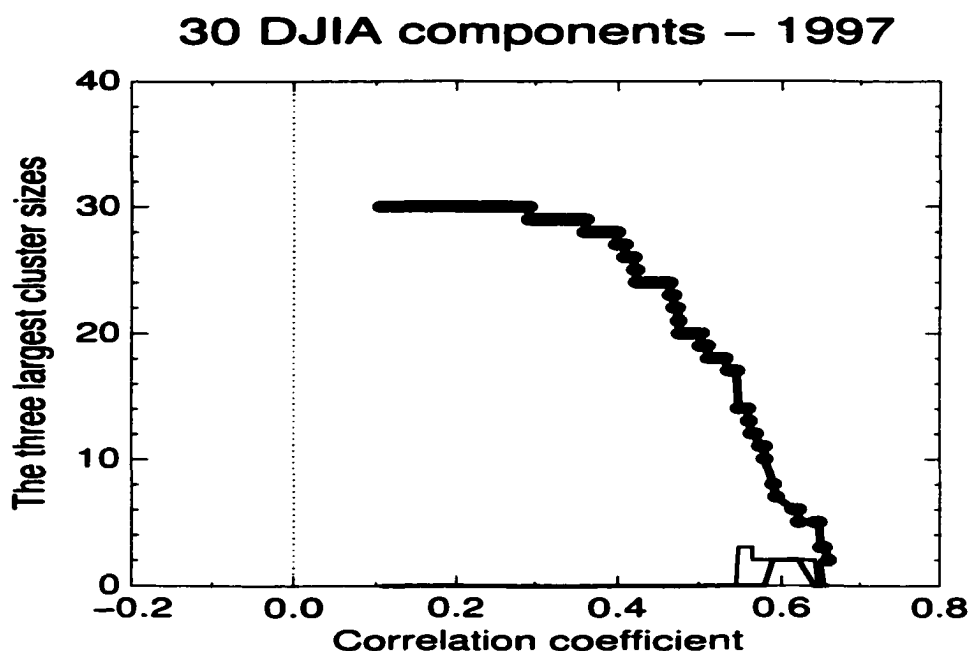


Figure 5.10: The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 1997.

which then increases in two steps to 18 and then to 23. When the markets are weak, we conclude that the stocks have a stronger intra-cluster correlation and a lower inter-cluster one.

During the year 1997, most of the stocks form a rather compact group. As we can see in the Figure 5.10, there is a small plateau in the largest cluster size that contains a mixed group of consumer goods producers and banks: PG, KO, GE, JPM and C, but it is quickly absorbed in a common larger cluster. More noticeable is the second cluster that gathers initially two and

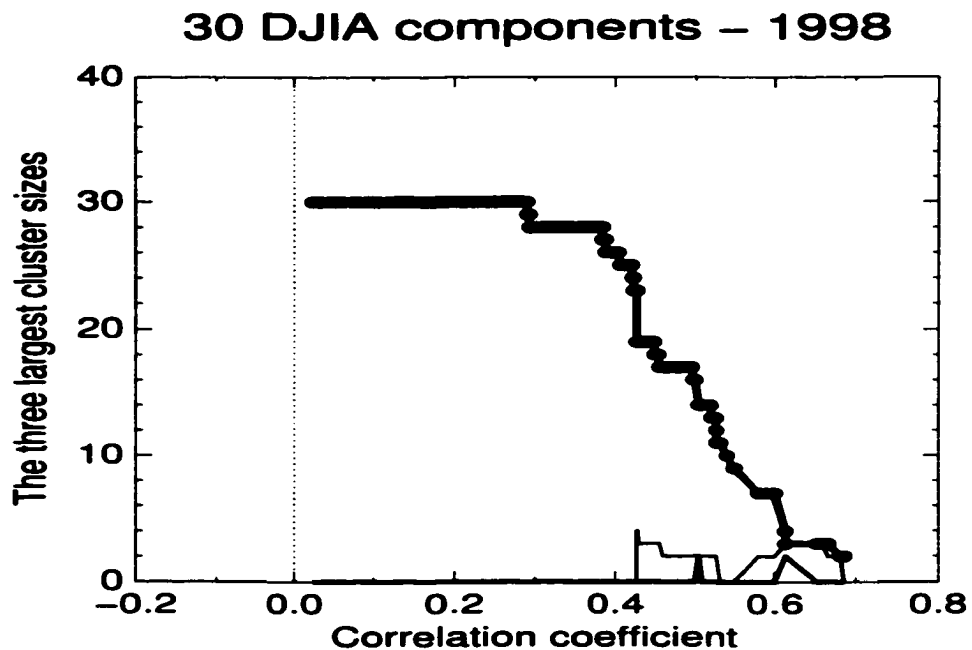


Figure 5.11: The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 1998.

then three assets: MSFT, INTC and IBM, identifying the technology group. The third cluster containing MRK and JNJ represents the pharmaceuticals.

A more complex structure is revealed for 1998 and is presented in Figure 5.11. The largest cluster is initiated by three strongly correlated items: WMT, HD and GE that are providers and distributors of consumer goods. The financial assets, JPM, C and AXP, form the second largest cluster. The third group, MSFT and INTC, indicate the existence of the technology class. All three clusters merge together and connect with other DJIA components.

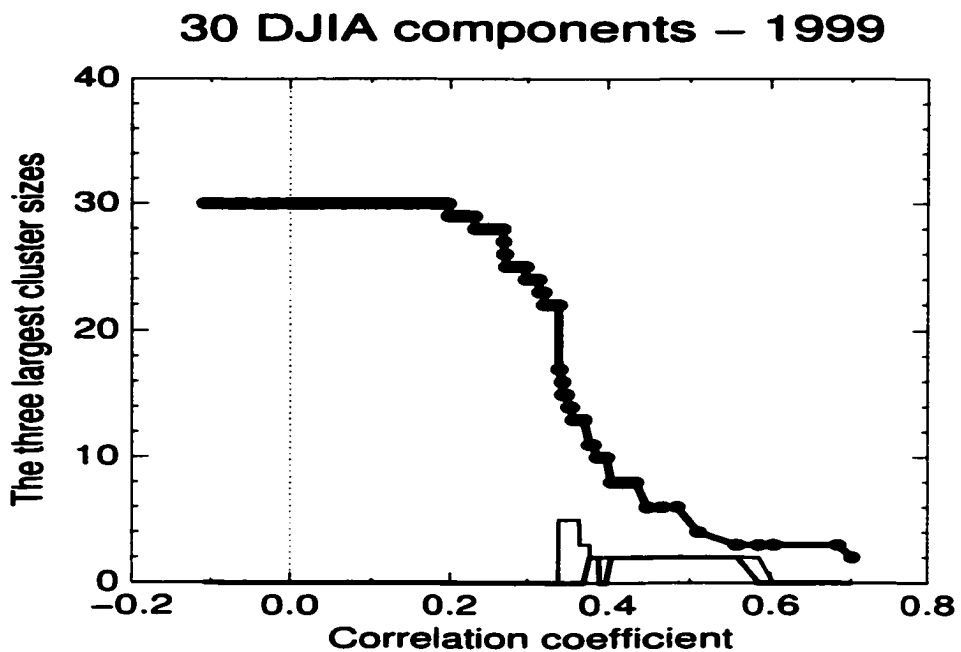


Figure 5.12: The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 1999.

Continuing to monitor the second and third largest clusters, one notices two emerging groups. The largest one related to commodities and heavy industries contains MMM, IP and DD. The smaller group represents diversified technologies and contains HON and UTX.

The year 1999 is represented in Figure 5.12. During this period the average correlation over all DJIA components decreases, but the correlation within clusters remains strong. Therefore the plateaus are longer and the clusters more stable. The groups not only indicate a category, but contain

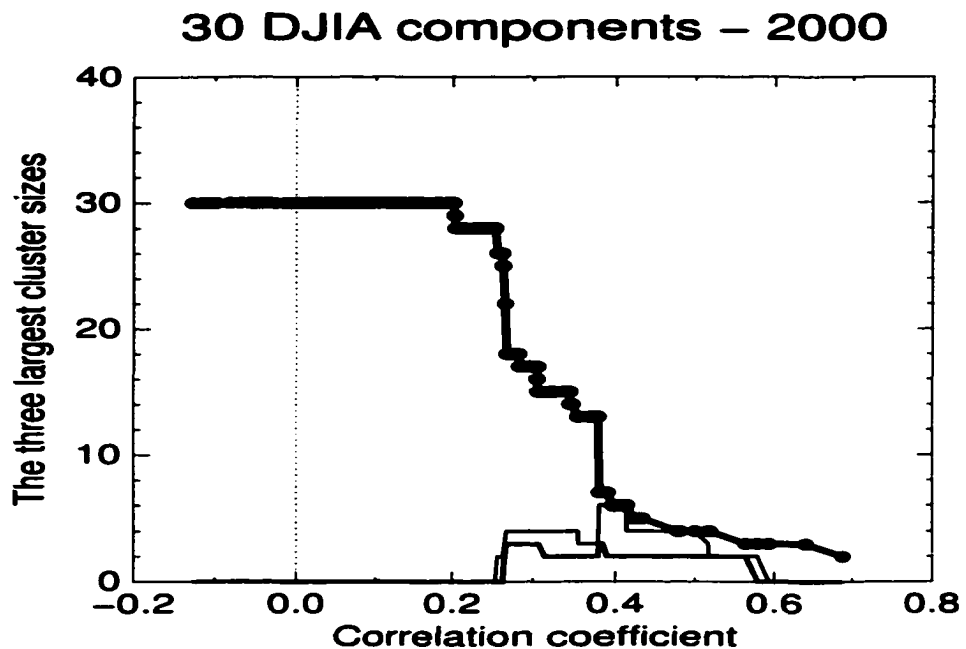


Figure 5.13: The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 2000.

all the DJIA components related to that industry. The largest cluster starts with a plateau formed by financial stocks AXP, C and JPM. The second cluster contains initially MSFT and INTC, merging later with the other two technology related companies IBM and HPQ. The third cluster contains MRK and JNJ. Once all these groups are joined together, there still exists a separate cluster of four items all involved in commodities related heavy industries: AA, IP, DD and MMM. Notice also the existence of several small negative correlation values.

In the year 2000 the DJIA ends lower than it started. Out of the interval 1997-2001, the year 2000 is the one with the lowest average correlation between the DJIA components. The different classes of stocks are easily discernible in Figure 5.13. The largest cluster starts with a group of three financial companies: AXP, C and JPM, and is soon joined by GE. In the second part of the 1990's GE extended its business from industrial durable goods to financial products. As a consequence, if in the 1980's it belonged to the consumer products group, in the 1990's it is closer to financial companies. The second cluster represents the heavy industries containing at the beginning four assets: CAT, DD, IP and MMM. It continues to increase quickly by adding AA and UTX. Once this group merges with the largest cluster, another class becomes visible: the technology group with HPQ, IBM, INTC and MSFT. The third cluster of two items contains the pharmaceuticals JNJ and MRK.

During the interval 1997 through 2000, the clusters of stocks with intra-cluster correlation coefficients larger than 0.6 each contain two or three assets. As we can see in Figure 5.14, the year 2001 reveals the existence of several clusters of five or even six assets, which have an intra-cluster correlation coefficient larger than 0.6. The largest group initially includes C and JPM, and is then connected rapidly with APX, GE and HON. The second largest cluster starts with AA, DD, IP, MMM and, soon after, CAT. The third group holds the four technology items: HQP, IBM, INTC and MSFT. The two clusters that are noticeable around a correlation value of 0.3 represent the pharmaceuticals (JNJ and MRK) and the nondurables consumer goods

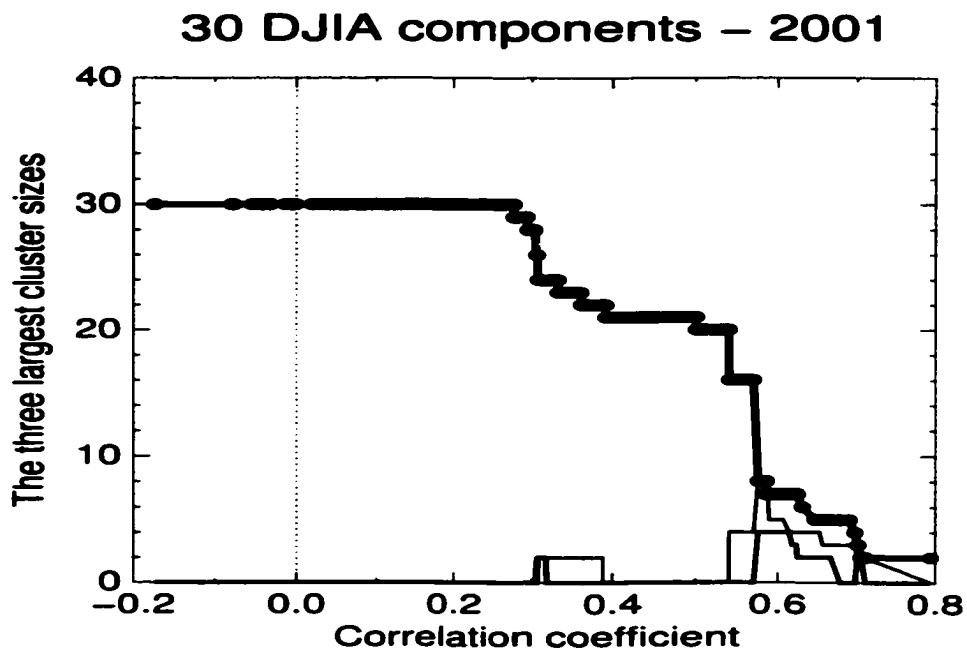


Figure 5.14: The three largest cluster sizes as a function of correlation coefficient for the portfolio of the 30 DJIA components during the year 2001.

group (KO and PG).

The examples presented above demonstrate how the graphical representation of the largest cluster size as a function of correlation coefficient helps to identify clusters of stocks with economic relevance.

By using the same limits on the axes of all the graphs, we can easily compare the differences in the behavior of the DJIA components from one year to the next. An interesting observation is how in each year, starting from 1997 to 2000, the average correlation coefficient over the DJIA ensemble de-

creases, thus the graphs glide leftward. Simultaneously, the plateaus become longer and the jumps in the main cluster size larger. Thus, the market as an ensemble gradually loses its “strength”, it becomes more fragmented and the different stock categories become more noticeable, *i. e.* the intra-cluster correlations become stronger, while the inter-cluster ones decline. Once the transition from a strong to a weak market ends and the prices continue to drop, the average correlation increases again, as happens in 2001. but becomes almost irrelevant. An underlying structure is more and more visible and the DJIA contains several separate clusters. A clear definition of stock groups can be seen during the years when the market moves sideways or underperforms, as in 1990, 2000 or 2001. In order to better describe this behavior, the following section analyzes the statistical properties of the correlation coefficients.

5.5 Statistical Properties of Correlation Coefficients

It has been shown that the indexed hierarchical trees associated with the DJIA or the S&P 500 index, obtained by means of the MST, vary slowly in time, maintaining a basic structure on a time scale of several years [47]. The same result was obtained partitioning the DJIA components with the Percolation Clustering Algorithm. For the period 1997-2001, four separate classes of stocks were detected: financial companies (AXP, C, JPM), technology

(HPQ, IBM, INTC, MSFT), pharmaceuticals (MRK, JNJ), and heavy industry (AA, CAT, DD, HON, MMM, UTX). Though distinguishable each year, these clusters are “closer” during the “bullish” periods, when the intra-cluster correlation becomes of a similar magnitude to the inter-cluster correlation. During these years, the groups connect before they are completely formed and the DJIA portfolio behaves more as one unit. There are other small groups, such as the retailers (WMT, HD) or nondurable consumer goods producers (KO, PG), that can be identified only during some years. Describing this behavior by analyzing the statistical properties of the correlation coefficients is the main focus of the current section. Between the 26 major companies considered for the period 1986-1990, there are 325 correlation coefficients calculated quarterly and annually, and 435 correlation coefficients calculated between the 30 DJIA components for the interval 1997-2001. Their histograms and the moments of the ensemble are discussed below.

We start by analyzing twenty six U.S. major companies, considered representative for the interval 1986-1990. The histograms of the 325 correlation coefficients are presented in the Figures 5.15 and 5.16. Histogram representations are extremely susceptible to the size of the bins. For consistency, the limits of all histograms are chosen to be between -0.3 and 0.9, in order to accommodate all of the annual correlation coefficient values encountered during this interval. Furthermore all histograms are divided into 150 bins.

An interesting observation regarding Figures 5.15 is that the positive market years 1986, 1988 and 1989 are characterized roughly by a Gaussian shape of the histograms. In contrast, the year 1987 displays a histogram

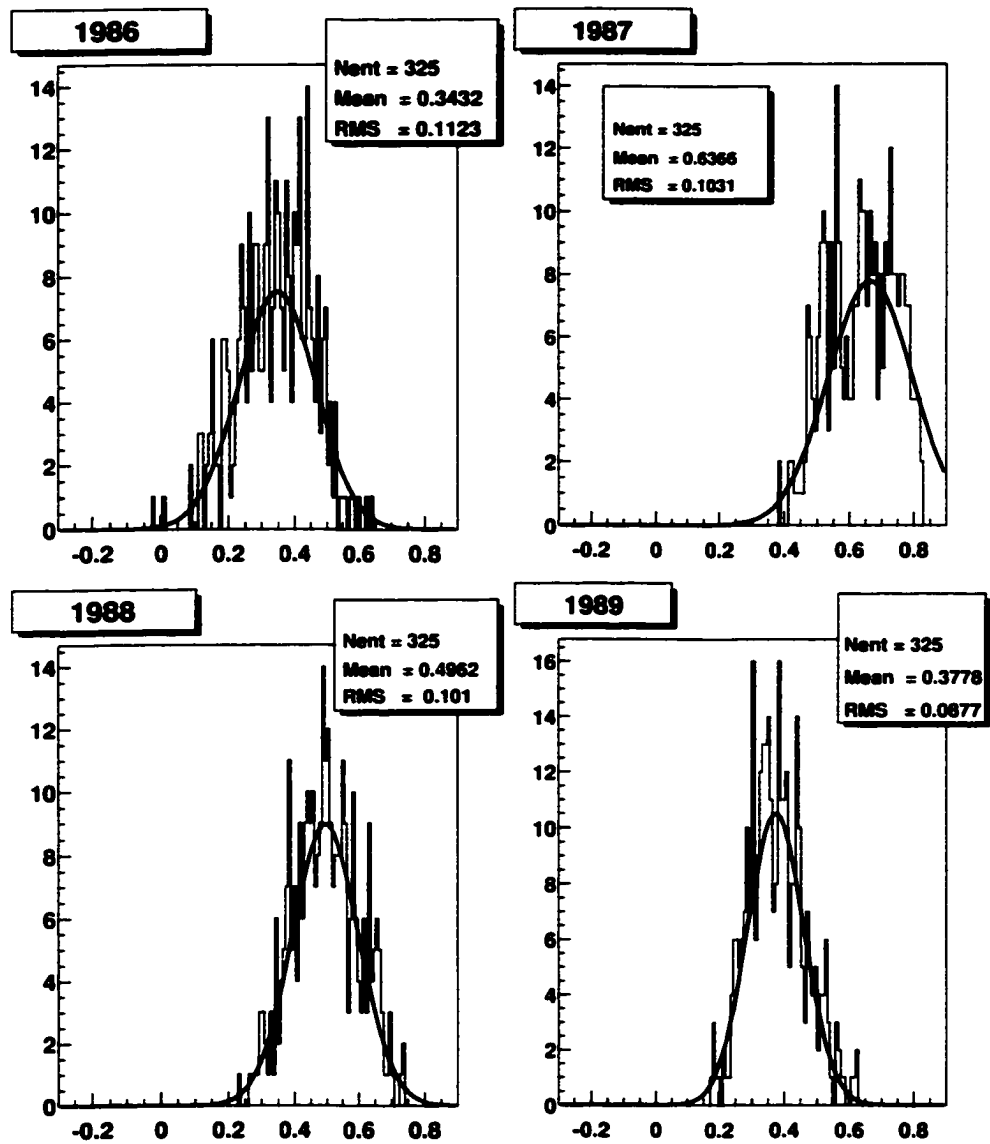


Figure 5.15: Histograms of yearly correlation coefficients between 26 major US companies considered representative for the interval 1986-1989.

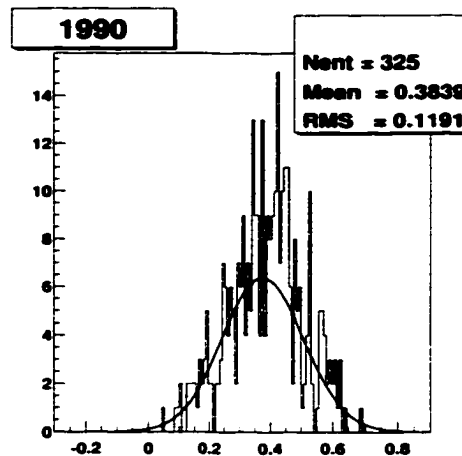


Figure 5.16: Histograms of yearly correlation coefficients between the 26 major US companies considered representative for the year 1990.

quite different from the Gaussian shape.

The same remark regarding the limits of the histograms can be made concerning Figures 5.16 and 5.18, which present the histograms of the 435 correlation coefficients between the 30 DJIA components calculated yearly for the period 1997 - 2001, the only difference being that the limits of the histograms are chosen to be between -0.4 and 0.8, in order to suit the annual correlation coefficient values during this period. The number of bins remains at 150. Notice that the distribution is almost Gaussian during the bullish years 1997 through 1999 with a decreasing ensemble average. As we make the transition to a bear market in 2000, the histograms become more complex and in 2001 the distribution is not unimodal anymore. It is obvious that the thirty stocks no longer behave as a unit with an average correlation, but

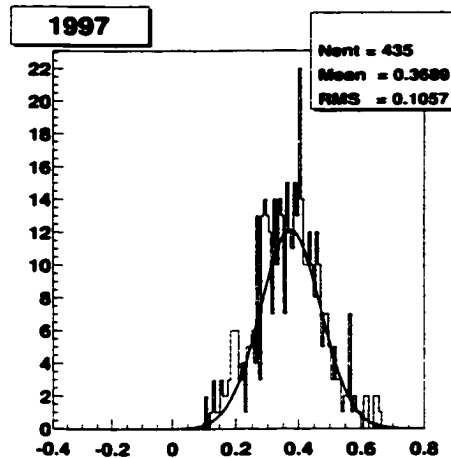


Figure 5.17: Histograms of yearly correlation coefficients between the 30 DJIA components for the year 1997.

rather form several separate classes.

Detailed histograms of the correlation coefficients calculated quarterly for the ten studied years 1986-1990 and 1997-2001 are given in Appendix B. For each period, the limits of the histograms and the number of bins remain as specified above.

Since the time distribution of the price changes $Z(t)$ or the variation in the natural logarithm of price $S(t)$ are not completely known, their correlation might generate unexpected results. In order to determine if the statistical properties of the correlation coefficients carry information about stock interactions and are not the result of changes in the underlying time series, we calculate the correlation coefficients between the shuffled time series. We select the thirty time series of the daily logarithmic price variation, $S(t)$.

