Computer Science and Data Analysis Series

Clustering for Data Mining

A Data Recovery Approach

Boris Mirkin

Chapman & Hall/CRC
Taylor & Francis Group
Boca Raton London New York Singapore

Published in 2005 by
Chapman & Hall/CRC
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-10: 1-58488-534-3 (Hardcover)
International Standard Book Number-13: 978-1-58488-534-4 (Hardcover)
Library of Congress Card Number 2005041421

Library of Congress Cataloging-in-Publication Data

Mirkin, B. G. (Boris Grigorévich)
Clustering for data mining : a data recovery approach / Boris Mirkin.
p. cm. -- (Computer science and data analysis series ; 3)
Includes bibliographical references and index.
ISBN 1-58488-534-3
1. Data mining. 2. Cluster analysis. I. Title. II. Series.

QA76.9.D343M57 2005
006.3'12--dc22 2005041421

T&F informa
Taylor & Francis Group
is the Academic Division of T&F Informa plc.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Chapman & Hall/CRC

Computer Science and Data Analysis Series

The interface between the computer and statistical sciences is increasing, as each discipline seeks to harness the power and resources of the other. This series aims to foster the integration between the computer sciences and statistical, numerical, and probabilistic methods by publishing a broad