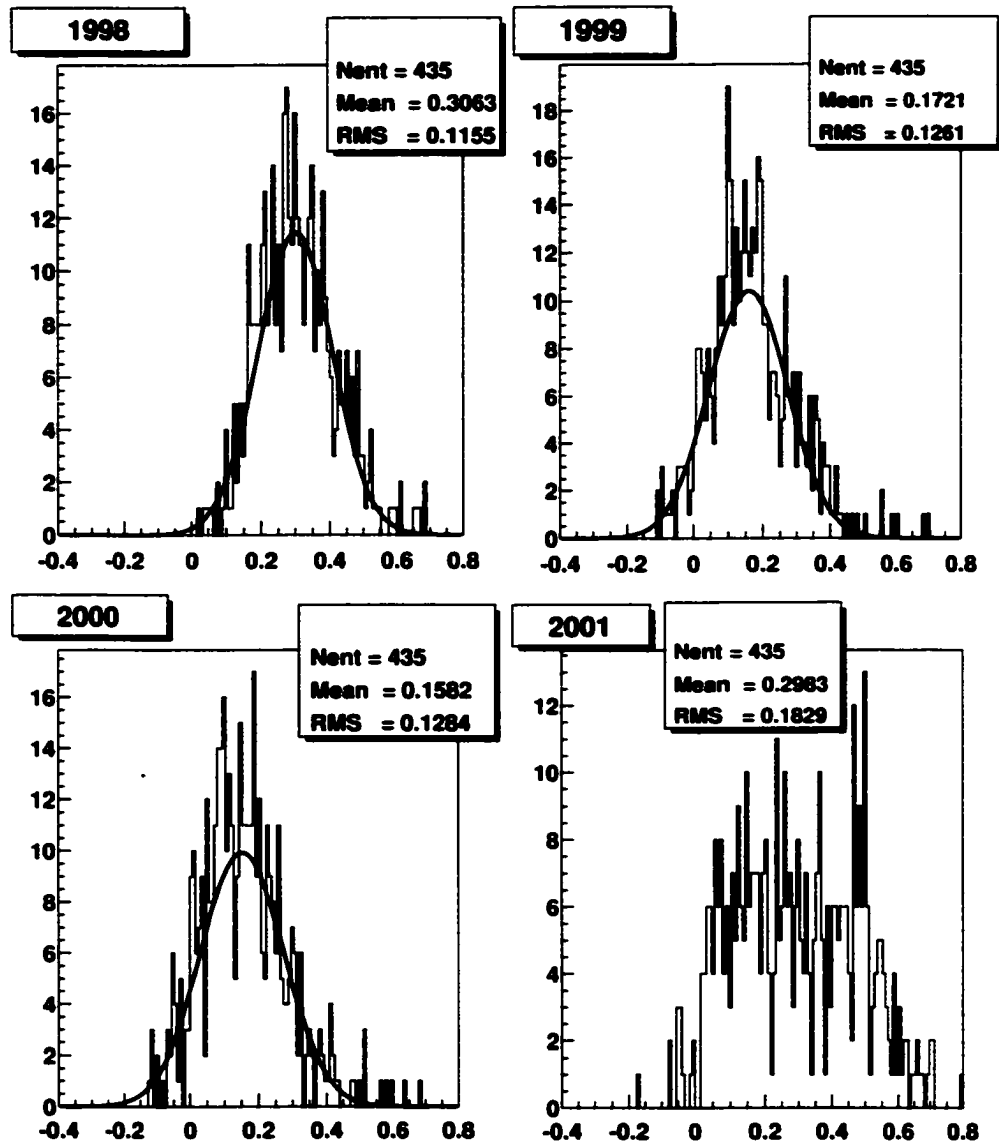


Figure 5.18: Histograms of yearly correlation coefficients between the 30 DJIA components for the period 1998-2001.

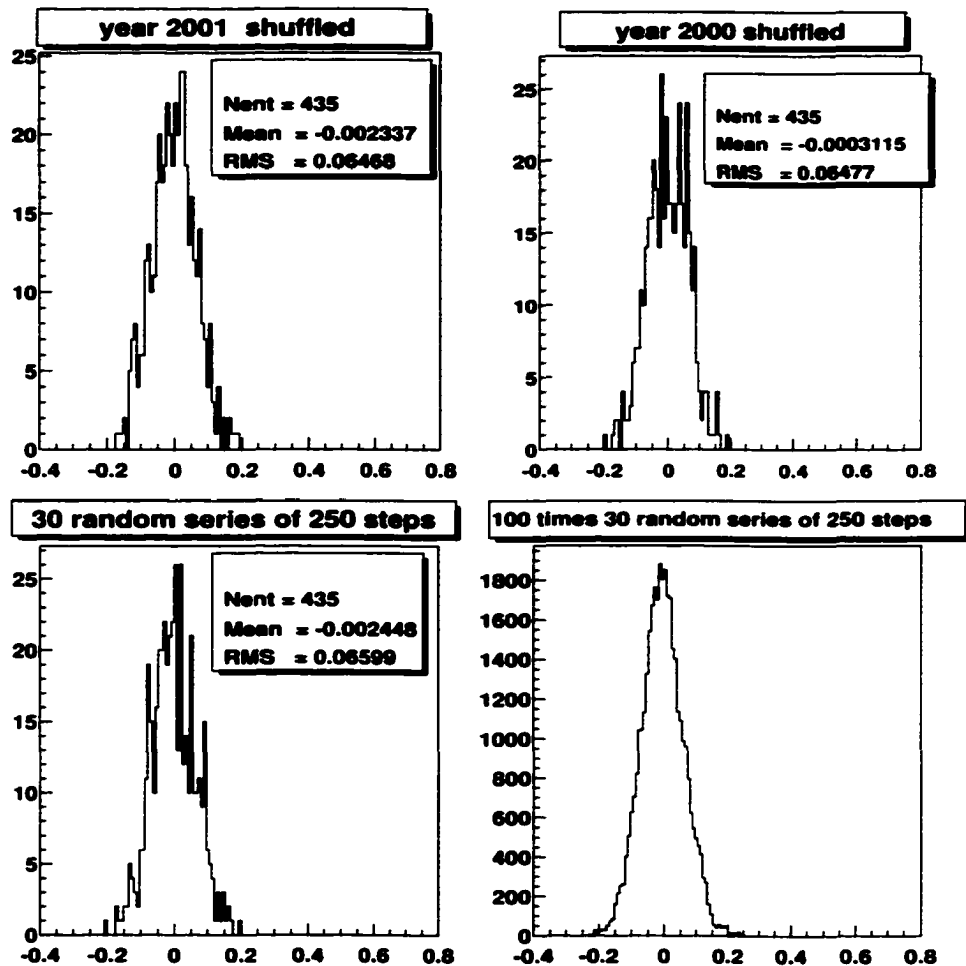


Figure 5.19: *Top*: Histograms of correlation coefficients between shuffled time series of daily logarithmic price variation for DJIA stocks during 2000 and respectively 2001. *Bottom*: Histograms of correlation coefficients between 30 series of 250 normal distributed random numbers and 100×30 series of 250 normal distributed random numbers.

for DJIA stocks, during the interval 2000-2001, since the index structure is most noticeable throughout these two years. We shuffle the records, wiping out the information about the stocks' synchronous behavior, but their distribution remains unchanged. The histograms of the 435 correlation coefficients between the 30 shuffled time series are presented on the top of Figure 5.19. The bottom of this figure contains the histograms of correlation coefficients between series of normally distributed random numbers with the same length. The similarity between the four graphs is obvious. Hence once the synchronicity of logarithmic price variation is removed the correlation coefficients become white noise. Therefore, correlation coefficient histograms reflect interaction between stocks and changing market conditions.

The mean value of the correlation coefficients over an ensemble of N stocks is calculated as:

$$\bar{\rho} = \frac{1}{M} \sum_{i,j=1,(j>i)}^N \rho_{ij}, \quad (5.15)$$

where ρ_{ij} is defined by equation (5.5) and M is the number of independent coefficients $M = \frac{N(N-1)}{2}$. Figure 5.21 presents the average quarterly correlation coefficient between the DJIA components during the interval 1997-2001. As mentioned above, from 1997 throughout 2000 this average decreases, indicating a weakening market unity. The smallest value is recorded in the third quarter of 2000 and, once a weak market is established, the average correlation increases, however, it is no longer representative for the whole ensemble.

The same fragility of the market is revealed by monitoring the variability

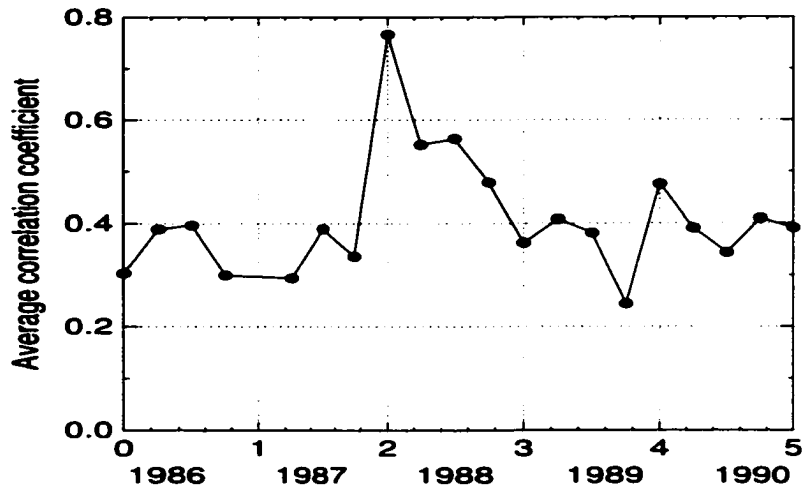


Figure 5.20: Average value of quarterly correlation coefficients between 26 major US companies for the interval 1986-1990. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.

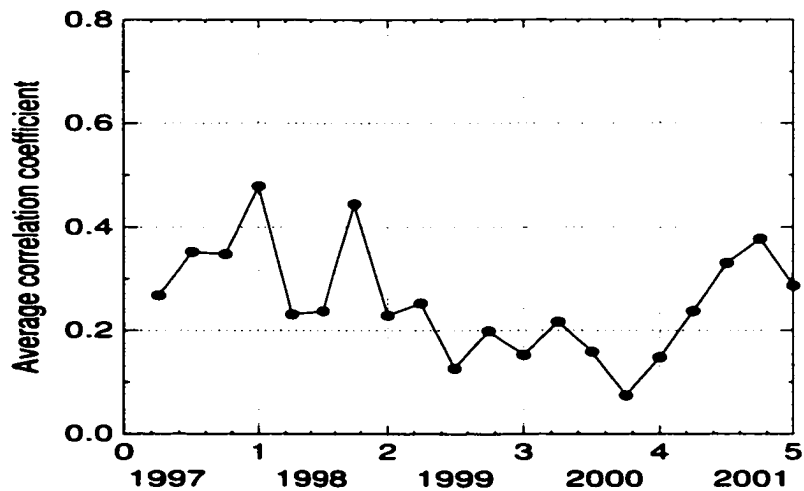


Figure 5.21: Average value of quarterly correlation coefficients between DJIA components for the interval 1997-2001. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.

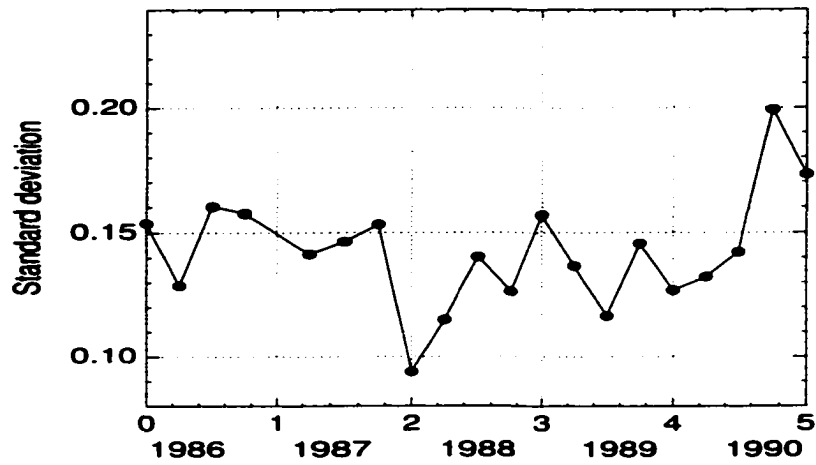


Figure 5.22: Standard deviation of quarterly correlation coefficients between 26 major US companies for the interval 1986-1990. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.

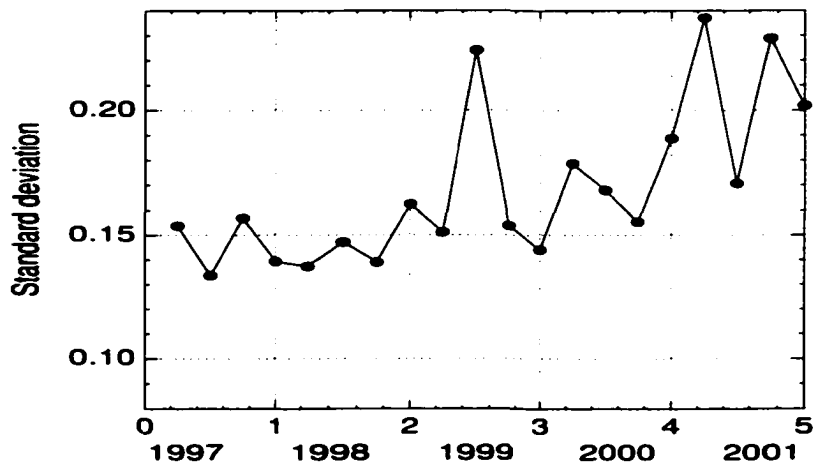


Figure 5.23: Standard deviation of quarterly correlation coefficients between DJIA components for the interval 1997-2001. On the abscissa, each unit represents one year and the data points are placed at the end of each quarter.

of the correlation coefficients. The fluctuations from the average are described by the ensemble standard deviation (also known as root mean square - RMS) described as:

$$\sigma = \sqrt{\overline{(\rho_{ij} - \bar{\rho})^2}}. \quad (5.16)$$

This quantity is calculated for the quarterly correlation coefficients during the interval 1997-2001 and is represented graphically in Figure 5.23. While the RMS is low for the first two years of the interval, starting with 1999 and simultaneously with a decreasing average value, the variability increases, displaying two peaks in the second quarter of 1999 and the first quarter of 2001.

The inspection of the ensemble average or the deviation from the average is not sufficient to characterize asymmetric or multimodal distributions. Therefore it make sense to monitor the third and fourth normalized cumulants [53] of the correlation coefficients ensemble. The first of these quantities, called the *skewness* is defined as:

$$\lambda_3 = \frac{\overline{(\rho_{ij} - \bar{\rho})^3}}{\sigma^3} \quad (5.17)$$

where σ is the standard deviation defined by the equality (5.16). In the case of a unimodal distribution, the skewness expresses the asymmetry, specifically by how much the values larger than the average differ from the values smaller than the average, normalized by the standard deviation. A negative skewness is obtained for the histograms (or distributions) where the values smaller than the average are more frequent, while a positive skewness reflects the reverse situation.

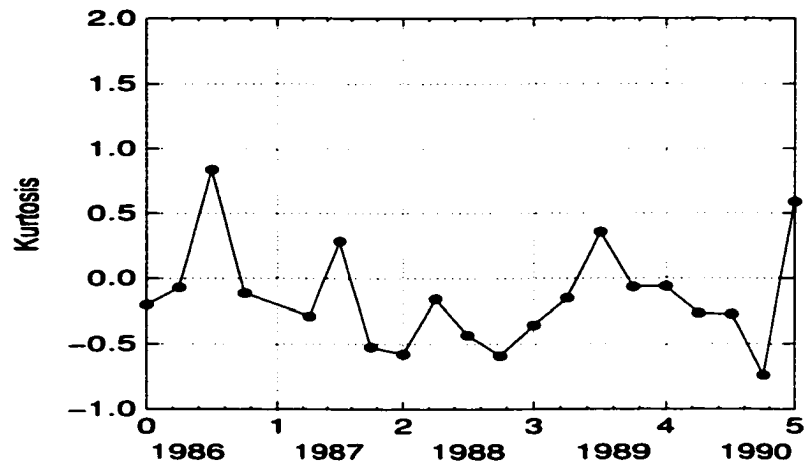


Figure 5.24: Kurtosis of quarterly correlation coefficients between 26 major US companies for the interval 1986-1990. On the abscissa each unit represents one year and the data points are placed at the end of each quarter.

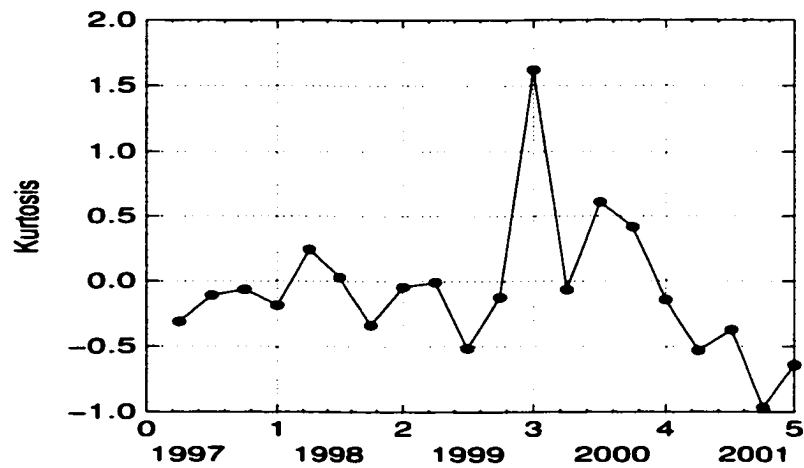


Figure 5.25: Kurtosis of quarterly correlation coefficients between DJIA components for the interval 1997-2001. On the abscissa each unit represents one year and the data points are placed at the end of each quarter.

Another relevant quantity is the fourth normalized cumulant, the *kurtosis*, described as:

$$\kappa = \lambda_4 = \frac{\overline{(\rho_{ij} - \bar{\rho})^4}}{\sigma^4} - 3. \quad (5.18)$$

The meaning of the kurtosis for unimodal distributions can be understood from the following example. Since the kurtosis of a Gaussian distribution is 3, any distribution with a positive kurtosis would have a longer tail. In other words, the extreme values of the random variable obeying the positive kurtosis distribution are more frequent than in a Gaussian distribution. Obviously, a negative kurtosis is representative for a unimodal distribution narrower than a Gaussian. In general the kurtosis can be interpreted as the measure of the dissimilarity between a given distribution and a Gaussian distribution with the same mean and variance.

The kurtosis for the ensemble of the quarterly correlation coefficients between the DJIA components for the period 1997-2001 is presented in Figure 5.25. We notice the small values of the kurtosis during 1997 throughout the third quarter of 1999 which are also reflected by the Gaussian shape of the correlation coefficients' histograms during this interval. In the last quarter of 1999, it seems that the market loses the strength that kept its unity and the ensemble of correlation coefficients encounters the highest value of the kurtosis. The highest value implies large variations of the correlation coefficient values. Thereafter, this parameter continues to oscillate indicating significant deviations of the histograms from a Gaussian distribution.

5.6 Properties of Correlation Matrix

To obtain relevant information from the analysis of the return cross-correlation matrix of a N stocks portfolio we have to eliminate the noise inherent in the time series [53, 54, 55]. The symmetrical $N \times N$ correlation matrix has $\frac{N(N-1)}{2}$ independent coefficients which must be determined from N time series of length T . The larger the number of assets N , the larger the number of independent entries and, to reduce the noise in the correlation matrix, the longer time series, T , are needed to obtain significant information. In order to distinguish the noise from the “signal” we have to compare the empirical correlation matrix with the null hypothesis, *i. e.* a random matrix obtained from a given number of strictly independent finite time series.

In our calculations we consider a $N \times T$ rectangular matrix \mathbf{M} , composed of daily logarithmic closing price variations for N assets over a time series of length T . The number of considered assets is $N = 26$ for the interval 1986-1990, and $N = 30$ for the period 1997-2001. Table 5.3 presents the length of the analyzed time series. The number of examined trading days in each quarter is between 59 and 64 and between 246 and 253 in each year. The total number of days in the first five-year interval is 1260 and in the second 1250. Thus the number of elements, T , in the logarithmic closing price variation time series is between 58 and 63 for the quarter, between 245 and 252 for the year, and between 1259 and 1249 the five-years interval. The specific dimensions of the empirical matrix \mathbf{M} are important in choosing the null hypothesis model. To calculate the correlation matrix \mathbf{C} , the matrix \mathbf{M}

is transformed into matrix $\widetilde{\mathbf{M}}$ by normalizing each time series to zero mean and unit variance and:

$$\mathbf{C} = \frac{1}{T} \widetilde{\mathbf{M}} \widetilde{\mathbf{M}}^\dagger, \quad (5.19)$$

where $\widetilde{\mathbf{M}}^\dagger$ is the transpose of matrix $\widetilde{\mathbf{M}}$. The correlation coefficients calculated this way correspond to the ones defined in equation (5.5). For each of the two periods 1986-1990 and 1997-2001, we calculate the correlation coefficients quarterly, annually and for the five-year interval and build the $N \times N$ correlation matrices. All these matrices are symmetric and therefore have real eigenvalues which are presented in the histograms in Figures 5.27 through 5.29 and in Appendix C. Before analyzing the eigenvalue spectra, we must eliminate the noise effect emulated by the null hypothesis.

The null hypothesis model, in our case, has $N = 30$ uncorrelated series of identically distributed random numbers. The length, T , of the series is chosen to be 60 records for quarterly analysis, 250 for the annual case, and 1250 for the five-year interval. Thus, we build random matrices \mathcal{M} of dimensions $N \times T$ whose elements, ε_t^i (where $t = 1, \dots, T$ and $i = 1, \dots, N$), are zero mean Gaussian distributed random variables. The variance of each time series is chosen to be $\frac{1}{\sqrt{T}}$, with T specified above for different intervals. Now define a matrix \mathcal{C} as:

$$\mathcal{C} = \mathcal{M} \mathcal{M}^\dagger. \quad (5.20)$$

The total variance of this new matrix is equal to unity. Even if the original matrix \mathcal{M} is randomly generated, the properties of the matrix \mathcal{C} are known

for the case when $T, N \rightarrow \infty$, with a fixed ratio

$$Q = \frac{T}{N} \geq 1. \quad (5.21)$$

The theory of random matrices shows that the eigenvalues, λ , of the matrix defined by equation 5.20 have the following distribution:

$$p(\lambda) = \frac{Q}{2\pi} \frac{\sqrt{(\lambda_{max} - \lambda)(\lambda - \lambda_{min})}}{\lambda}, \quad (5.22)$$

and the eigenvalue spectrum has an upper and a lower limit equal to:

$$\lambda_{max,min} = 1 + \frac{1}{Q} \pm 2\sqrt{\frac{1}{Q}}. \quad (5.23)$$

Our null hypothesis model considers $Q = \frac{60}{30} = 2$ for the quarterly calculations. $Q = \frac{250}{30} = 8.33$ for the annual ones, and $Q = \frac{1250}{30} = 41.7$ for the five-year interval. The theoretical eigenvalue spectra of the infinite random matrices for the cases in discussion are presented in Figure 5.26. Notice that all of these spectra have strictly positive lower limits and the density of the eigenvalues exhibit well defined peaks. Even more important for our purposes is the fact that the density of the eigenvalues becomes zero for values larger than the upper limits. These results are valid for matrices \mathcal{M} with a large number of elements, therefore the boundaries can be considered only roughly for matrices with a smaller number of components.

To detect the differences between the spectra of infinite random matrices and the spectra of finite ones, we used Mathematica to generate groups of 10,000 \mathcal{M} random matrices of required parameter Q . For quarterly correlation matrices the null hypothesis considers $N = 30$ uncorrelated series

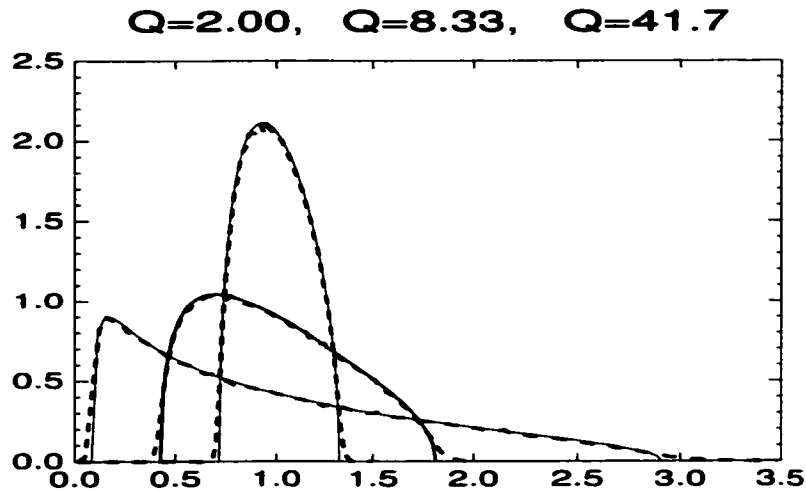


Figure 5.26: Eigenvalue spectra (solid lines) of infinite random matrices for the cases: $Q = 2$ (widest), $Q = 8.33$ (middle) and $Q = 41.7$ (narrowest and highest) and normalized eigenvalue histograms (dotted lines) for 10,000 random 30×30 matrices of the same parameters Q .

of length $T = 60$. Each series has a Gaussian distribution with zero mean and $\sigma = 1/\sqrt{60} = 0.12909$ standard deviation. The histogram of the 300,000 eigenvalues of the 10,000 symmetrical matrices \mathcal{C} , defined by equation (5.20), for the case $Q = 2$ is normalized to unity so that it can be compared to the theoretical distribution. This normalized histogram is drawn as a dotted line that closely follows the theoretical graph of $Q = 2$ in Figure 5.26. For the annual correlation matrices, we create again 10,000 \mathcal{M} matrices, but with $N = 30$ and $T = 250$. The elements of the series are still random variables with a zero mean Gaussian distribution, however each series now has a stan-

standard deviation $\sigma = 1/\sqrt{250} = 0.06324$. The histogram of the eigenvalues for the 10,000 \mathcal{C} matrices for $Q = 8.33$ is also normalized to unit area in order to be compared to the theoretical eigenvalues' distribution. The histogram is represented again as a dotted line following the middle theoretical graph in Figure 5.26. Finally, the null hypothesis for the five-year interval has $T = 1250$ and $\sigma = 1/\sqrt{1250} = 0.02828$. The eigenvalues histogram for the 10,000 \mathcal{C} matrices ($Q = 41.7$) obtained with these values and normalized to unit area is represented by the dotted line following the narrowest theoretical plot in Figure 5.26. Notice that these spectra are in line with what is expected for infinite matrices. Nevertheless, a close inspection of Figure 5.26 reveals longer tails for the spectra of the finite sized random matrices.

The results for the empirical correlation matrices that deviate from the ones obtained for random matrices are the ones that carry information about market behavior. Hence the relevant eigenvalues of the empirical correlation matrices must be above 3.3 for the quarterly calculation, above 2 for the annual cases, and above 1.4 for the five-year periods.

For risk management purposes the most significant eigenvalues are the smallest ones. They correspond to the directions of minimum correlation. The components of the corresponding eigenvectors indicate the composition of the least risky portfolio [53]. Unfortunately, the smallest eigenvalues are the most sensitive to noise and in most cases cannot be determined. Nevertheless, the study of the correlation matrix eigenvalues, while inappropriate for risk management, is of use in determining the general behavior of the market. The largest eigenvalues are way above the noise level and can give

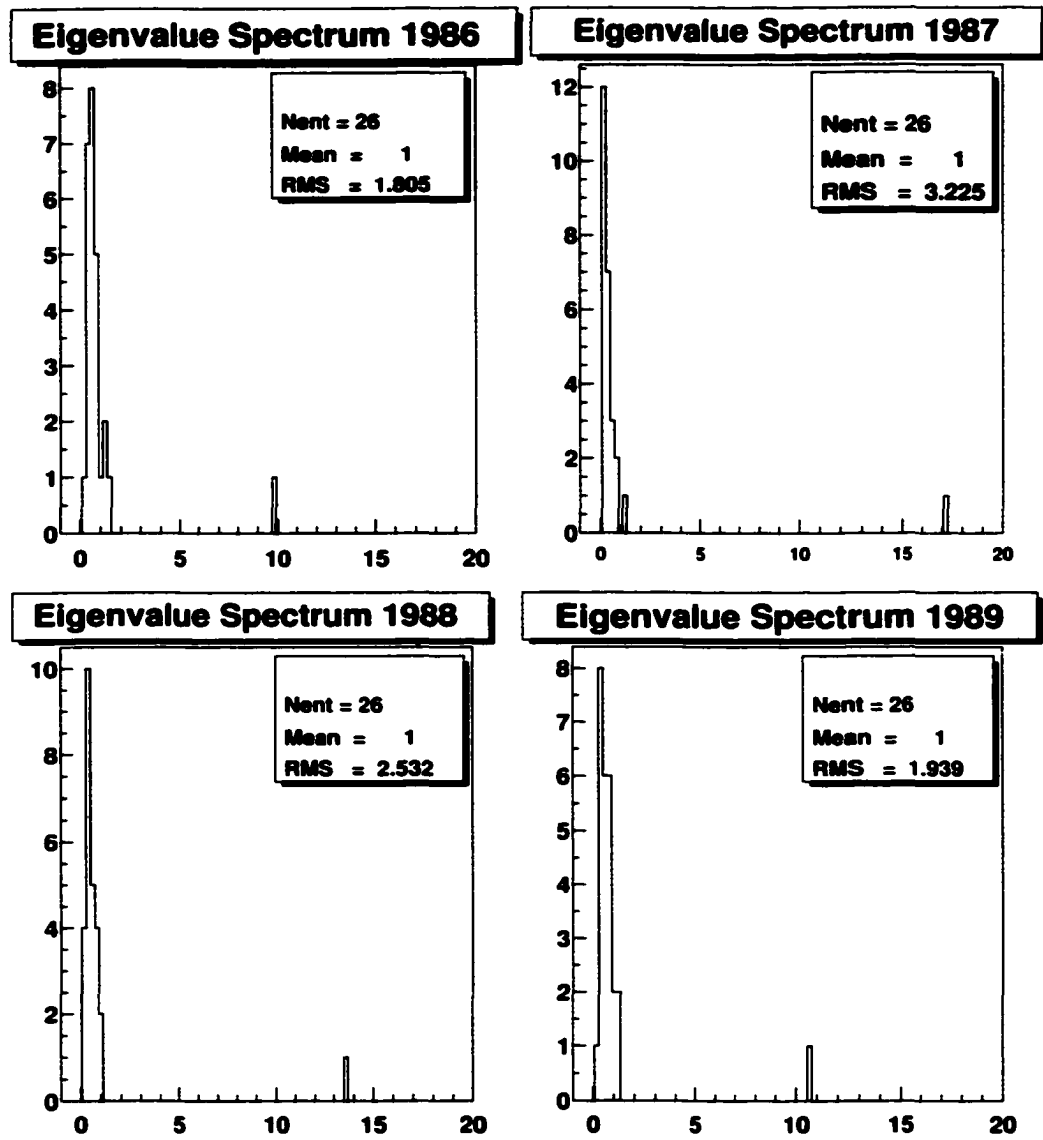


Figure 5.27: Eigenvalue spectra of the annual correlation matrices for the 26 major US companies studied during the years 1986-1989.

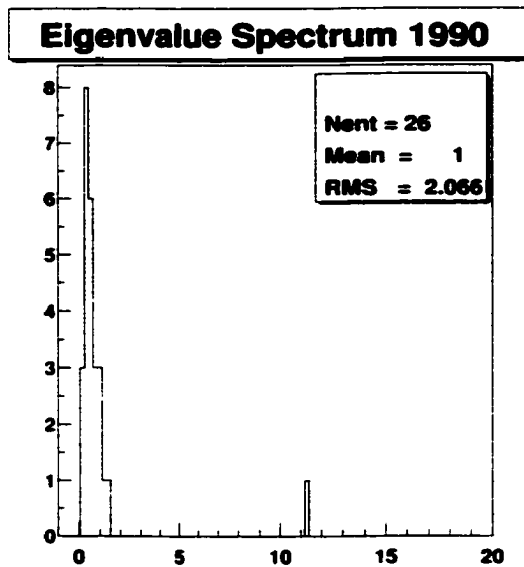


Figure 5.28: Eigenvalue spectrum of the annual correlation matrix for the 26 major US companies studied during the year 1990.

a good indication about the number of determining factors in the market.

Figures 5.27 and 5.28 present the eigenvalues of the annual correlation matrices for the 26 assets studied during the interval 1986-1990. Notice that most of the eigenvalues are within the noise level, *i. e.* smaller than 2. For the years 1986-1990, there is only one large eigenvalue of the correlation matrix, which corresponds to the market itself behaving in a unitary fashion. This observation is consistent with the one-index model [49], which assumes that the returns of all assets are controlled by one factor.

Similar observations can be made regarding the first two eigenvalue spectra of the annual correlation matrices for the 30 DJIA components examined

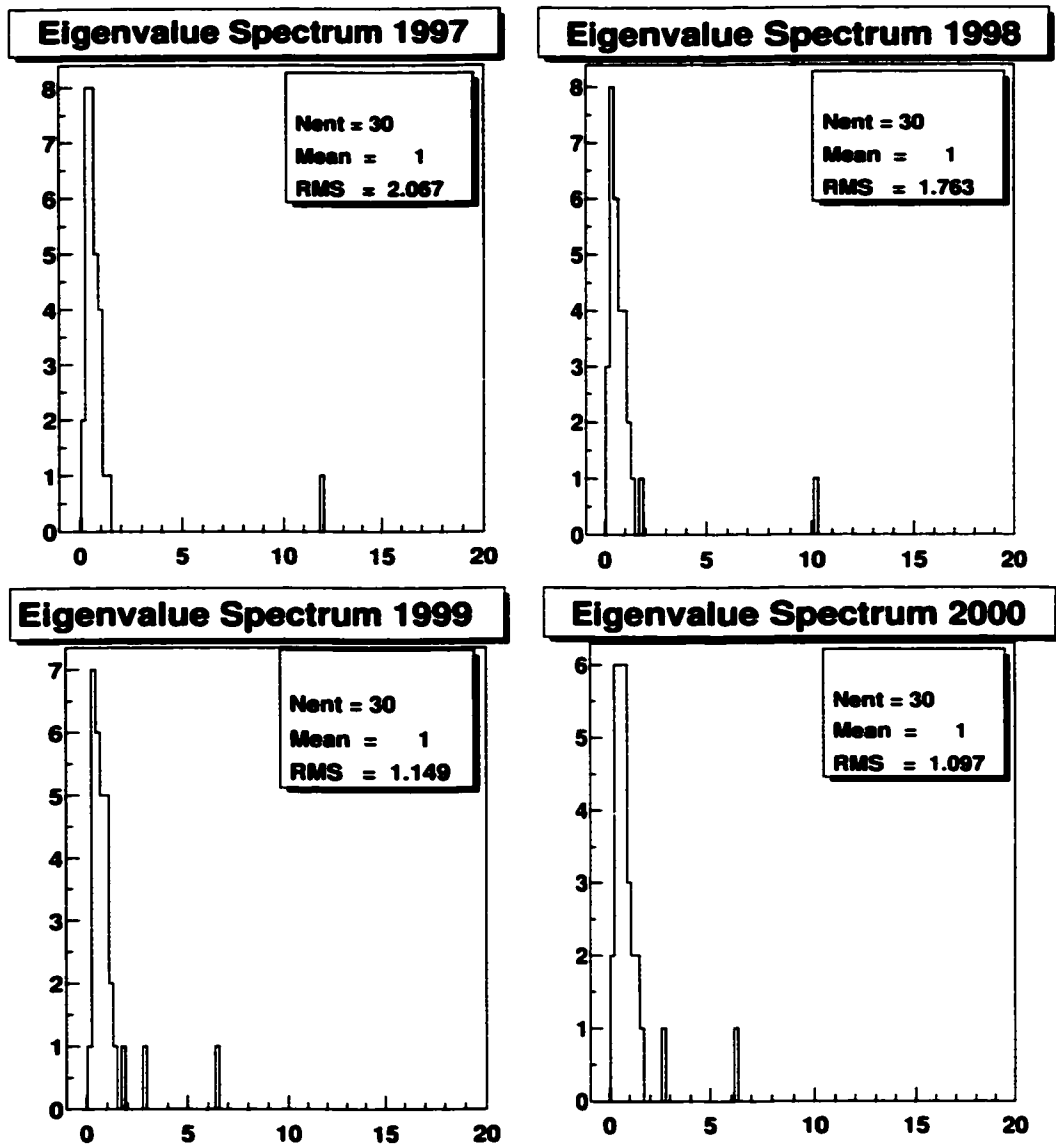


Figure 5.29: Eigenvalue spectra of the annual correlation matrices for the 30 DJIA components studied during the years 1997-2000.

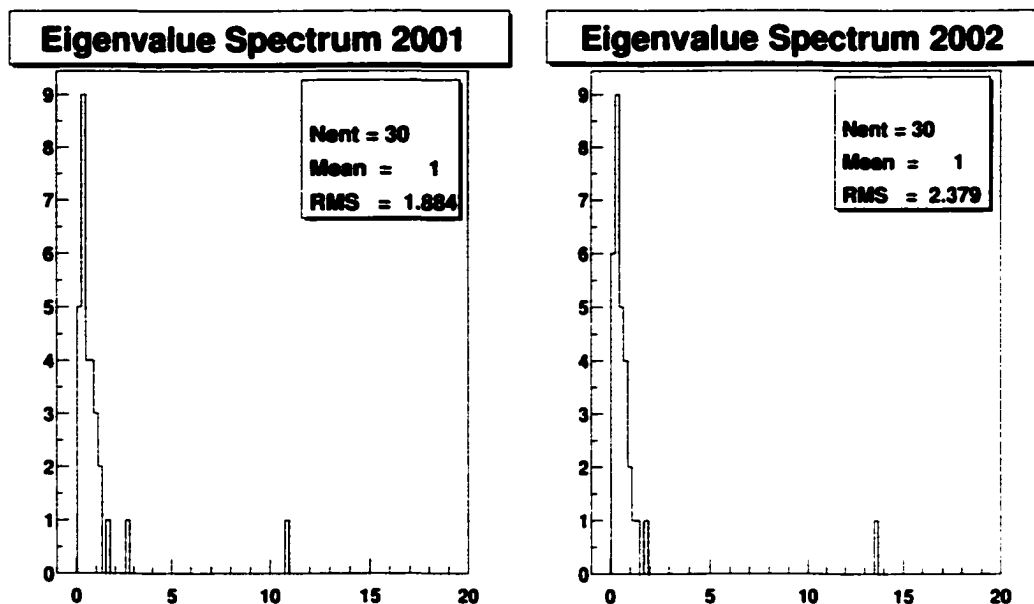


Figure 5.30: *Left*: Eigenvalue spectrum of the annual correlation matrix for the 30 DJIA components studied during the year 2001. *Right*: Eigenvalue spectrum of the annual correlation matrix for the 30 DJIA components studied during the year 2002.

during the period 1997-2001. Figures 5.29 and 5.30 illustrate these spectra. The years 1997 and 1998 are bullish years and the unitary behavior of the market is reflected in the existence of only one eigenvalue outside the noise level in each year. For the year 1999, when the market experienced unprecedented growth with the expansion of the dot-com bubble, the spectrum shows two prominent eigenvalues. We therefore hypothesize that the market has begun to lose its strength and stability, preparing itself for the

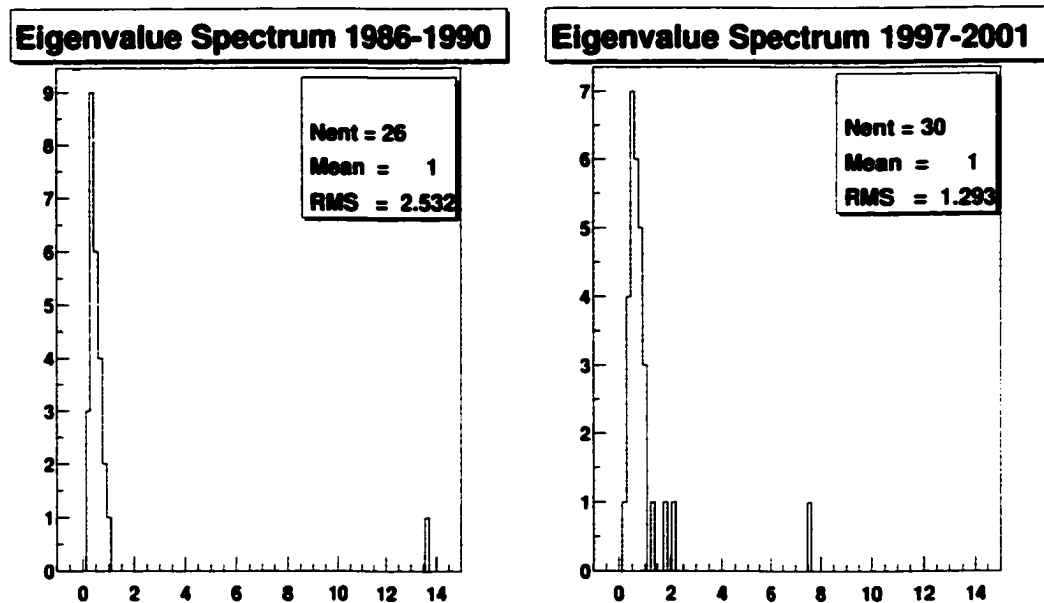


Figure 5.31: *Left*: Eigenvalue spectrum of the five-year correlation matrix for the 26 major US companies studied during 1986-1990. *Right*: Eigenvalue spectrum of the five-year correlation matrix for the 30 DJIA components analyzed during 1997-2001.

major change encountered in the following years. The phenomenon of non-unitary behavior is seen to continue in 2000 and 2001, which is implied by the existence of two significant eigenvalues in the spectra of these years. As one can see from the Figure 5.30, *right*, the year 2002 has a less prominent second largest eigenvalue, which might suggest that the market regained its unity and probably the major transition already took place.

The analysis of the eigenvalue histograms for the two five-year correlation

matrices, shown in Figure 5.31, reveals the same characteristic of the market behavior. Remember that the noise level is reduced, as mentioned previously, so that now eigenvalues above 1.4 are significant. During the first studied interval, 1986-1990, the one factor model seems to work well, as illustrated by the single eigenvalue above the noise threshold. The second period of 1997-2001, however, exhibits a markedly different characteristic. There are three significant eigenvalues, with one still being much larger than the others and therefore measuring the general market correlation. We can explain the other two eigenvalues, closer to the noise level, as new factors that influence the market. During this period, in the late 1990's, the DJIA was changed to contain stocks traded on the NYSE as well as NASDAQ, and these two markets have different trading procedures. Additionally, the broader ownership of stocks and the advent and popularity of online trading act to defocus the market and make it behave as less of a unit.

The eigenvalue spectra for the quarterly cross-correlation matrices are presented in Appendix C. The noise level is larger in this case due to the shorter length of the time series. This limits our ability to draw any strong conclusions about the numbers of factors that determine the market behavior.

The deviation of the market behavior from the one-factor model is asserted in many recent studies. The distribution of financial time series central moments (mean and standard deviation) are shown to deviate from the similar distributions obtained with one index model [56]. Similarly, the analysis of the correlation based MST for the financial assets and for the one factor model reveals important dissimilarities [57]. All these analyses have

been performed for a large number of assets and subsequently required long time series which include several market cycles.

By restricting our attention to a relatively small number of less volatile “blue chip” stocks, we were able to lessen the length of the time series and analyze separately different market conditions. The size of the chosen portfolio is still large enough to conform to the infinite random matrices theory, which helps us eliminate the noise.

5.7 The Meaning of the Correlation Matrix Eigenvectors

An interesting behavior of the annual correlation matrix between the studied assets emerges during the last analyzed five-year interval, 1997-2001. The second largest eigenvalue of the correlation matrix exceeds the noise level during the years 1999 through 2001, which corresponds to the transition years from a “bull” to a “bear” market. In order to focus on this transition, the current section examines the annual correlation matrix from 1997 to 2001 as well as the year 2002, which was added as a “post-transition” trial year. For comparison purposes several previous years, 1986-1990, have been added for some parts of the analysis. The eigenvectors will be named in descending order of their corresponding eigenvalues, thus the first eigenvector corresponds to the largest eigenvalue of the correlation matrix, the second one corresponds to the second largest eigenvalue and so on.

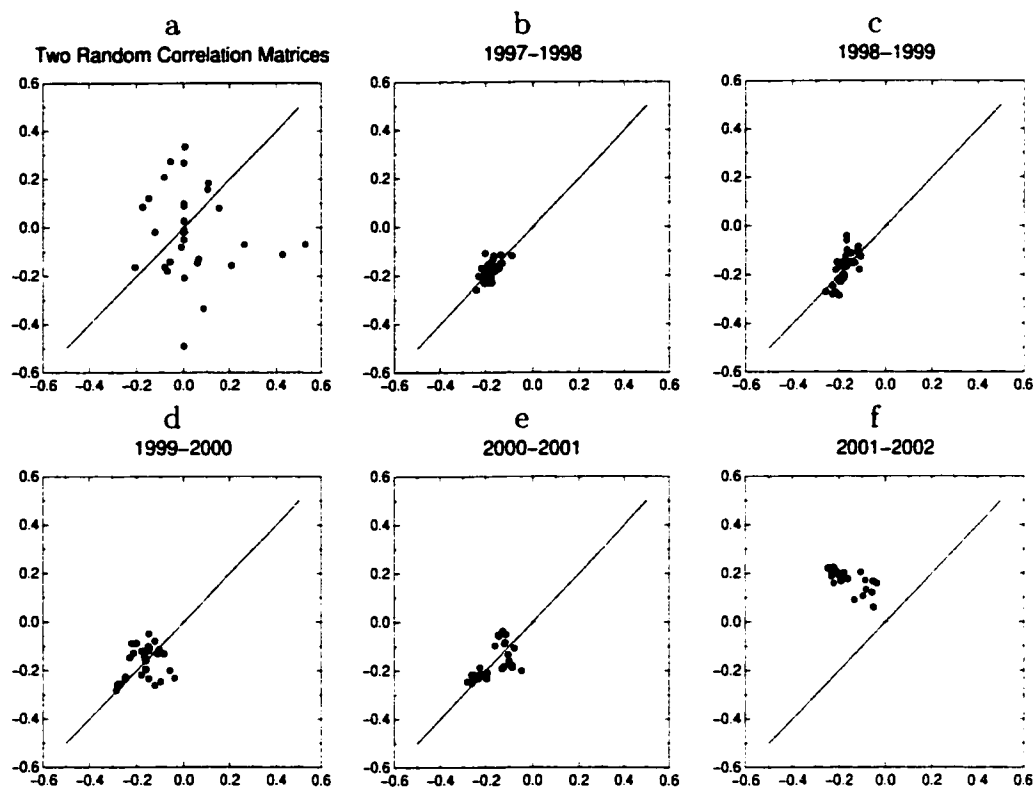


Figure 5.32: *a*: First eigenvector components for two random matrices. *b*, *c*, *d*, *e*, *f*: First eigenvector components for yearly correlation matrix during the years 1998 through 2002 as a function of the similar components during the previous years.

As previously mentioned in many publications [54, 58], the first eigenvector reflects the correlation existing in the market as a whole and its components along all assets have almost equal values. This property can be seen in Figure 5.32 which presents the first eigenvector components during one year as a function of the same components in the previous year for the interval 1997-2002. For ease of visualization, the first bisector is drawn to underline the equality between components and the second bisector is also shown in

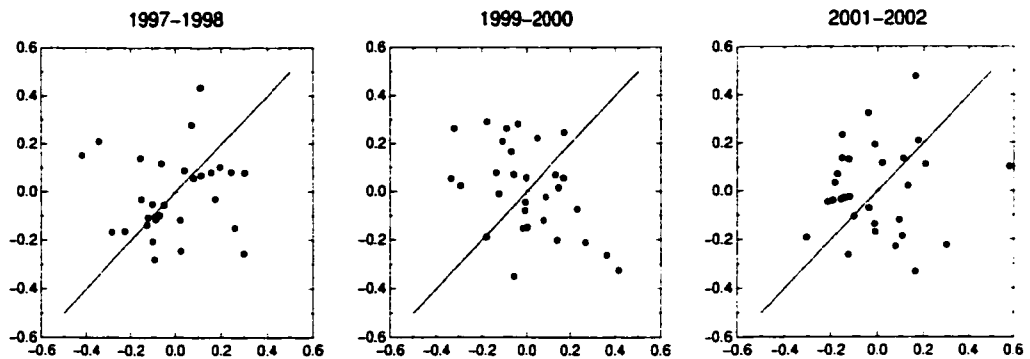


Figure 5.33: Tenth eigenvector components for the years 1998, 2000 and 2002 a function of the similar components during the previous years.

order to emphasize the cases when the components have equal magnitude but opposite signs. The sign of a specific eigenvector is determined merely on the normalization condition and has no practical meaning. One notices the narrow distribution of the first eigenvector components as well as their nonzero average, as opposed to the null hypothesis represented in Figure 5.32a. The null hypothesis diagram represents the first eigenvector components of two correlation matrices between sets of 30 series of “white noise”. Each series contains 250 independent, identically distributed random numbers chosen from a zero-mean Gaussian distribution of standard deviation $\sigma = 0.063$. Comparing diagram *a* in Figure 5.32 with the ones *b* through *f*, it is obvious that the first eigenvector components of the empirical correlation matrices contain information about the asset portfolio.

In fact, the same difference can be noticed between the first eigenvector components and the components describing the eigenvectors corresponding to the eigenvalues below the noise level. As an example, Figure 5.33 presents

the components of the tenth eigenvector of the annual correlation matrix for the years 1998, 2000 and 2002 as a function of the same components during the previous years. These diagrams reveal a “white noise” distribution of the tenth eigenvector components very similar to the one presented in Figure 5.32a. Therefore, the consensus is that the eigenvectors corresponding to the eigenvalues far beneath the noise limit given in equation (5.23), do not carry discernable information about the analyzed portfolio.

Returning the discussion to the components of the first eigenvector, one notices the robustness of its components over time. This behavior is very similar to one exhibited by the first eigenvector of a trivial correlation matrix \mathcal{C} which has the diagonal elements $c_{ii} = 1$ and all non-diagonal ones of constant value $c_{ij} = \rho$, where $i \neq j$ and $\rho \in (0, 1)$. Such a matrix, which mirrors a system of equally correlated assets, has one large eigenvalue given by [59]:

$$\Lambda_1 = 1 + (N - 1)\rho, \quad (5.24)$$

and $N - 1$ degenerate eigenvalues $\Lambda_{i \geq 2} = 1 - \rho$. Note that as long as N is large, the correlation between the assets does not have to be strong in order for Λ_1 to have a large value. In other words, a large first eigenvalue does not necessarily reflect a strong correlation between assets, but arises in large systems due to the non-vanishing average correlation between a large fraction of its elements [59]. The eigenvector \mathbf{v}_1 (corresponding to the largest eigenvalue Λ_1) is delocalized, with all its components equal to $1/\sqrt{N}$. For $N = 30$, these components are all equal to 0.182, which is a value close to

those exhibited by the empirical first eigenvector components presented in Figure 5.32. Moreover, the largest eigenvalues of the empirical correlation matrices seem to follow equation (5.24), with the constant ρ replaced by the average correlation coefficient, $\bar{\rho}$, during the corresponding year. Table 5.7 lists the average correlation coefficients over all thirty assets, $\bar{\rho}$, their standard deviations, σ , the largest eigenvalue Λ_1 , calculated according to equation (5.24), and the empirical largest eigenvalue, λ_1 , for the years 1986-1990 and 1997-2002. Notice the close values in the last two columns of this table, which shows that the largest empirical eigenvalue, λ_1 , has nearly the same magnitude as the largest eigenvalue, Λ_1 , of the trivial correlation matrix, \mathcal{C} , described above. Thus the first eigenvector represents the global correlation between the studied assets. It identifies the main factor in the market, not in the sense that it is the strongest, but in the sense that it simultaneously influences all stocks.

The small discrepancies between the values predicted by equation (5.24) and the empirical eigenvalues λ_1 can be attributed to the noise inherent in the time series. In fact, an eigenvalue spectrum closer to the empirical one, *i. e.* with multiple smaller eigenvalues, may be obtained by adding a small random component to the non-diagonal elements of the trivial correlation matrix \mathcal{C} , such that:

$$c_{ij} = \rho + \epsilon \cdot a_{ij}. \quad (5.25)$$

The coefficients $a_{ij} = a_{ji}$ are generated from a zero mean Gaussian distribution of standard deviation σ and fulfill all the necessary constraints such that

Year	$\bar{\rho}$	σ	Λ_1	λ_1
1986	0.343	0.112	10.9	9.9
1987	0.637	0.103	19.5	17.1
1988	0.496	0.101	15.4	13.6
1989	0.378	0.088	11.9	10.6
1990	0.384	0.119	12.13	11.2
1997	0.369	0.106	11.7	12.0
1998	0.306	0.116	9.9	10.3
1999	0.172	0.126	6.0	6.5
2000	0.158	0.128	5.6	6.2
2001	0.298	0.183	9.6	10.8
2002	0.415	0.150	13.0	13.6

Table 5.7: Average correlation coefficients over all thirty assets, $\bar{\rho}$, their standard deviations, σ , the largest eigenvalue Λ_1 , calculated according to equation (5.24) and the empirical largest eigenvalue, λ_1 , for the years 1986-1990 and 1997-2002.

the newly obtained matrix is positive definite and has a probability of one.

The largest eigenvalue for such a matrix has the expectation value [59]:

$$E[\Lambda_1] = 1 + (N-1)\rho + \frac{(N-1)(N-2)}{N^2} \cdot \frac{\epsilon^2 \sigma^2}{\rho} + \mathcal{O}(\epsilon^3), \quad (5.26)$$

and the expectation value for the second largest eigenvalue is given by:

$$E[\Lambda_2] \leq 2\sigma\sqrt{N} + \mathcal{O}(N^{1/3} \log N). \quad (5.27)$$

Such a matrix replaces the degenerate second eigenvalue of the trivial correlation matrix \mathcal{C} with a complex set of small eigenvalues. As one can see from equation (5.26), noise superimposed on a trivial correlation matrix produces an increase in the expectation value of the largest eigenvalue Λ_1 . Notice in Table 5.7 that for all the years of the first studied interval, 1986-1990, the

largest empirical eigenvalue, λ_1 , is smaller than the one obtained from the trivial correlation matrix. This is not unexpected, as equation (5.26) refers to the expectation value and not to the exact eigenvalue. However, in the next six analyzed years 1997-2002, all the empirical eigenvalues λ_1 are larger than those given by equation (5.24).

Once the meaning of the first eigenvalue and eigenvector are established, it is normal to continue the analysis with the second eigenvector and its corresponding eigenvalue, especially when this last one exceeds the noise level. There is an interplay between the magnitude of different eigenvalues. Representing a matrix along its eigenvectors corresponds to a rotation and translation operation. These operations preserve the matrix's trace. Since the number of studied assets is 30, the trace of the empirical correlation matrix is also 30 and this equals the sum of all of the eigenvalues. For a given system size N , the first eigenvalue grows along with the increase in the average correlation coefficient $\bar{\rho}$, and therefore ρ , as shown in equations (5.24) and (5.26). The larger the first eigenvalue, the smaller the rest of the eigenvalues and hence it is more likely that the second eigenvalue is under the noise level. The reverse situation is also true, in that when the first eigenvalue is relatively small, the other eigenvalues increase and the second eigenvalue has a higher probability of being above the noise. In other words, when the global, *i. e.* inter-cluster, correlation weakens, other local correlations become more visible. Such a situation is encountered during the years 1999 and 2000, when the average correlation coefficient reaches its lowest values for the interval 1997-2001, as shown in Figure 5.21.

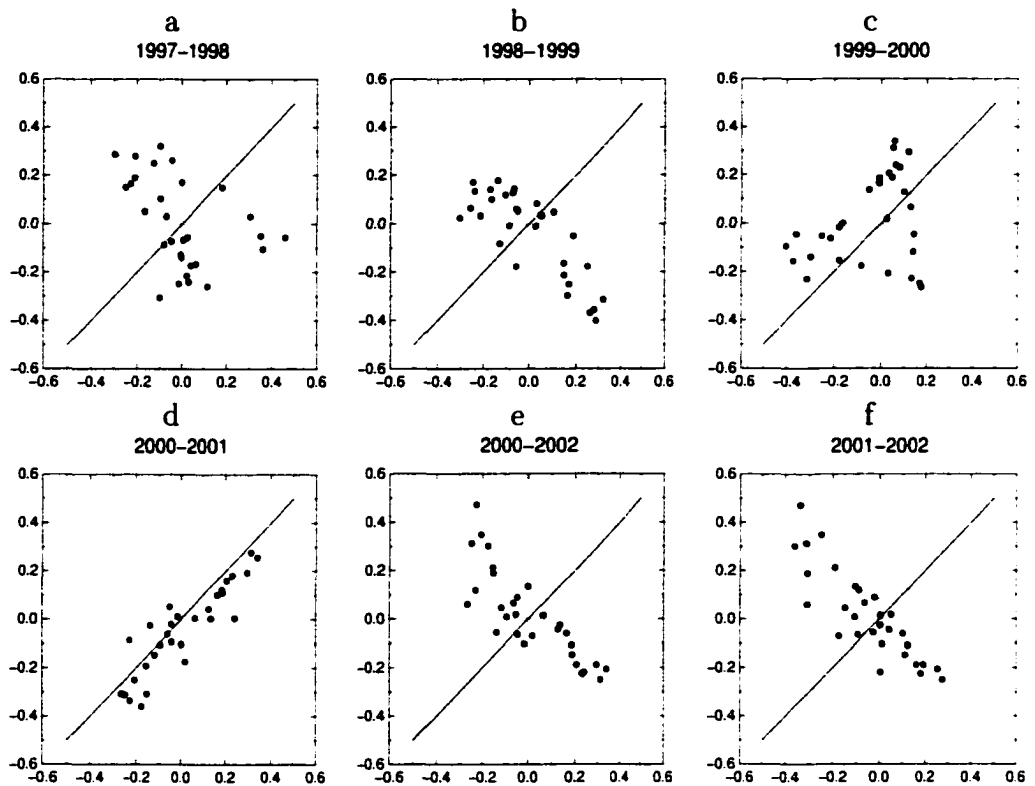


Figure 5.34: *a, b, c, d, e, f*: Second eigenvector components for yearly correlation matrix during the years 1998 through 2002 as a function of the similar components during the previous years.

Keep in mind that out of the interval 1997-2002, only three years have the second eigenvalue above the noise level: 1999, 2000 and 2001. Analyzing the components of the second eigenvector in a similar way to what was done for the first eigenvector components, the results presented in Figure 5.34 are obtained. Notice the small dispersion of the data points with respect to the first or second bisector in diagrams *b, d, e* and *f*. The low dispersion occurs mainly when at least one of the two considered years has the second

eigenvalue above the noise level and is the lowest when both years have a large second eigenvalue, as is the case of diagram *d*. This observation implies that the second eigenvector components have a time stability from one year to the next as long as market conditions are similar. This caveat helps to explain the anomalous behavior of plot *c* (1999-2000), in which the dispersion is large in spite of the large second eigenvalue for both years. The year 1999 was a very bullish year, whereas 2000 was a down year, due to the start of the dot-com collapse. This implies that in a market transition the components of the second eigenvector change.

In order to analyze the time stability of the second eigenvector components from one year to the next, we represent graphically pairs of successive years out of which at least one has the corresponding eigenvalues above the noise level. Figures 5.35, 5.36 and 5.37 illustrate these projections for the years 1998-1999, 2000-2001 and 2001-2002 respectively. If the second eigenvector projections align along the second bisector, as seen in Figure 5.34*b* and *f*, it means that the components have equal magnitude but opposite signs, thus the components for one of the paired years must be represented with an opposite sign. Therefore the components for the years 1998 and 2002 are represented with a reversed sign.

At this point the natural question is what is the information carried by the second eigenvector. Our analysis suggests that the second eigenvector expresses the correlation inside clusters of stocks corresponding to different industrial primary groups, or classes of assets with similar economic performance and corporate health. Though independent of, *i. e.* orthogonal to.

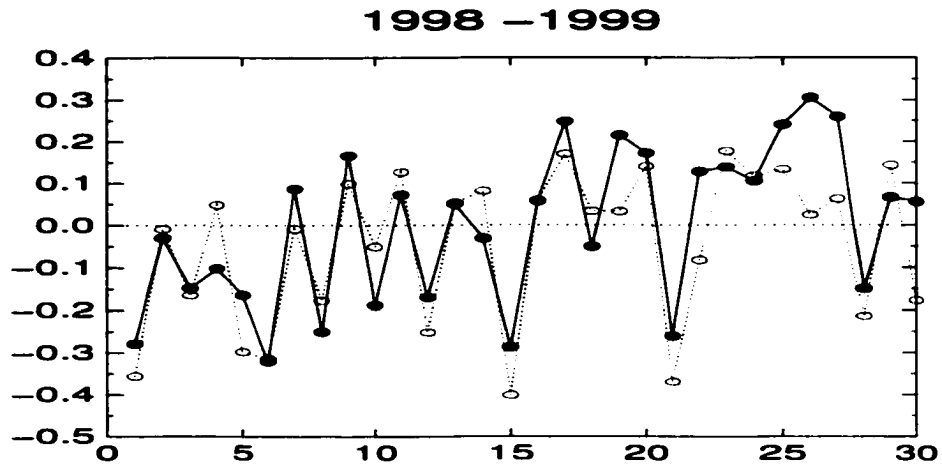


Figure 5.35: Second eigenvector projections along the 30 DJIA components for the yearly correlation matrix during the years 1998 (solid line) and 1999 (dotted line). The components for the year 1998 are represented with an inversed sign.

the whole market behavior expressed by first eigenvector, the second eigenvector becomes relevant in the transition periods. In other words, the second eigenvalue becomes larger than the noise when the market as a whole loses its strength and the stocks are not traded indiscriminately, but depending more on the groups they belong to.

In order to quantitatively decide which of the second eigenvector components are significant, we used the concept of the inverse participation ratio (IPR) [58]. The IPR of an eigenvector \mathbf{v}_k is defined as:

$$I_k = \sum_{i=1}^N (v_{ki})^4, \quad (5.28)$$

where v_{ki} is the eigenvector k projection on asset i ($i = 1, \dots, N$). The meaning of the IPR is easy to understand when one considers the example

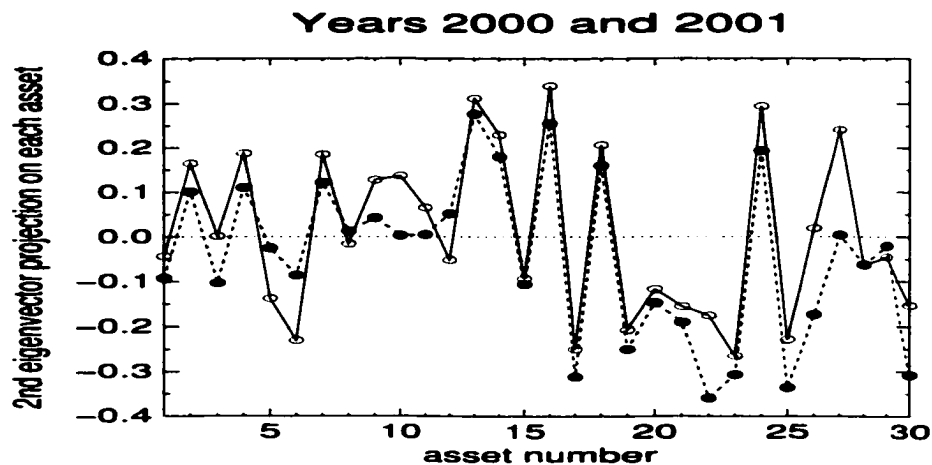


Figure 5.36: Second eigenvector projections along the 30 DJIA components for the yearly correlation matrix during the years 2000 (solid line) and 2001 (dotted line).

of a delocalized vector \mathbf{v}_k with identical components $v_{ki} \equiv 1/\sqrt{N}$. According to definition (5.28), its inverse participation ratio is $I_k = 1/N$ which means that the parameter I_k quantifies the inverse number of significant eigenvector components. Thus, the participation of the eigenvector \mathbf{v}_k is calculated as $1/I_k$.

The average participation over all 30 eigenvectors compared with the participation of the first and second eigenvectors during the interval 1997-2002 is presented in Table 5.8. Notice that the participation of the first eigenvector is at least twice that of the average participation over all 30 eigenvectors, while the participation of the second eigenvector is about the same value as the average, which is consistent with the observation that the second eigenvalue is highly susceptible to noise. The smaller the participation

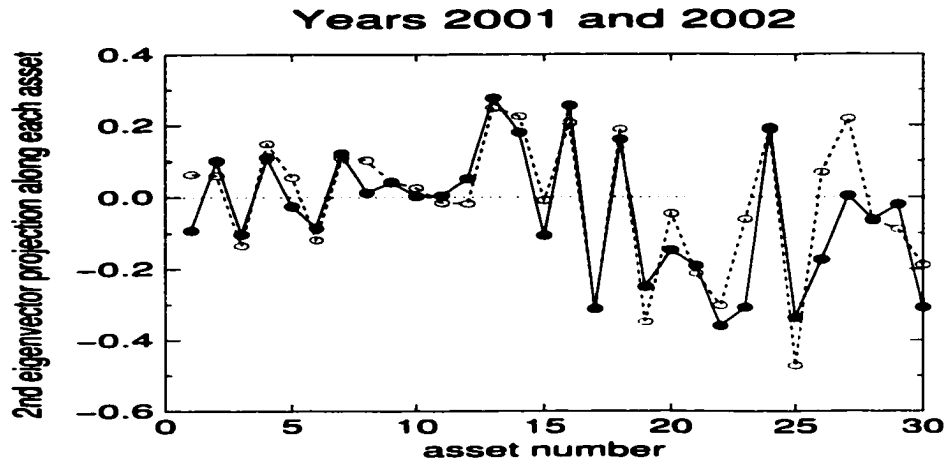


Figure 5.37: Second eigenvector projections along the 30 DJIA components for the yearly correlation matrix during the years 2001 (solid line) and 2002 (dotted line). The components for the year 2002 are represented with an inversed sign.

of one vector, the closer the distribution of its components to that predicted by Random Matrix Theory.

Based on the participation of the second eigenvector for each year, listed in Table 5.8, we calculate the average inverse participation ratio associated to one component as:

$$I_2^{avg} = \frac{I_2}{N}, \quad (5.29)$$

where I_2 is the IPR of second eigenvector and $N = 30$. The significant components j are the ones whose participation at I_2 is equal to or larger than the average, thus their magnitude $|v_{2j}|$ is:

$$|v_{2j}| \geq (I_2^{avg})^{1/4}, \quad (5.30)$$

where $j = 1, \dots, N$. Table 5.9 presents the second eigenvector inverse partic-

Year	1997	1998	1999	2000	2001	2002
$1/I_1$	27.24	25.92	20.56	20.36	22.60	26.38
$1/I_2$	9.33	16.43	11.01	16.42	12.94	10.03
$1/\bar{I}$	10.66	10.06	10.27	10.56	9.88	9.23

Table 5.8: Numbers of significant components for the first the second eigenvectors as well as the average over all eigenvectors of the annually correlation matrix during the interval 1997-2002.

Year	1997	1998	1999	2000	2001	2002
I_2	0.1070	0.0609	0.0908	0.0609	0.0774	0.0997
$I_2^{avg} (\times 10^{-3})$	3.57	2.03	3.03	2.03	2.58	3.32
$ v_{2j} _{min}$	0.244	0.212	0.234	0.212	0.225	0.240

Table 5.9: Second eigenvector inverse participation ratio, I_2 , the average inverse participation associated to each component, I_2^{avg} , and the minimum value of the projection considered significant $|v_{2j}|_{min}$ for the interval 1997-2002.

ipation ratio I_2 , the average inverse participation associated to each component I_2^{avg} , as well as the minimum value of the projection considered significant $|v_{2j}|_{min}$ calculated with equation (5.30) for each of years 1997-2002.

According to the minimum significant projection values listed in Table 5.9, we choose the significant components of the second eigenvector for the selected years and list their ticker symbol in Table 5.10.

The grid represented in this table shows stocks that move in a direction separate from the market as a whole, *i. e.* lagging or leading it. In 1997, the technology sector was outpacing the rest of the DJIA components, and this is visible in the cluster of stocks containing HPQ, IBM, INTC and MSFT. The

Year	1997	1998	1999	2000	2001	2002
1	-	AA	AA	-	-	-
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	-	-	-	-	-	-
5	-	-	CAT	-	-	-
6	-	DD	DD	DD	-	-
7	-	-	-	-	-	-
8	-	EK	-	-	-	-
9	-	-	-	-	-	-
10	-	-	-	-	-	-
11	-	-	-	-	-	-
12	-	-	HON	-	-	-
13	HPQ	-	-	HPQ	HPQ	HPQ
14	IBM	-	-	IBM	-	-
15	INTC	INTC	INTC	-	-	-
16	IP	-	-	IP	IP	-
17	-	JNJ	-	JNJ	JNJ	JNJ
18	-	-	-	-	-	-
19	-	KO	-	-	KO	KO
20	-	-	-	-	-	-
21	-	MMM	MMM	-	-	-
22	-	-	-	-	MO	MO
23	-	-	-	MRK	MRK	-
24	MSFT	-	-	MSFT	-	-
25	-	PG	-	PG	PG	PG
26	-	SBC	-	-	-	-
27	-	AT&T	-	AT&T	-	-
28	UTX	-	-	-	-	-
29	-	-	-	-	-	-
30	-	-	-	-	XOM	-

Table 5.10: Significant components of the second eigenvector for the interval 1997-2002.

technology bubble also lifted telecommunication stock interest, and we see a cluster containing SBC and T in 1998. By 1998 and 1999, the heavy industry sector, containing AA, DD, MMM, HON and CAT, was lagging the market as investors rushed into technology stocks. In these two years, the market as a whole followed the technology sector and therefore the cluster seen in 1997 disappeared, being included in the global market behavior. In the period 2000-2002, the technology and pharmaceutical sectors are moving downward faster than the rest of the market, as seen in the cluster containing HPQ, IBM, JNJ, MRK, MSFT, T. Individual corporate effects can cause companies to appear in as significant components of the second eigenvector, although they are not following their primary industrial groups. The cause of this may be due to mergers, lawsuits and so on. This may explain the presence of IP and UTX in 1997, KO and PG in 1998, 2001 and 2002, XOM in 2001. etc.

In order to elucidate the significance of the second eigenvector and its components, a new procedure has been suggested [58]. The method proposes to eliminate the contribution of the first eigenvector, which can mask some of the information carried by other eigenvectors. First one calculates the scalar product of the first eigenvector on all the assets' time series:

$$G_1(t) = \sum_{i=1}^{30} v_{1i} S_i(t) \quad (5.31)$$

where v_{1i} is the projection of the first eigenvector on asset i and $S_i(t)$ is the logarithmic closing price variation of the same asset, defined in equation (5.4).

A linear regression is then done on the return time series of each stock, as:

$$S_i(t) = \alpha_i + \beta_i G_1(t) + \epsilon_i(t) \quad (5.32)$$

and to calculate the residuals $\epsilon_i(t)$. The correlation matrix \mathcal{K} of the residuals is then analyzed and its eigenvectors are computed. The last step is to choose the significant participants of the analyzed eigenvector as explained above. The major components for the first eigenvector of matrix \mathcal{K} during the interval 1997-2002 do not correspond to the significant components of the second eigenvector of the empirical correlation matrix. This is an area of ongoing research and a larger number of assets has to be analyzed in order to firmly establish the significance of these eigenvectors.

Chapter 6

Summary and Suggestions for Future Work

The current interest in the development of new clustering techniques is due primarily to two factors. One is the vast amount of extremely diverse data facing all sectors of society on a daily basis. The other factor is the specificity of clustering procedures to particular characteristics of the data, *i. e.* the clustering is very dependent on the type of data and its structure. There are no clustering algorithms applicable to all situations and no absolute partitioning criteria. Each method has its own advantages and its own area of applicability. The recent use of statistical physics methods to clustering problems conveys strong validation criteria and allows a “natural” partitioning of the data with a minimum of assumptions. The three new clustering algorithms presented in the previous chapters are all based on physical phenomena.

The Percolation Clustering Algorithm is a version of the shortest linkage method and seems to work well for situations when the noise or outliers

cannot essentially distort the classification. Its advantage is that it can use a wide variety of similarity functions that can be adopted depending on the characteristics of the data set. There is no need to embed the sample points in a Euclidian space, which makes the technique applicable to a diverse set of problems such as the taxonomy of stock portfolios, biological classifications, etc. For the more complex case of intertwined clusters, a recursive application of the algorithm can give more reliable results. When there is prior information about sensitivity to noise, a procedure for eliminating the noise can then be applied.

The results of the Percolation Clustering Algorithm to econophysics are very encouraging in that we obtain clusters corresponding to primary industrial groups. Further confirmation of the algorithm would be to obtain economically significant classifications for larger portfolios, such as the components of the S&P 500 and the Russell 2000 indices during the same time periods discussed in this work (1986-1990 and 1997-2001). Also important is to verify the time stability of our results by analyzing other major market transition periods such as 1928 -1932 or the interval 1939-1943. This last interval includes, for the first time in history, three consecutive years in which the market finished lower than it started, similar to the years 2000-2002.

For cases when the sample data can be embedded in a Euclidian space and a metric can be defined between vector points, the analogy with real physical phenomena becomes more obvious. In such situations, one can imagine more robust algorithms by simulating a self-organizing evolution of the system which increases its chances to end up in the global minimum. At this stage

the Nucleation and Growth Algorithm offers a more noise-resistant clustering method than the Percolation Clustering Algorithm due to its randomly adding data points. This gives the noise and outliers a lower probability to disturb the “real” partitioning. Nevertheless, uphill moves are not included and the system cannot escape eventual local minima. A further development of the algorithm should introduce a temperature dependent mobility and interaction as well as a validity criterion of partitioning. All of these require a careful investigation of the motion in a continuous Euclidian space, which becomes more and more difficult with the increasing dimensionality of the problem. A careful, efficient implementation of this technique is a project for future work.

Since many clustering problems deal with large regions of empty space that increase with the increasing dimensionality of sample points, a random deposition in a continuum can be very time consuming. A natural development of the NGA would be to restrict the deposition sites to fixed points such as the Discrete Deposition Clustering Algorithm does. Future work regarding this last method is to apply it to different data sets in order to establish its robustness and limits. It is important to analyze the algorithm with different long and short range interaction functions, which would influence the convergence of the method as well as the sharpness of the simulated phase transition. These considerations become important when dealing with hierarchical clusters and when several phase transition are expected. Therefore a future project would be to use the DDCA on a data file which contains hierarchical clusters. The resolution of the obtained partition must be analyzed

in connection with different interaction functions.

Finally, in the study of the correlation matrix between the portfolio assets, the economic relevance of the eigenvectors corresponding to the significant eigenvalues has to be more firmly established. Studying the distribution of these components may offer important information about the main factors that influence the market.

Appendix A

DJIA Components

The appendix lists the stocks studied in the dissertation. The assets are presented in alphabetical order of their ticker symbol along with the company's name and the primary industrial group they belong to.

For the interval 1986-1990, due to the lack of complete historical records, the representative chosen assets are from the DJIA components tracked during the year 1991. These components are listed in Table A.1. Only 26 of these assets, recorded in normal font, have complete updated data and were analyzed; the other four, written in italic were overlooked.

During the years 1997 through 2002 the studied assets, listed in Table A.2, are the 30 DJIA components as defined for the year 2001.

A.1 DJIA Components for 1991

Ticker	Company Name	Primary Group
AA	Alcoa Inc.	Aluminum Commodity
ALD	Allied Signal	Industrial Diversified
AXP	American Express Co.	Diversified Financial
BA	Boeing Co.	Aerospace
BS	Bethlehem Steel	Steel Commodity
CAT	Caterpillar Inc.	Heavy Machinery
CHV	Chevron	Oil Company, Major
DD	E.I. DuPont de Nemours & Co.	Chemicals Commodity
DIS	Walt Disney Co.	Broadcasting
EK	Eastman Kodak Co.	Recreational Products & Services
GE	General Electric Co.	Industrial Diversified
GM	General Motors Corp.	Automobile Manufacturers
GT	Goodyear Tire Inc.	Tire & Rubber manufacturing
IBM	International Business Machines Corp.	Computers
IP	International Paper Co.	Paper Products
JPM	J.P. Morgan Chase & Co.	Banks, Ex-S&L
KO	Coca-Cola Co.	Soft Drinks
MCD	McDonald's Corp.	Restaurants
MMM	Minnesota Mining and Manufacturing Company	Heavy Industry
MO	Philip Morris Cos. Inc.	Tobacco
MRK	Merck & Co. Inc.	Pharmaceuticals
PG	Procter & Gamble Co.	Household Products, Nondurable
S	Sears, Roebuck & Co.	Retailers, Broadline
T	AT&T Corp.	Fixed-Line Communications
TX	Texaco	Oil Company, Major
UK	Union Carbide	Chemical Manufacturing
UTX	United Technologies Corp.	Aerospace, Industrial Diversified
WX	Westinghouse	Industrial Diversified
XOM	Exxon Mobil Corp.	Oil Company, Major
Z	Woolworth	Retailers, Broadline

Table A.1: Dow Jones Industrial Average components during 1991 listed in the order of their ticker symbol, together with the primary group they belong to.

A.2 DJIA Components for 2001

Ticker	Company Name	Primary Group
AA	Alcoa Inc.	Aluminum Commodity
AXP	American Express Co.	Diversified Financial
BA	Boeing Co.	Aerospace
C	Citigroup Inc.	Diversified Financial
CAT	Caterpillar Inc.	Heavy Machinery
DD	E.I. DuPont de Nemours & Co.	Chemicals Commodity
DIS	Walt Disney Co.	Broadcasting
EK	Eastman Kodak Co.	Recreational Products & Services
GE	General Electric Co.	Industrial Diversified
GM	General Motors Corp.	Automobile Manufactuters
HD	Home Depot Inc.	Retailers, Specialty
HON	Honeywell International Inc.	Industrial Diversified
HPQ	Hewlett-Packard Co.	Computers
IBM	International Business Machines Corp.	Computers
INTC	Intel Corp.	Semiconductors
IP	International Paper Co.	Paper Products
JNJ	Johnson & Johnson	Pharmaceuticals
JPM	J.P. Morgan Chase & Co.	Banks, Ex-S&L
KO	Coca-Cola Co.	Soft Drinks
MCD	McDonald's Corp.	Restaurants
MMM	Minnesota Mining and Manufacturing Company	Heavy Industry
MO	Philip Morris Cos. Inc.	Tobacco
MRK	Merck & Co. Inc.	Pharmaceuticals
MSFT	Microsoft Corp.	Software
PG	Procter & Gamble Co.	Household Products, Nondurable
SBC	SBC Communications Inc.	Fixed-Line Communications
T	AT&T Corp.	Fixed-Line Communications
UTX	United Technologies Corp.	Aerospace, Industrial Diversified
WMT	Wal-Mart Stores Inc.	Retailers, Broadline
XOM	Exxon Mobil Corp.	Oil Company, Major

Table A.2: Dow Jones Industrial Average components during 2001 listed in the order of their ticker symbol, together with the primary group they belong to.

Appendix B

Histograms of Quarterly Correlation Coefficients

The appendix presents the histograms of the quarterly correlation coefficients between the studied assets. There are 325 independent correlation coefficients between the 26 stocks analyzed during the interval 1986-1990 and 435 independent correlation coefficients between the 30 assets studied during the period 1997-2001. The range of the histograms and the number of bins have been chosen to best accommodate the correlation coefficient values and their distribution. For ease of comparison, all histograms during the first five-year time interval have the correlation coefficient in the range of -0.3 to 0.9 and 150 bins. The histograms for the second interval are displayed with a correlation coefficient range of -0.4 to 0.8 and 150 bins.

Note the Gaussian shape of the histograms during some of the analyzed quarters and the deviation from this shape for other quarters. Usually, as one can see from the Tables 5.1 and 5.2, these deviations correspond to down market intervals.

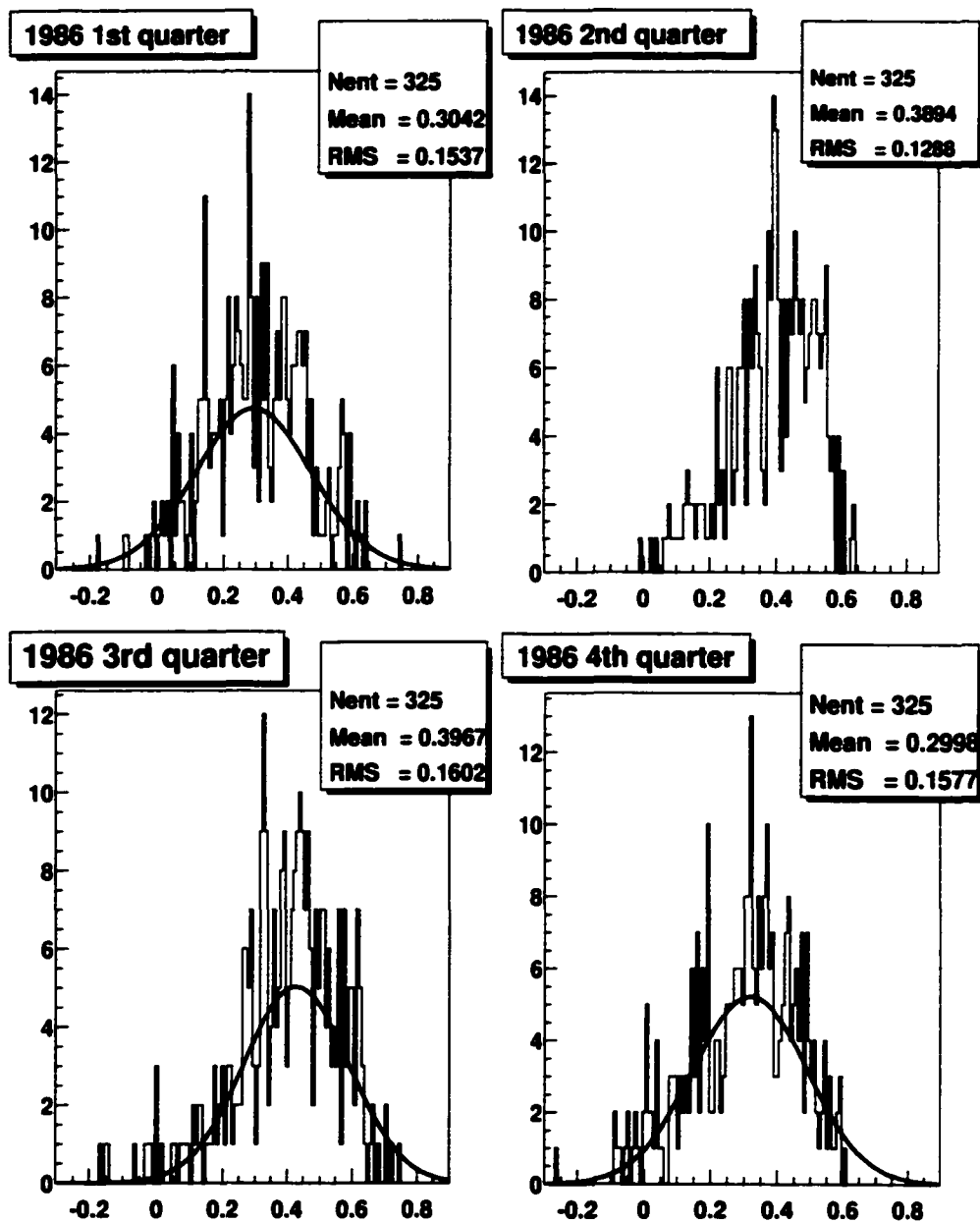


Figure B.1: Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1986.

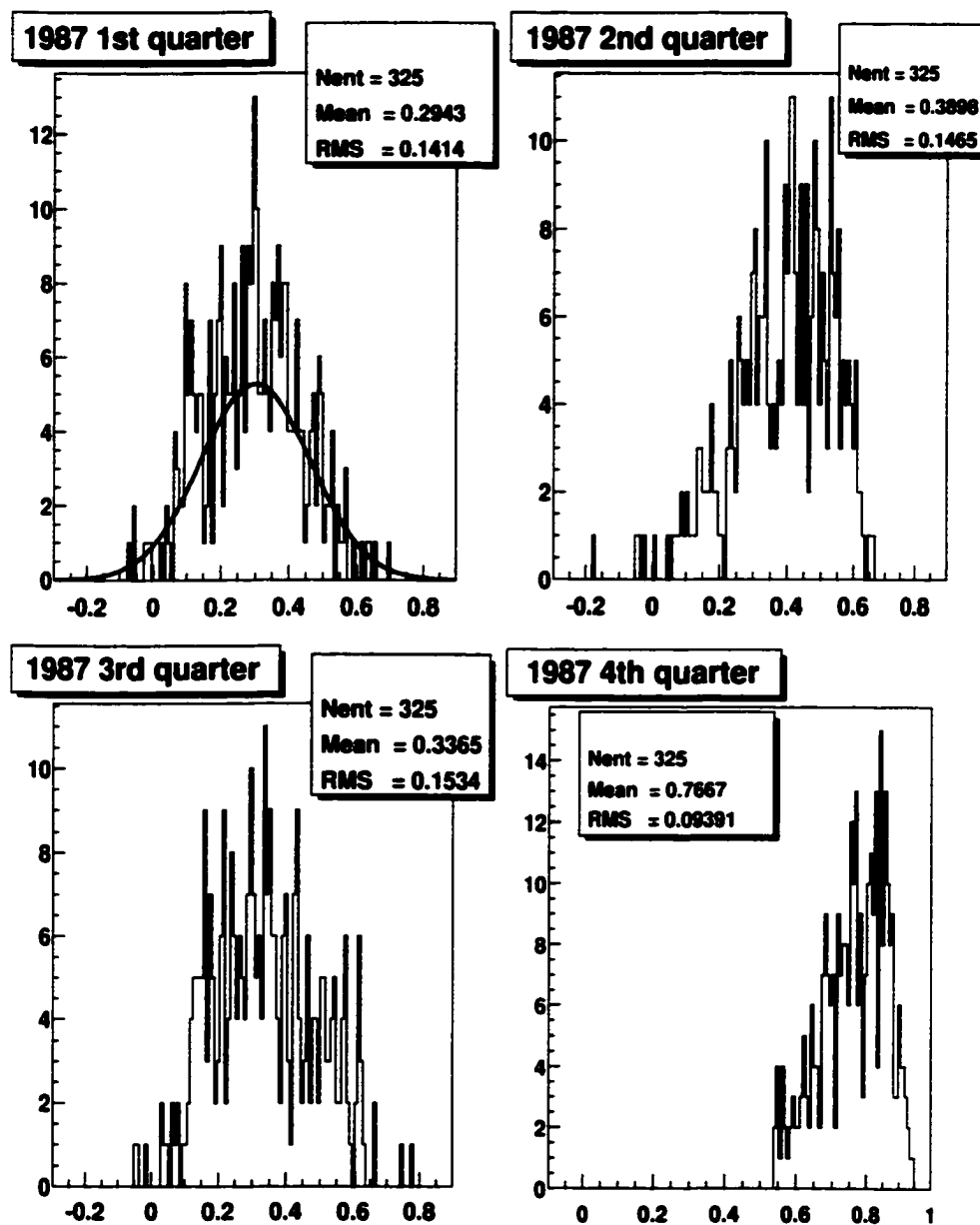


Figure B.2: Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1987.

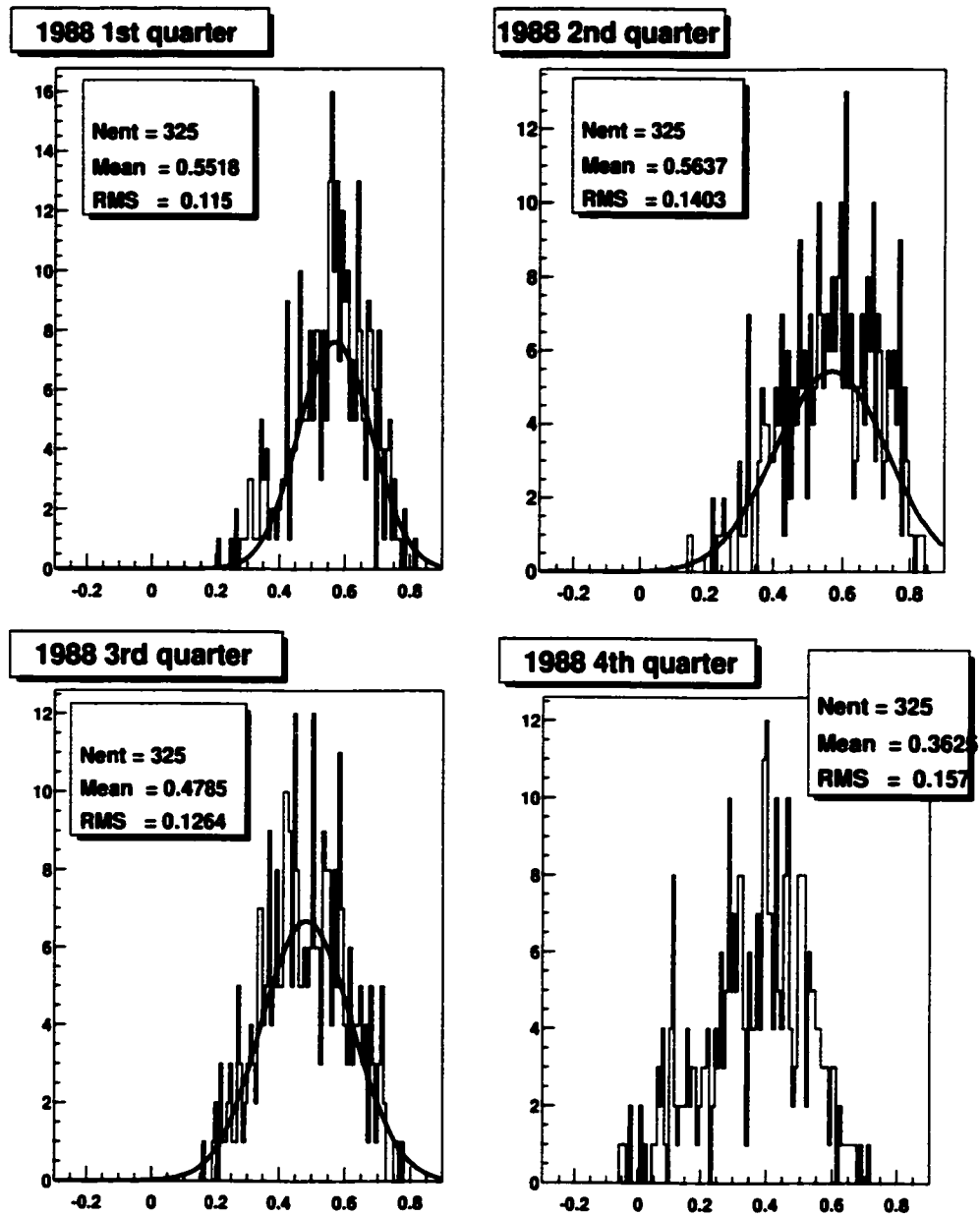


Figure B.3: Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1988.

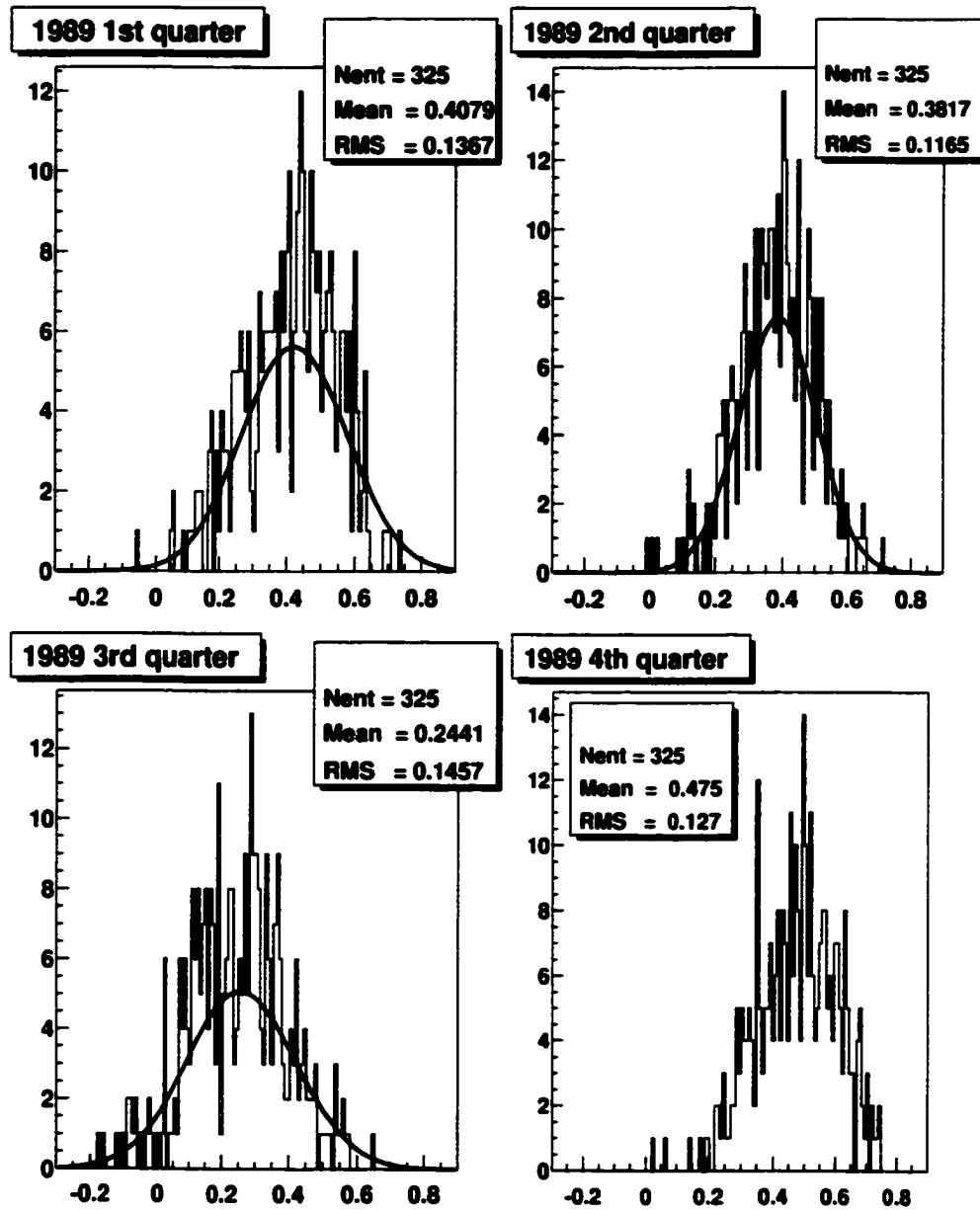


Figure B.4: Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1989.

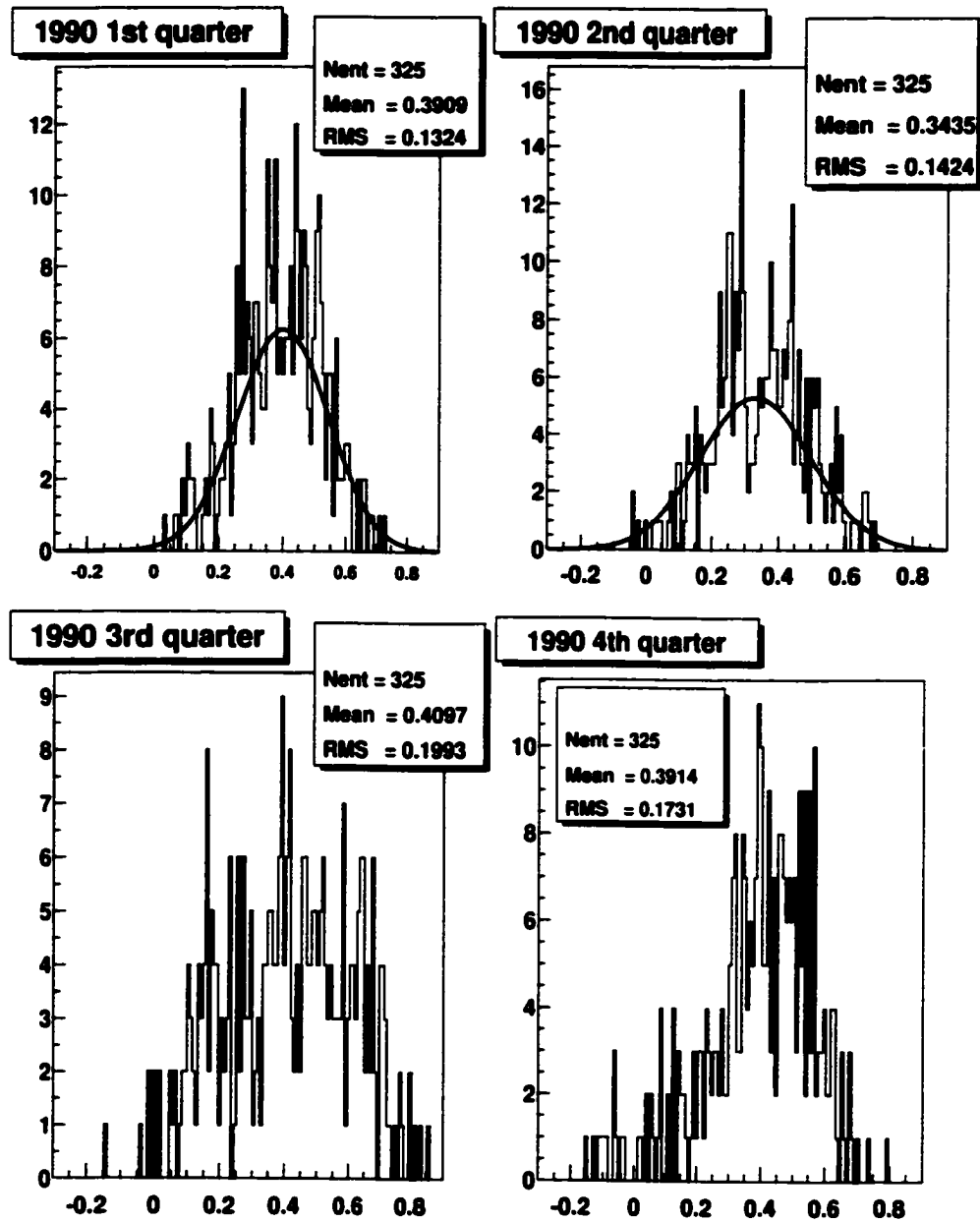


Figure B.5: Histograms of the quarterly correlation coefficients between 26 of DJIA components, highlighted in Table A.1, for the year 1990.

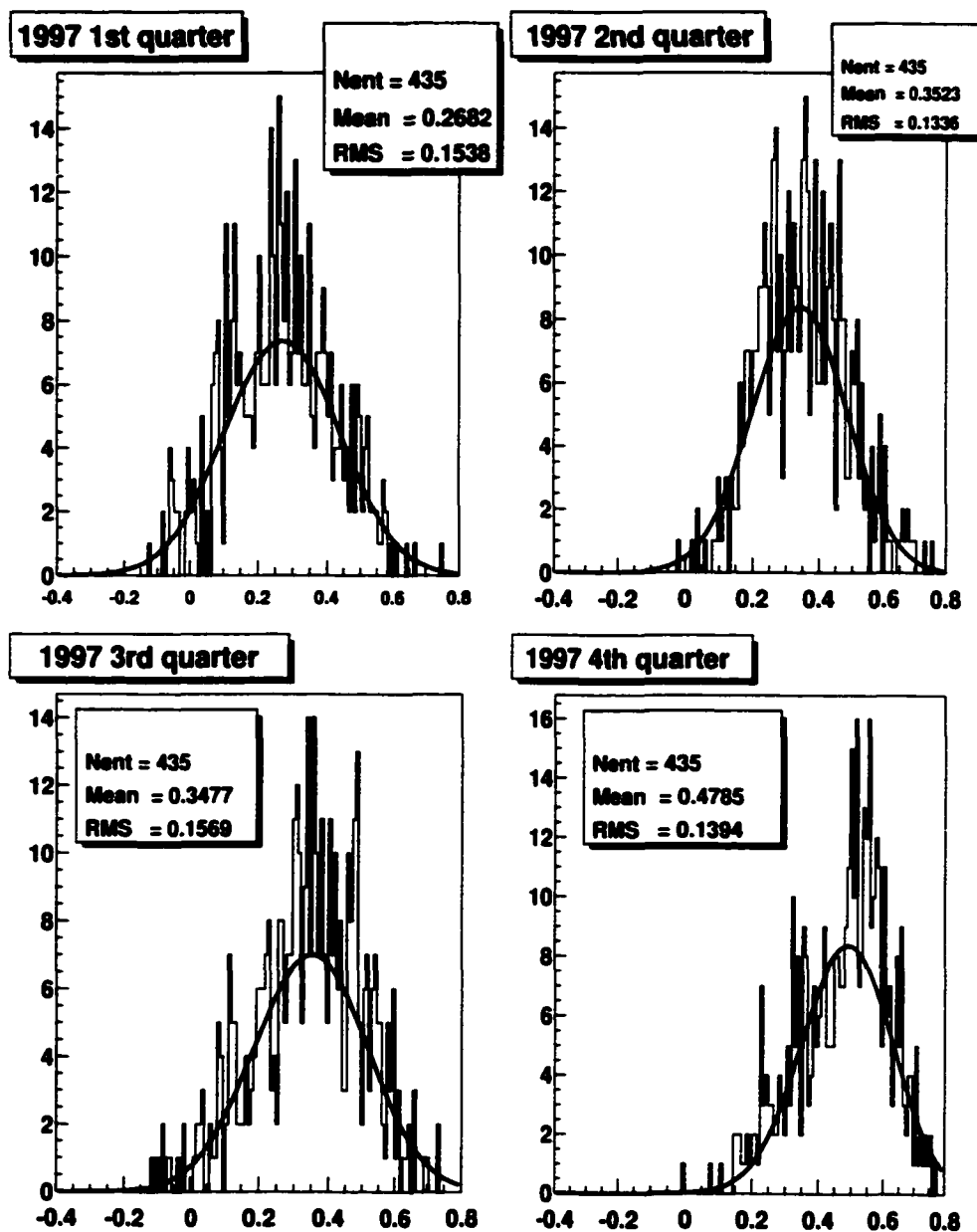


Figure B.6: Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 1997.

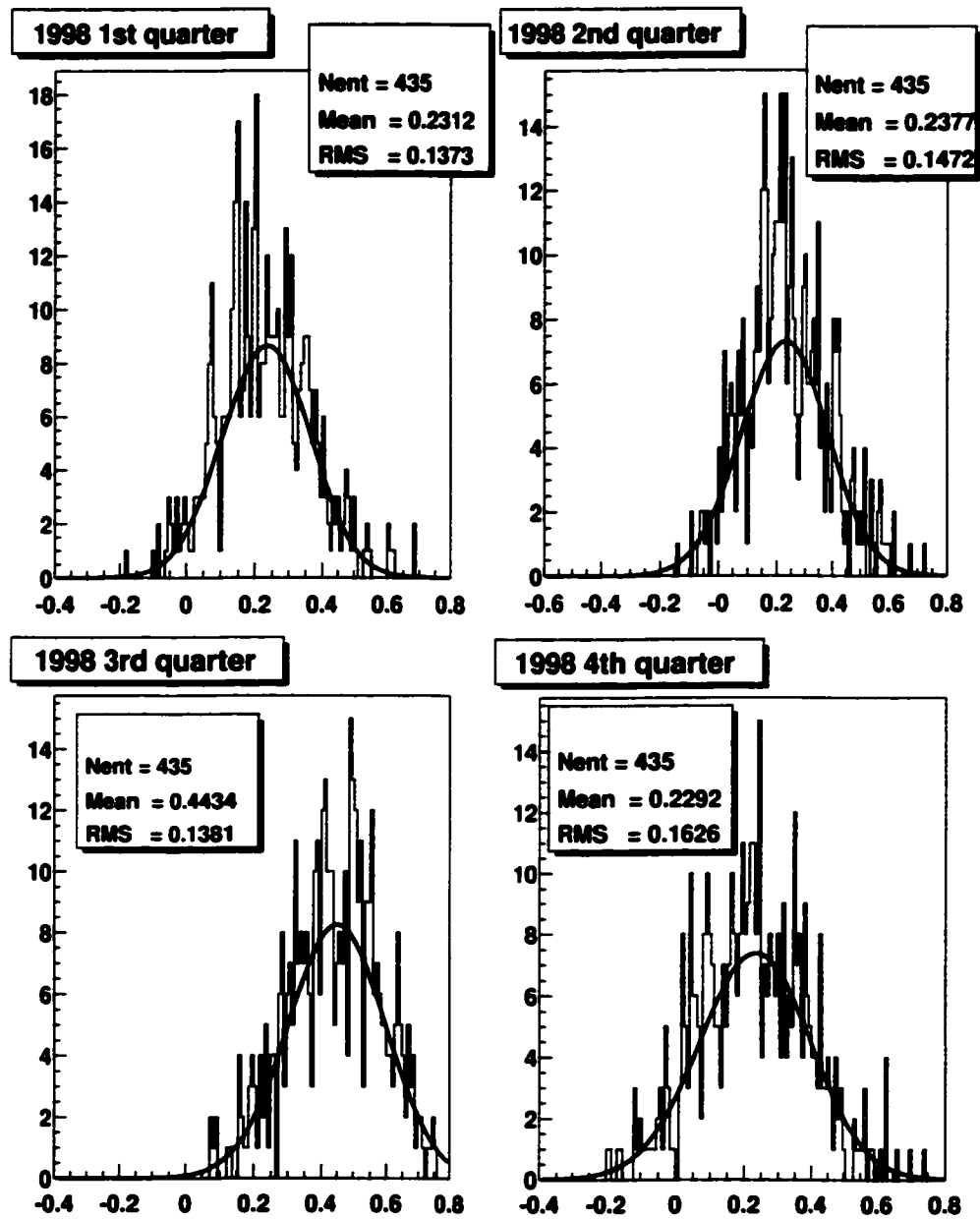


Figure B.7: Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 1998.

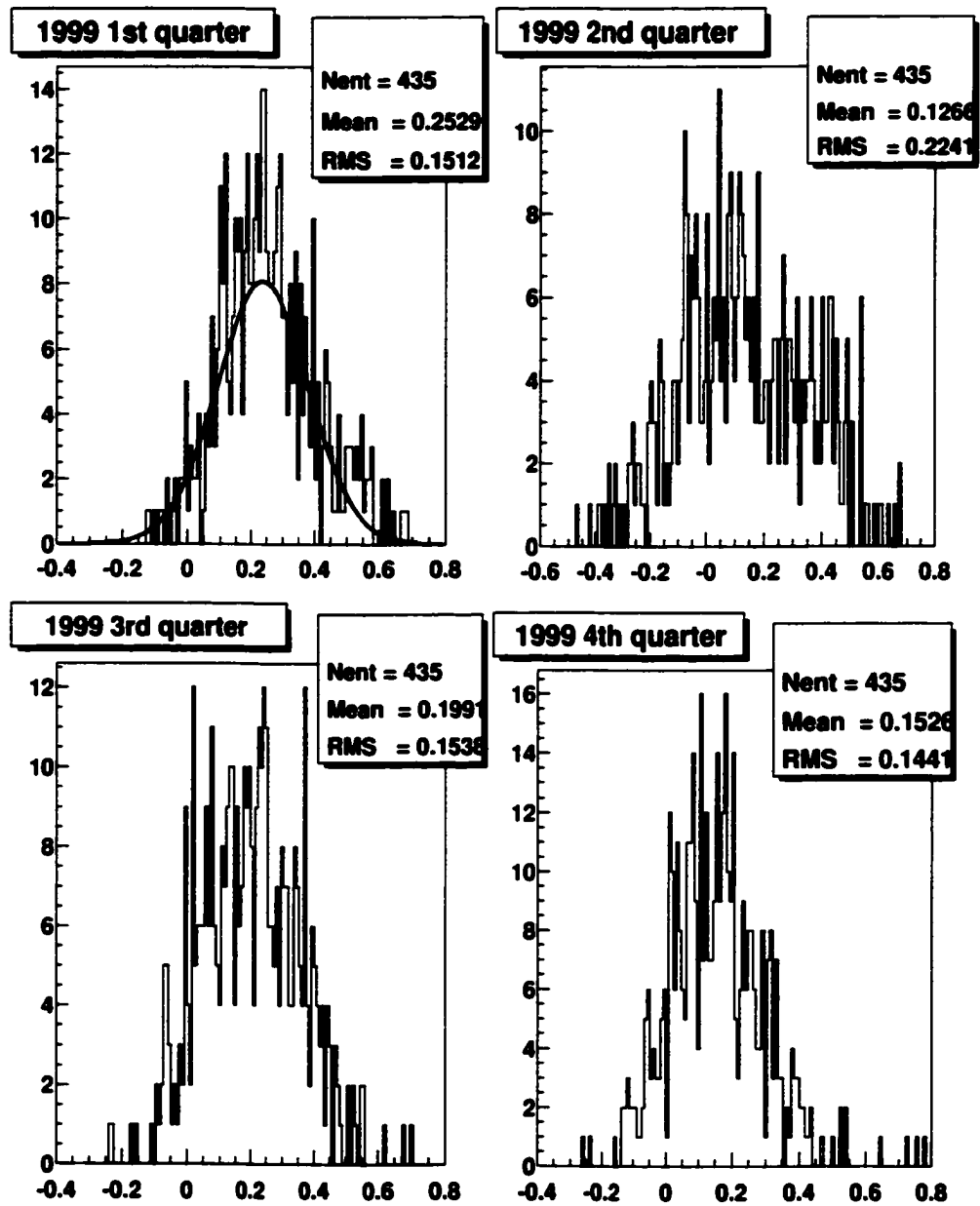


Figure B.8: Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 1999.

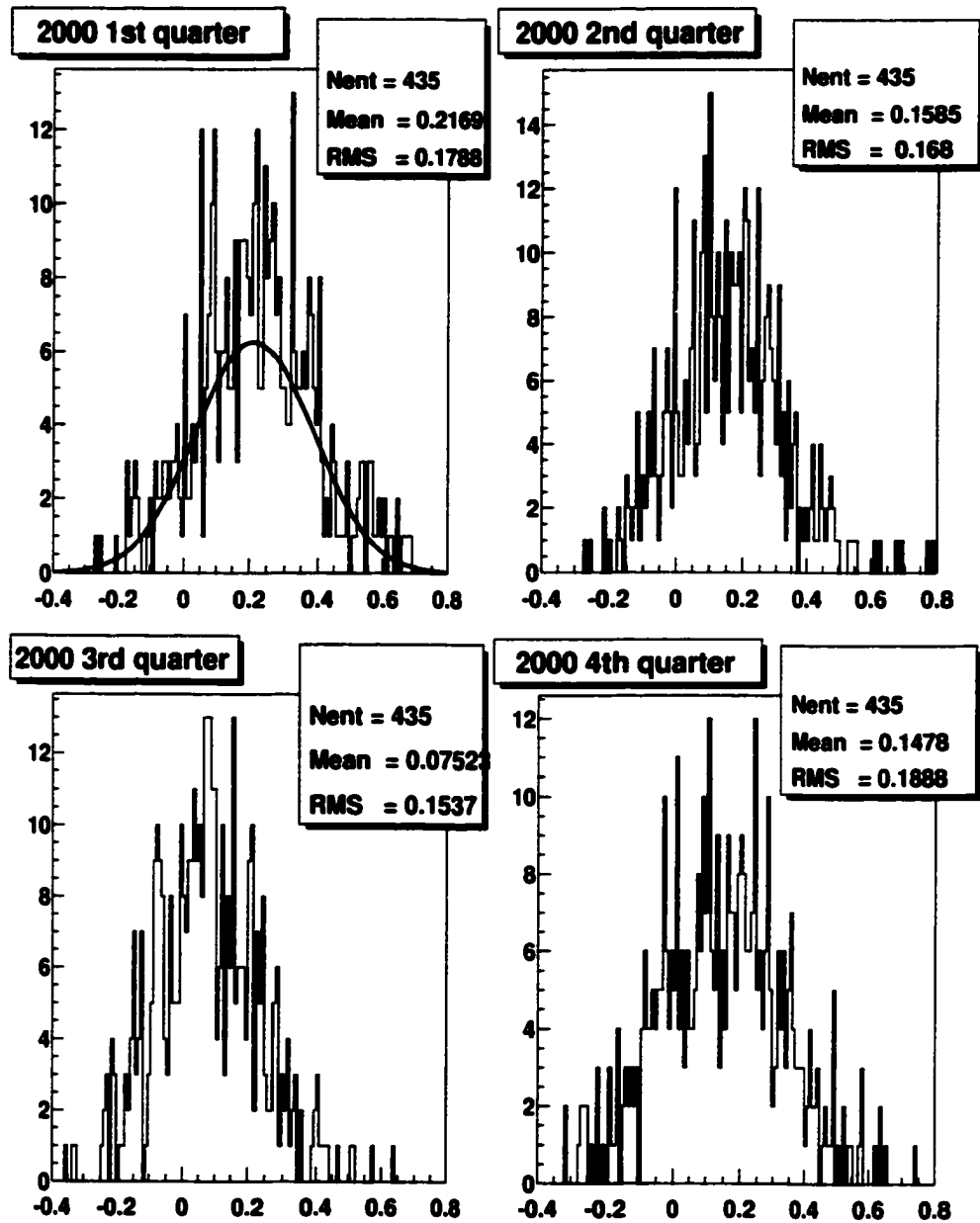


Figure B.9: Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 2000.

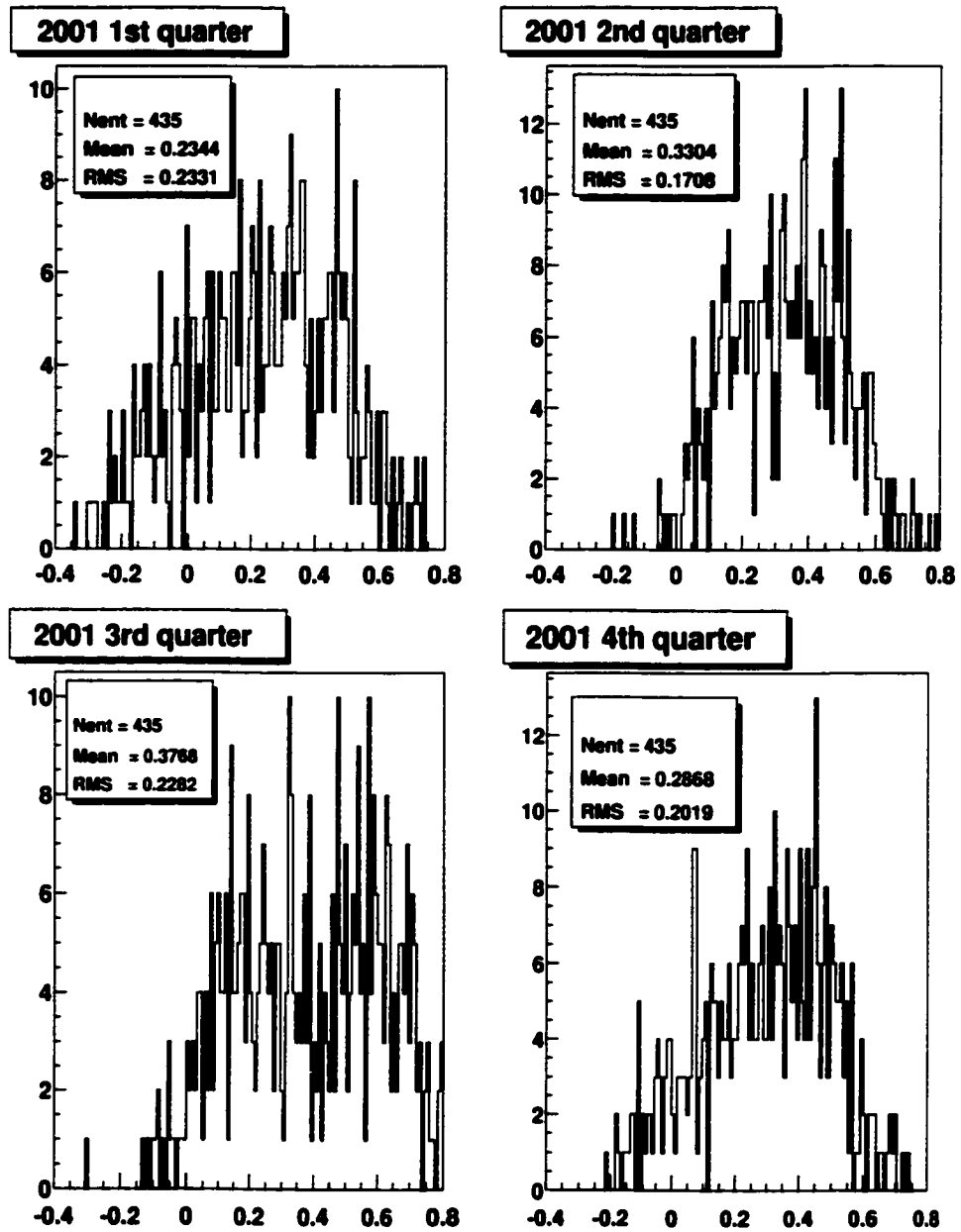


Figure B.10: Histograms of the quarterly correlation coefficients between the 30 DJIA components, listed in Table A.2, for the year 2001.

Appendix C

Eigenvalue Spectra of Quarterly Correlation Matrices

The appendix displays the histograms of the eigenvalues for the quarterly correlation matrices between the studied assets. For consistency, all of the histograms have the range of the represented eigenvalues between -1 and 20 with 100 bins.

There are 26 eigenvalues for the 26×26 correlation matrices between the assets studied during the period 1986-1990. The number of eigenvalues for the second analyzed interval, 1997-2001, is 30 since the correlation matrices have, in this case, the size 30×30 .

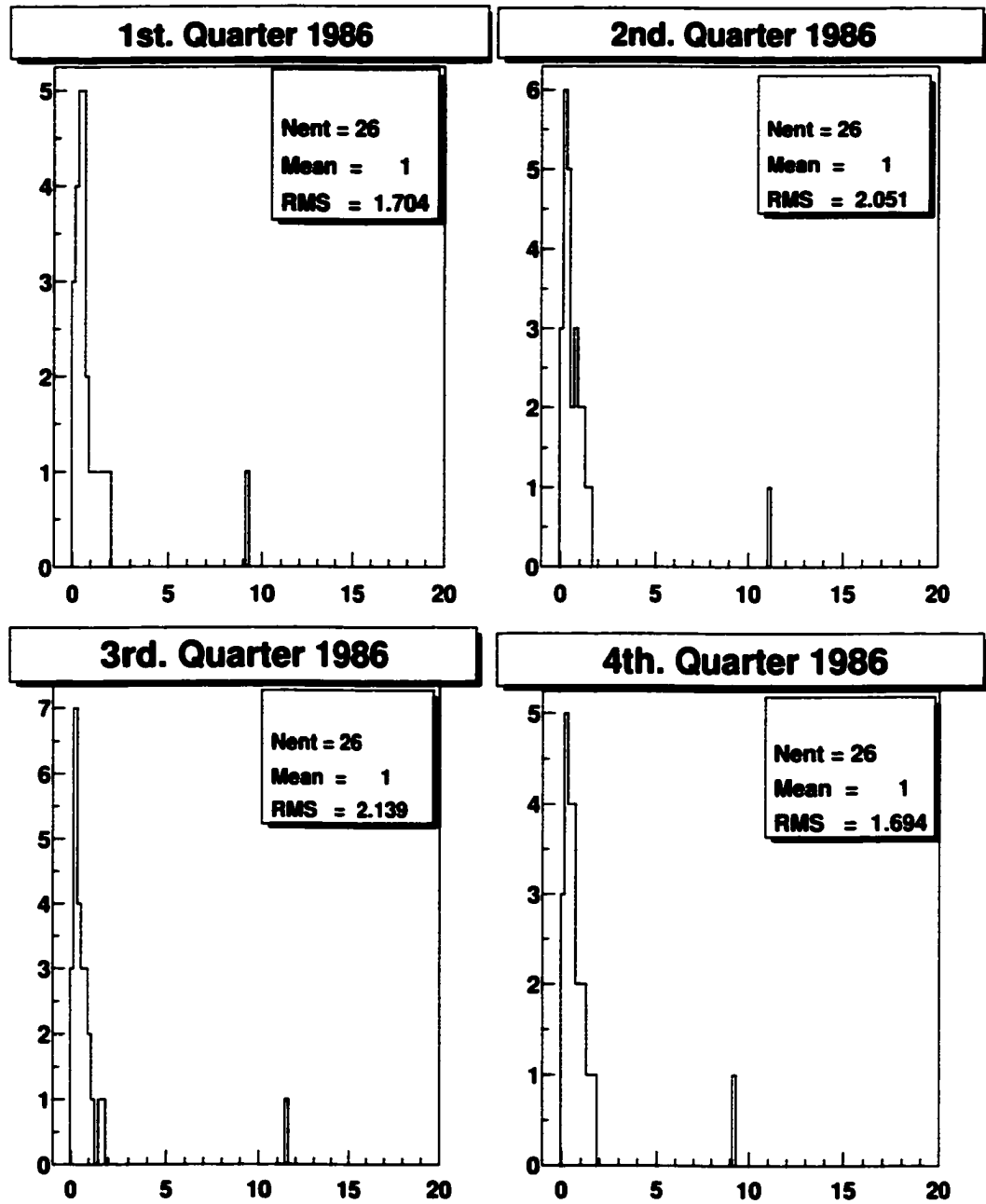


Figure C.1: Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1986.

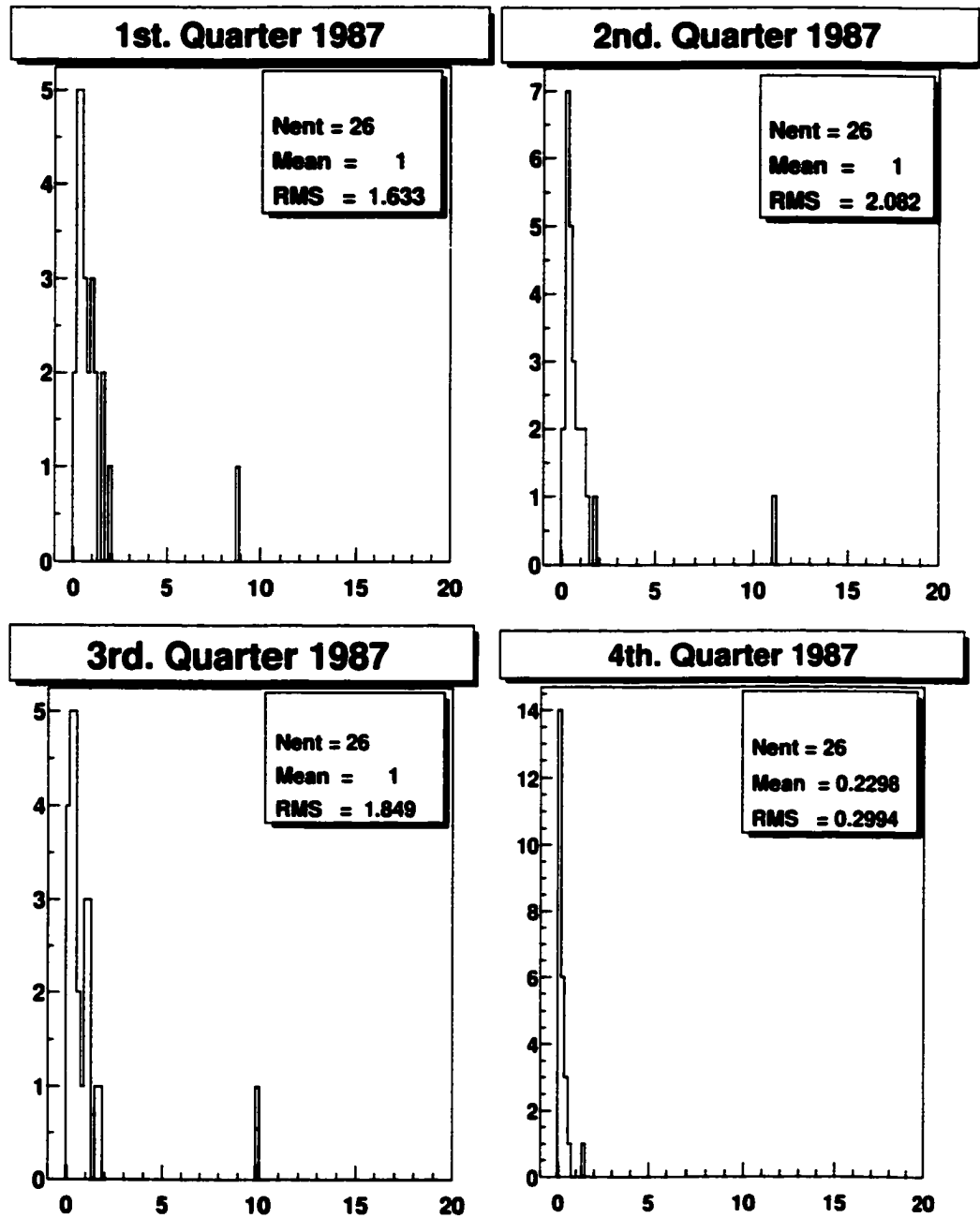


Figure C.2: Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1987.

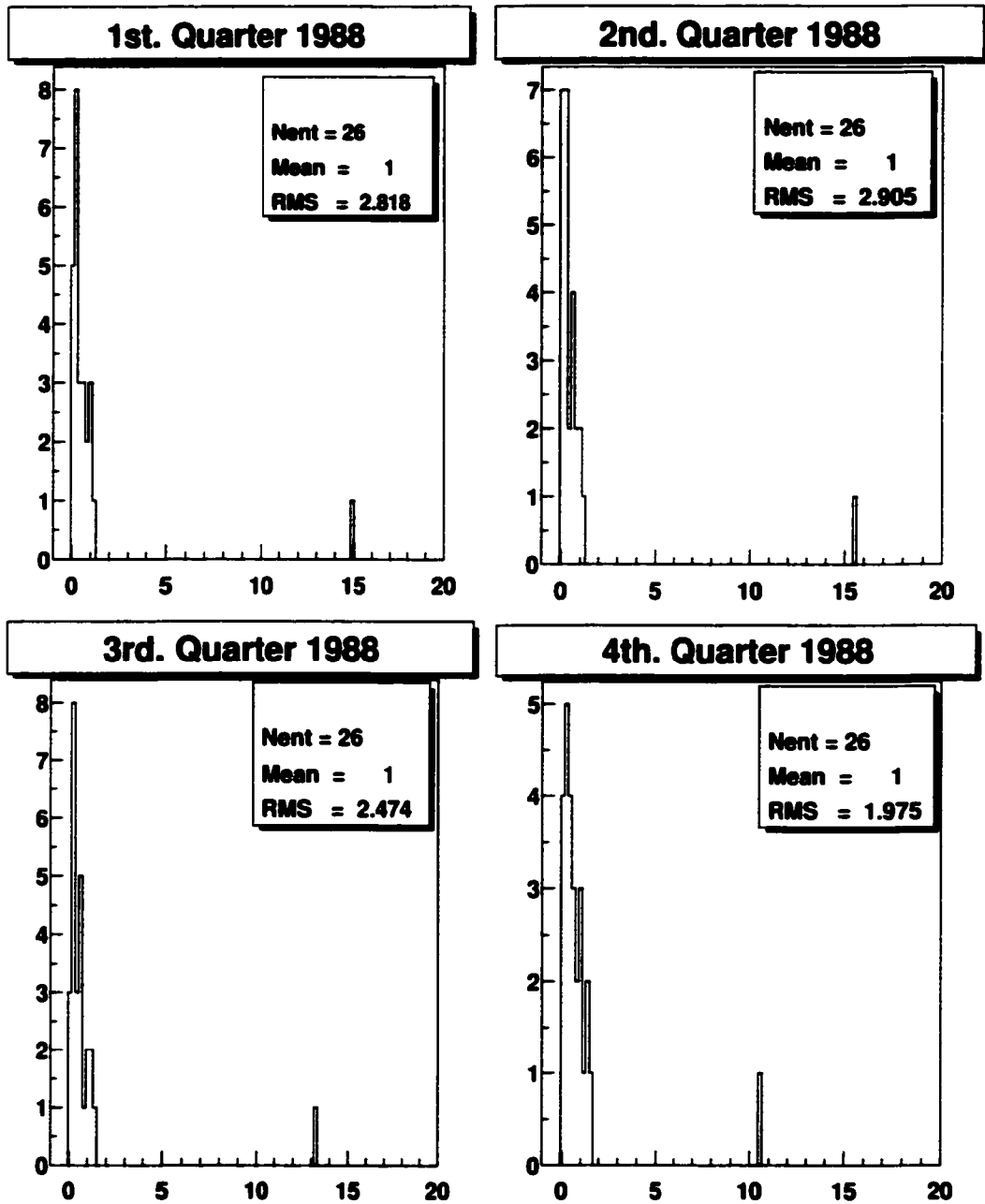


Figure C.3: Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1988.

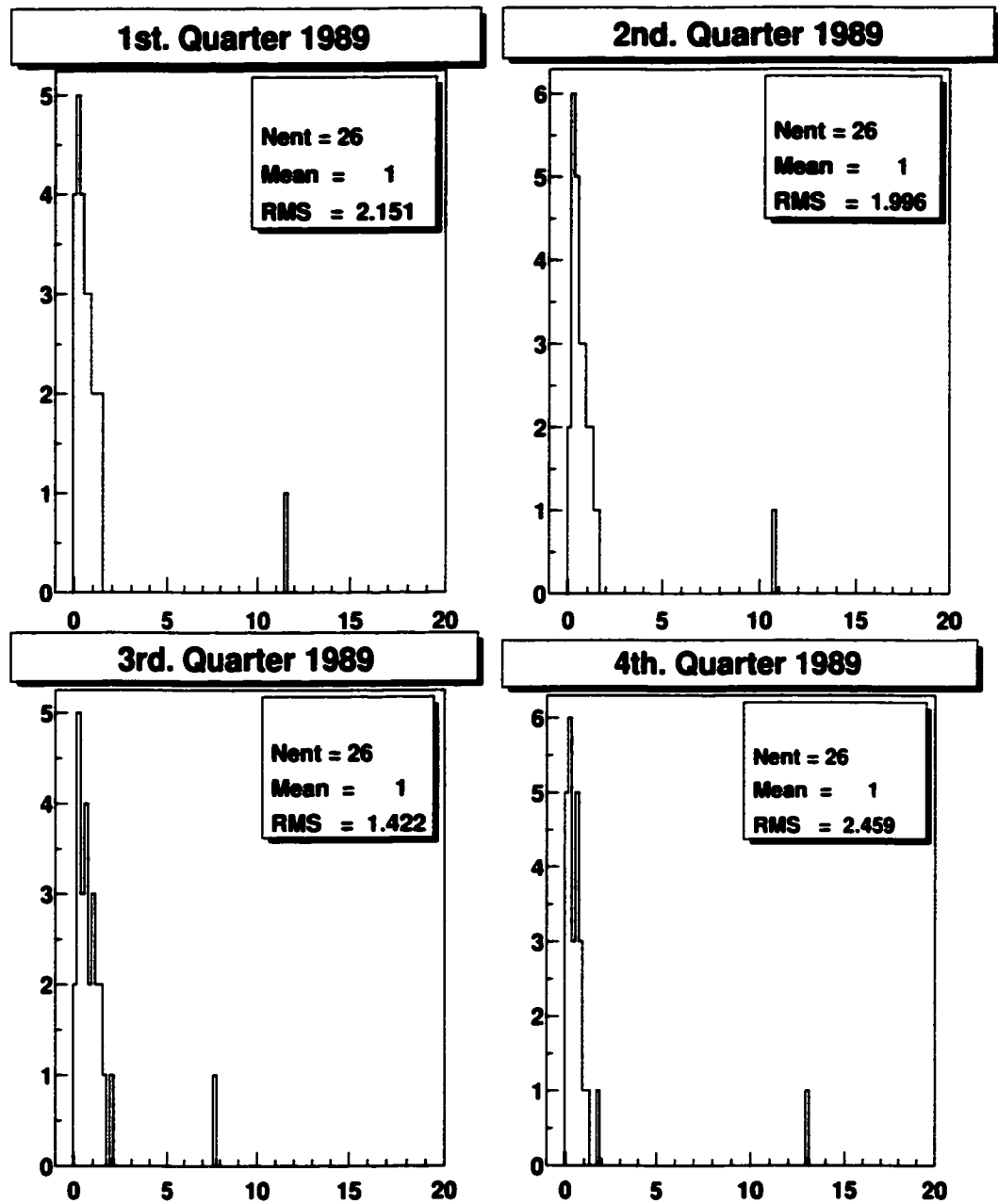


Figure C.4: Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1989.

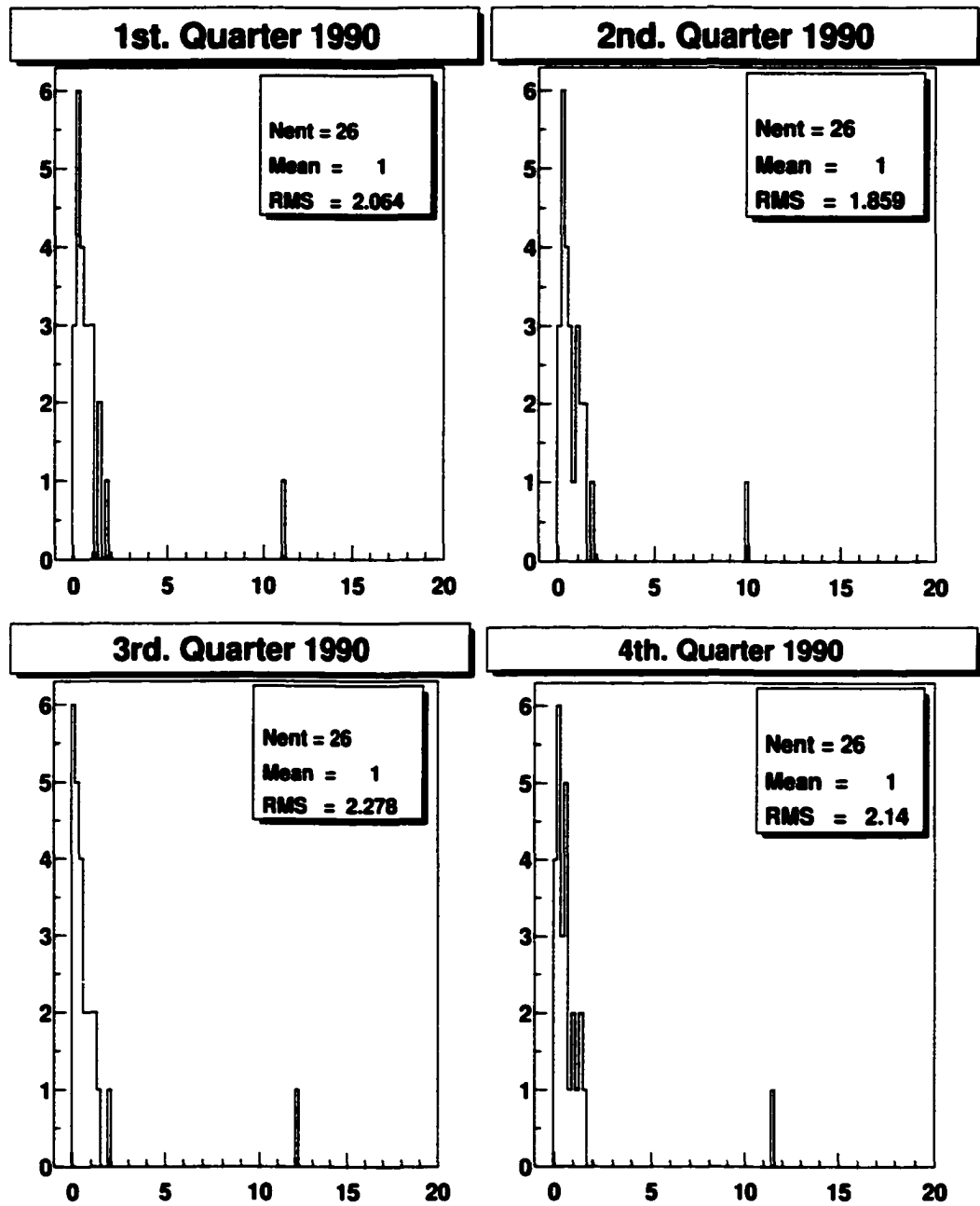


Figure C.5: Eigenvalue spectra of the quarterly correlation matrices for the 26 major US companies during the year 1990.

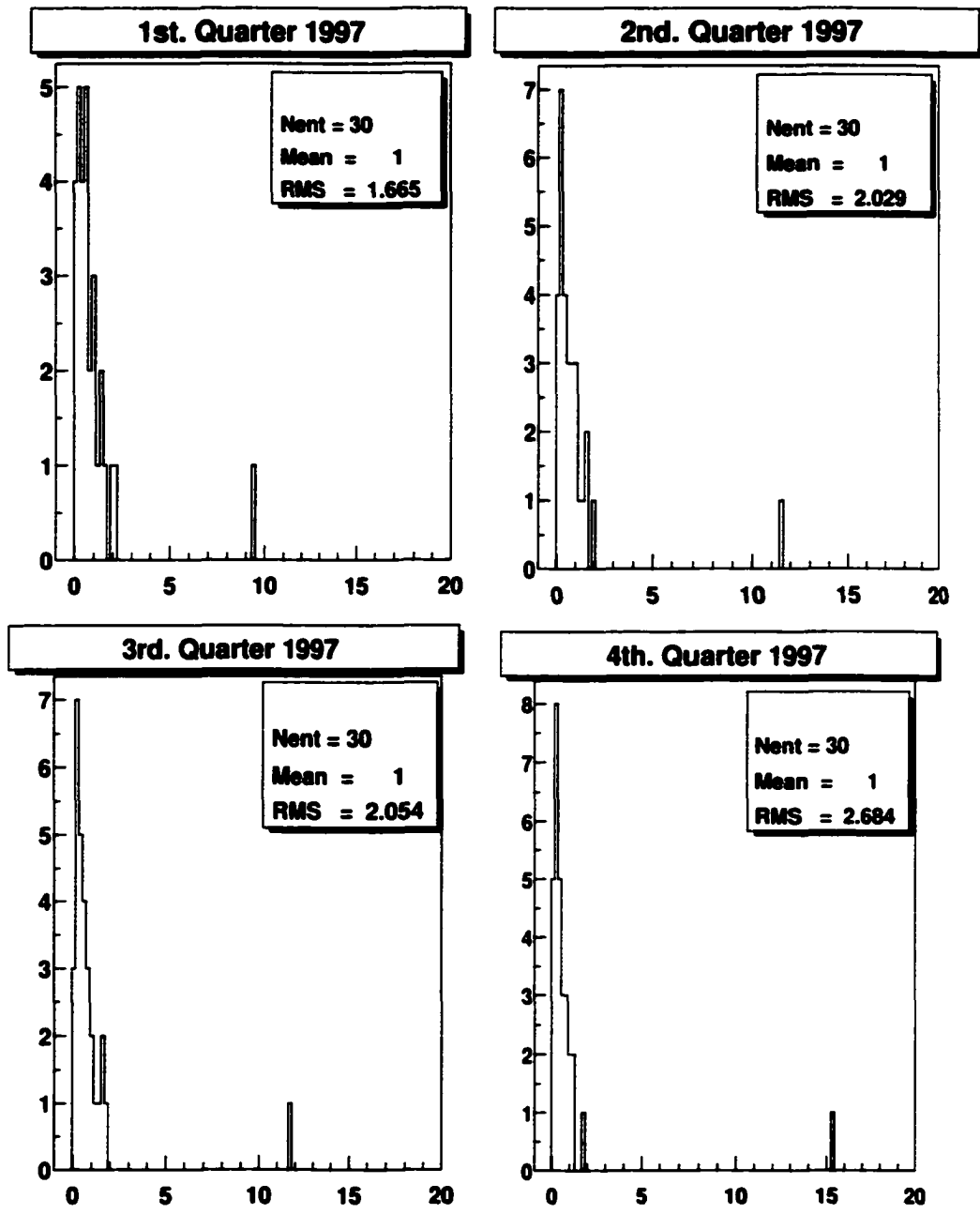


Figure C.6: Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 1997.

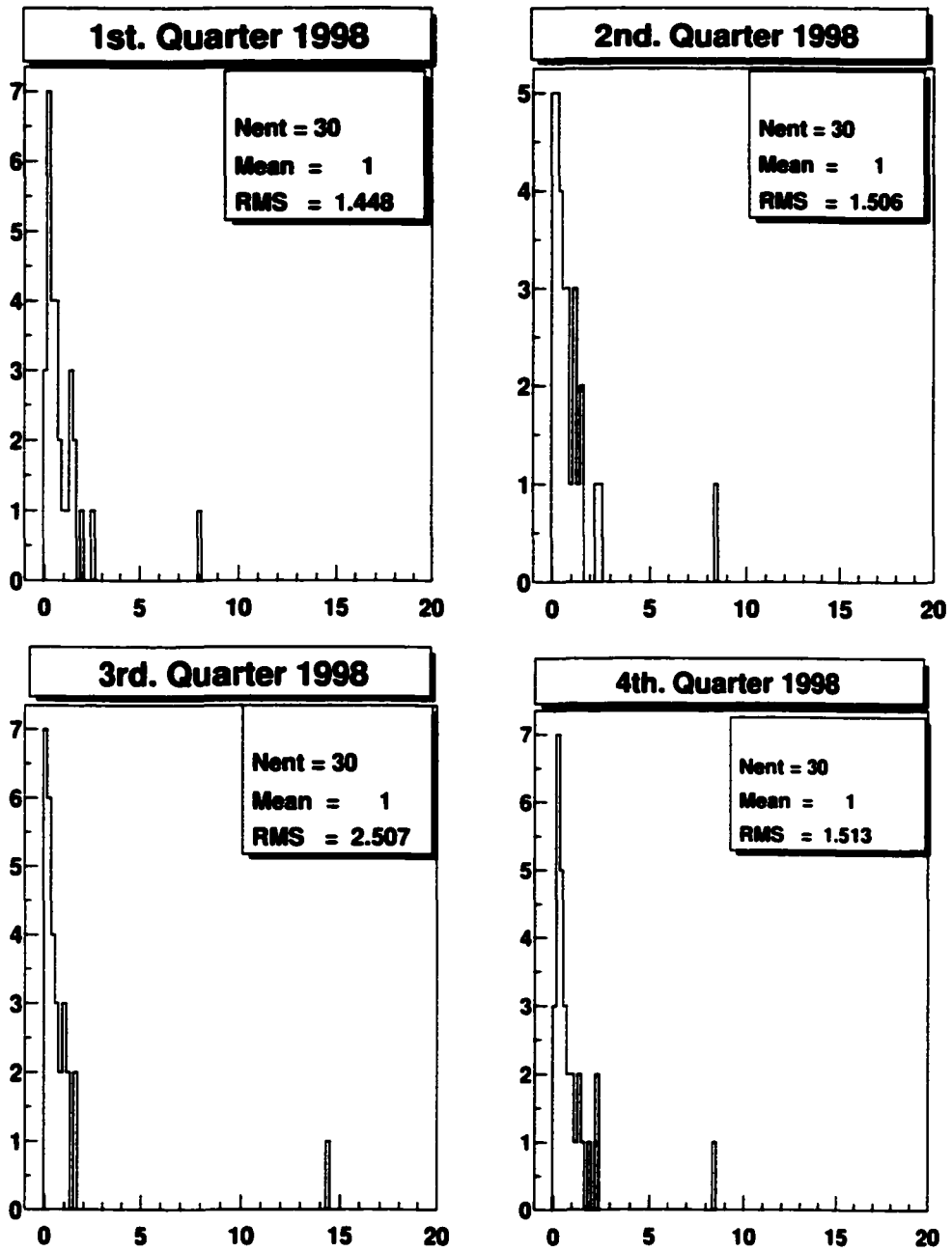


Figure C.7: Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 1998.

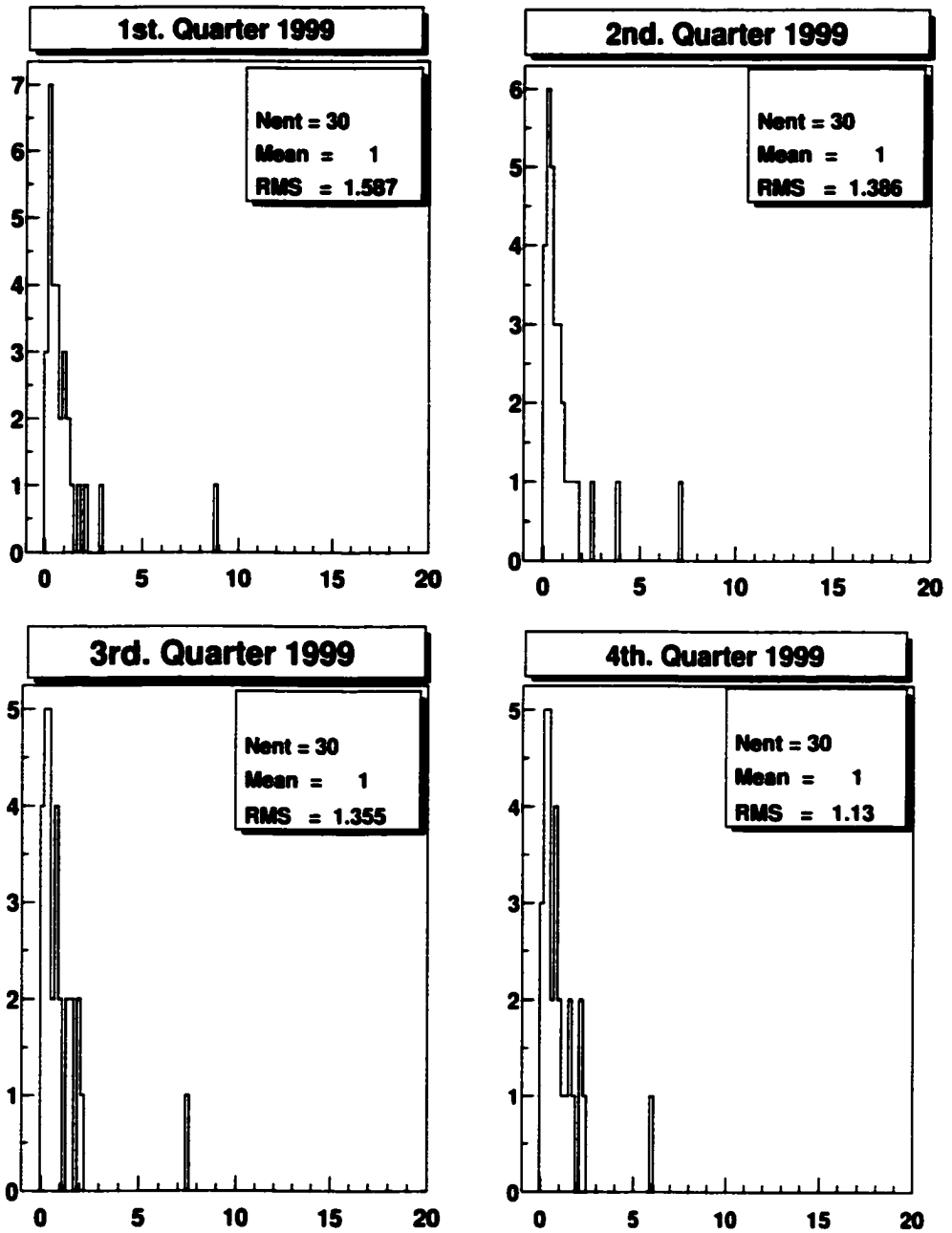


Figure C.8: Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 1999.

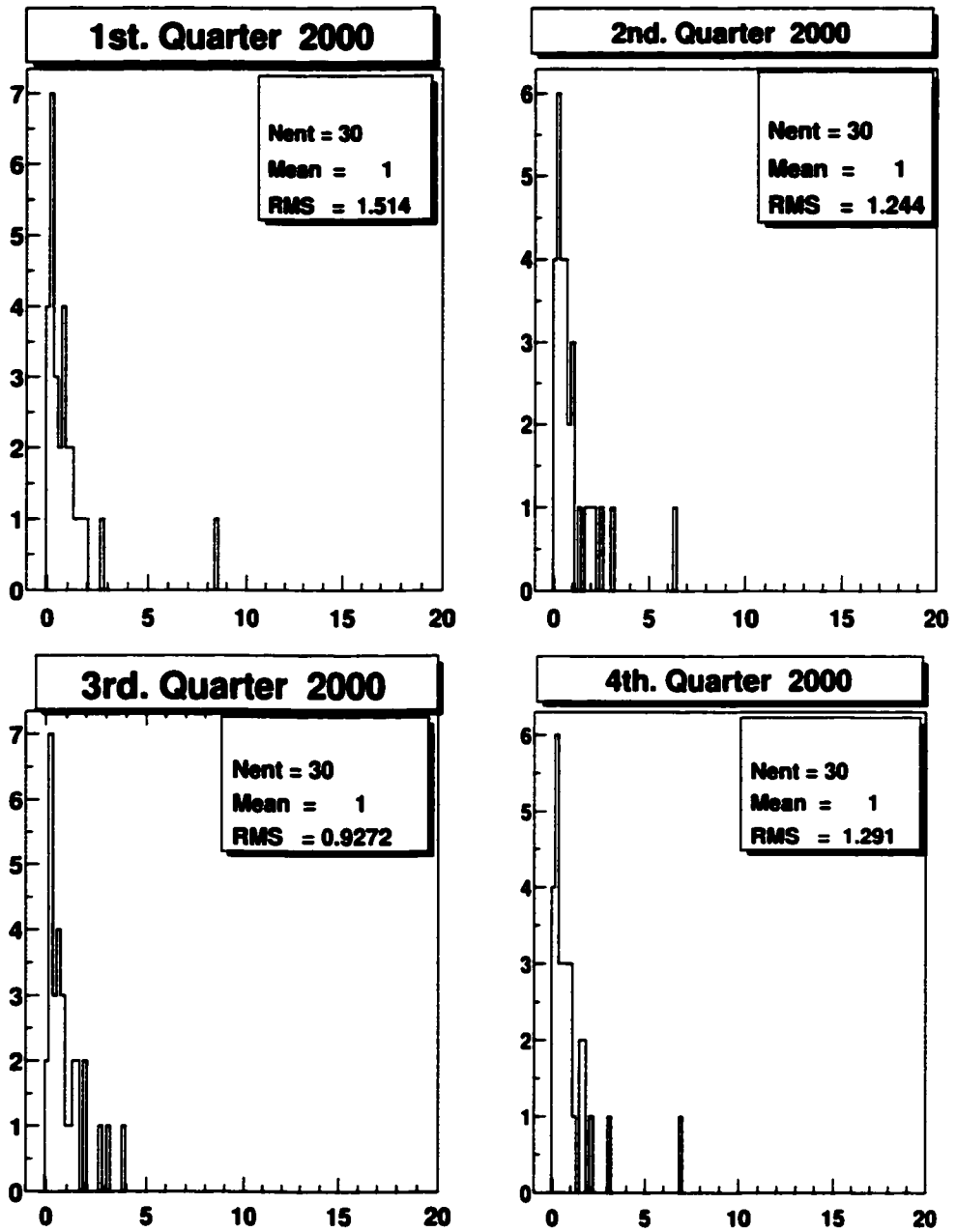


Figure C.9: Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 2000.

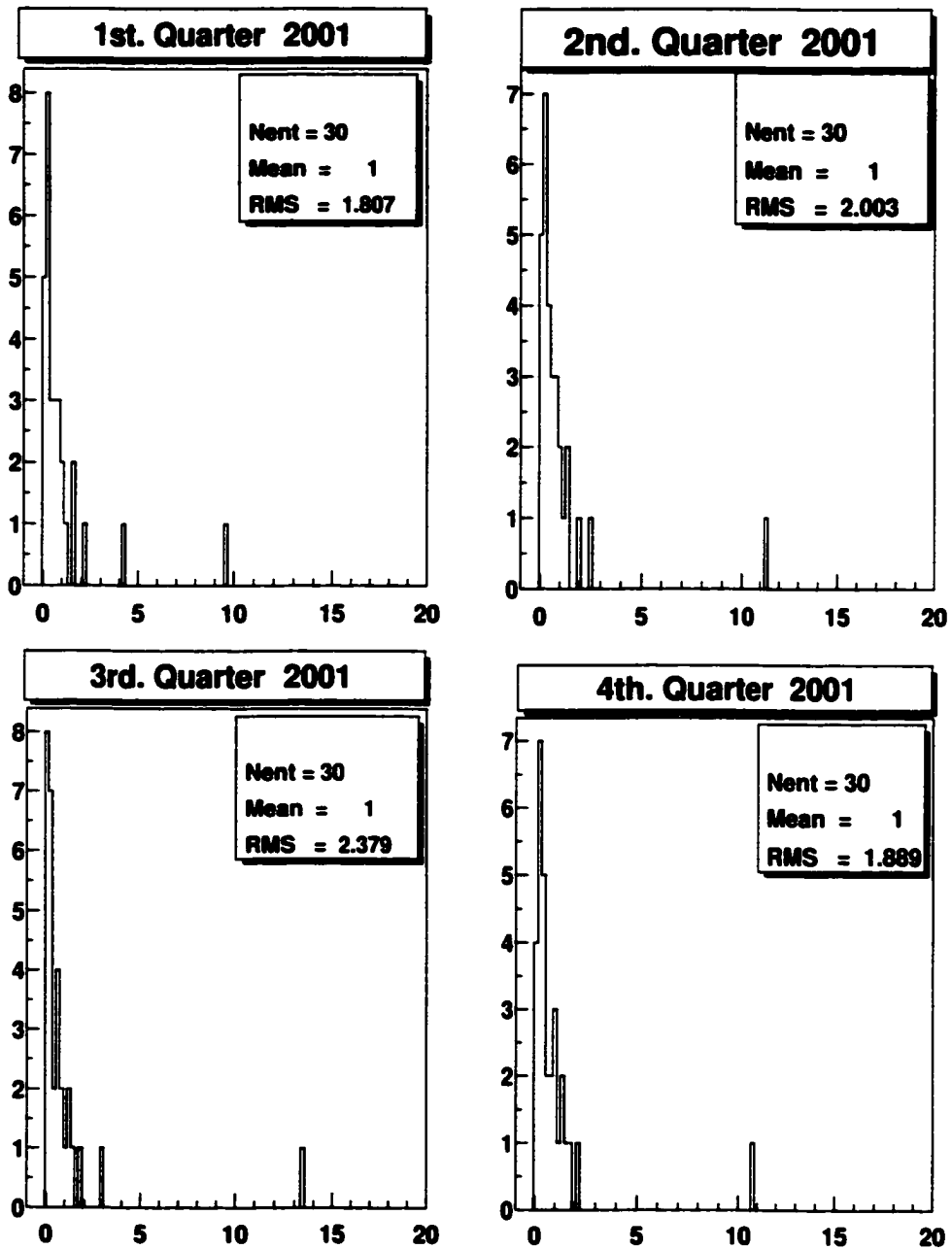


Figure C.10: Eigenvalue spectra of the quarterly correlation matrices for the 30 DJIA components during the year 2001.

Bibliography

- [1] J. Hertz, A. Krogh and R. G. Palmer, 1991 *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA
- [2] R. O. Duda and P. E. Hart, 1973, *Pattern Classification and Scene Analysis*, Wiley, New York.
- [3] J. MacQueen. 1967, Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, L. M. Le Cam and J. Neyman (eds.), University of California Press, Berkeley, 281-297.
- [4] J. M. Pena, J. A. Lozano and P. Larranaga, 1999. An Empirical Comparison of Four Initialization Methods for the K-means Algorithm. *Patt. Recogn. Letts.* **20**, 1027-1040.
- [5] G. Karypis, E. H. Han and V. Kumar, 1999, Chameleon: Hierarchical Clustering Using Dynamic Modeling, *Computer*, Vol. 32, No. 8, pp 68-75.
- [6] B. Yu and B. Yuan, 1995, A Global Optimum Clustering Algorithm, *Engng. Applic. Artif. Intell.* **8**, 223-227.

- [7] K. Rose, 1998, Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems, *Proc. IEEE* **86**, 2210-2239.
- [8] S. Kirkpatrick, C. D. Gellatt and M. P. Vecchi, 1983, Simulated annealing, *Science*. **220**, 671-680.
- [9] D. E. Goldberg, 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- [10] M. Mitchell, 1996, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA.
- [11] K. Krishna and M. Narasimha Murty, 1999, Genetic K-means Algorithm, *IEEE Trans. Syst. Man. Cyb. B* **29**, 433-439.
- [12] M. Blatt, S. Wiseman and E. Domany, 1996, Superparamagnetic Clustering of Data, *Phys. Rev. Lett.* **76**, 3251-3254.
- [13] S. Wiseman, M. Blatt and E. Domany, 1998, Superparamagnetic Clustering of Data, *Phys. Rev. E* **57**, 3767-3783.
- [14] E. Domany, M. Blatt, Y. Gdalyahu and D. Weinshall, 1999, Superparamagnetic Clustering of Data: Application to Computer Vision, *Comp. Phys. Comm.* **121-122**, 5-12.
- [15] E. Domany, 1999, Superparamagnetic Clustering of Data - The Definitive Solution of an Ill-posed Problem, *Physica A* **263**, 158-169.

- [16] E. Domany, M. Blatt and S. Wiseman, 2000, Method and Apparatus for Clustering Data, *United States Patent* 6,021,383, Feb. 1, 2000.
- [17] F. Y. Wu, 1982, The Potts Model, *Rev. Mod. Phys.* **54**, 235-268.
- [18] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, 1953, Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21**, 1087-1092.
- [19] L. Angelini, F. De Carlo, C. Marangi, M. Pellicoro and S. Stramaglia. 2000, Clustering Data by Inhomogeneous Chaotic Map Lattices. *Phys. Rev. Lett.* **85**, 554-557.
- [20] I. H. Witten and E. Frank, 2000, *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco.
- [21] C. L. Blake and C. J. Merz, 1998, *UCI Repository of Machine Learning Databases*, University of California, Department of Information and Computer Sciences, Irvine, CA. [<http://www.ics.uci.edu/mlearn/MLRepository.html>].
- [22] H. H. Harman, 1976, *Modern Factor Analysis*, University of Chicago Press, Chicago, IL.
- [23] R. Šášik, T. Hwa, N. Iranfar and W. F. Loomis, 2001, Percolation Clustering: A Novel Algorithm Applied to the Clustering of Gene Expression

Patterns in *Dictyostelium* Development, *Pacific Symposium on Biocomputing*, **6** 335-347.

- [24] P. J. Flory, 1941, Molecular Size Distribution in Three-Dimensional Polymers: I, Gelation, *J. Amer. Chem. Soc.* **63**, 3083-3100.
- [25] W. H. Stockmayer, 1943, Theory of Molecular Size Distribution and Gel Formation in Branched Chain Polymers, *J. Chem. Phys.*, **11**, 45-55.
- [26] S. R. Broadbent and J. M. Hammersley, 1957, Percolation Processes I. Crystals and Mazes, *Proc. Cambr. Phil. Soc.* **53**, 629-641.
- [27] J. M. Hammersley, 1982, Origins of Percolation Theory, Percolation Structures and Processes, *Annals of Israel Physical Society*, Vol. 5. G. Deutscher, R. Zallen and J. Adler.
- [28] D. Stauffer and A. Aharony, 1991, *Introduction to Percolation Theory*, 2nd Ed., Taylor and Francis Ltd., London.
- [29] H. O. Peitgen, H. Jürgens and D. Saupe, 1992, *Chaos and Fractals*, Springer-Verlag, NY.
- [30] H. E. Stanley, 1971, *Phase Transitions and Critical Phenomena*. Oxford University Press, Ely House, London.
- [31] D. Baker, G. Paul, S. Sreenivasan and H. E. Stanley, 2002, The Continuum Percolation Threshold for Interpenetrating Squares and Cubes, cond-mat/0203235.

- [32] A. Bunde and S. Havlin, eds., 1996, *Fractal and Disordered Systems, 2nd Ed.*, Springer-Verlag, Berlin.
- [33] E. J. Garboczi, K. A. Snyder, J. F. Douglas and M. F. Thorpe, 1995, Geometrical Threshold of Overlapping Ellipsoids *Phys. Rev. E*, **52**, 819-828.
- [34] N. J. Giordano, 1997, *Computational Physics*, Prentice Hall, Upper Saddle River, NJ.
- [35] L. R. Nyhoff. 1999, *An Introduction To Data Structures*, Prentice-Hall, Upper Saddle River, NJ.
- [36] J. W. J. Williams, 1964, Algorithm 232: Heapsort, *Communications of the Association of Computing Machinery*, **7**:347-348.
- [37] D. B. West, 1996. *Introduction to Graph Theory*, Prentice-Hall. Englewood Cliffs, NJ.
- [38] R. Rammal, G. Toulouse and M. A. Virasoro, 1986, Ultrametricity for Physicists, *Rev. Mod. Phys.* **58**, 765-788.
- [39] L. L. Baldwin and L. T. Wille, 2001, Ask Not What Data Mining Can Do For Materials Science, *Proceedings of the Third International Conference on Intelligent Processing and Manufacturing of Materials*, eds. J. A. Meech, S. M. Veiga, M. M. Veiga, S. R. LeClair and J. F. Maguire.

- [40] S. R. LeClair, 2000, Computational “materials-process” design and control. Toward the future - virtual materials research. *Engng. Applic. Artif. Intell.* **13**, 495-496.
- [41] A. K. Jain and R. C. Dubes, 1988, *Algorithms for Clustering Data*. Prentice Hall, Upper Saddle River, NJ.
- [42] D. A. Porter and K. E. Easterling, 1992, *Phase Transformations in Metals and Alloys*, Chapman and Hall, London.
- [43] A. L. Barabasi and H. E. Stanley, 1995, *Fractal Concepts in Surface Growth*, Cambridge University Press, Cambridge.
- [44] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, 1992, *Numerical Recipes in C, 2nd Ed.*, Cambridge University Press. New York.
- [45] K. Binder, 1988, *Monte Carlo Simulation in Statistical Physics: An Introduction*, Springer, New York.
- [46] D. P. Landau and R. Alben, 1973, Monte Carlo Calculations as an Aid in Teaching Statistical Mechanics, *Am. J. Phys.* **41**, 394-400.
- [47] R. N. Mantegna and H. E. Stanley, 2000, *An Introduction to Econophysics*, Cambridge University Press, Cambridge.
- [48] H. M. Markowitz, 1991, *Portfolio Selection: Efficient Diversification of Investments*, Blackwell, Cambridge, MA.

- [49] S. J. Brown, 1989, The Number of Factors in Security Returns, *J. Finance* **44**, 1247-1262.
- [50] R. N. Mantegna, 1999, Hierarchical Structure in Financial Markets. *Eur. Phys. J. B.* **11**, 193-197.
- [51] G. Bonanno, N. Vandewalle and R. Mantegna, 2000. Taxonomy of Stock Market Indices, *Phys. Rev. E* **62**, 7615-7618.
- [52] L. Kullmann, J. Kertész and R. N. Mantegna, 2000. Identification of Clusters of Companies in Stock Indices via Pott Super-Paramagnetic Transitions, cond-mat/0002238.
- [53] J. F. Bouchaud and M. Potters, 2000, *Theory of Financial Risk*, Cambridge University Press. New York.
- [54] L. Laloux, P. Cizeau, J. P. Bouchaud and M. Potters, 1999, Noise Dressing of Financial Correlation Matrices, *Phys. Rev. Letters* **83**, 1467-1470.
- [55] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral and H. E. Stanley, 1999, Universal and Nonuniversal Properties of Cross Correlations in Financial Time Series, *Phys. Rev. Letters* **83**, 1471-1474.
- [56] F. Lillo and R. N. Mantegna, 2001, Variety of Behavior of Equity Returns in Financial Markets, No. 156 in *Computing in Economics and Finance 2001* from Society for Computational Economics.

- [57] G. Bonanno, G. Caldarelli, F. Lillo and R. N. Mantegna, 2002, Topology of Correlation Based Minimal Spanning Trees in Real and Model Markets, *cond-mat/0211546*.
- [58] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, T. Guhr and H. E. Stanley, 2002, Random matrix approach to cross correlations in financial data, *Phys. Rev. E* **65**, 066126-1–066126-18.
- [59] Y. Malevergne and D. Sornette, 2002, Collective Origin of the Coexistence of Apparent RMT Noise and Factors in Large Sample Correlation Matrices, *cond-mat/0210115*.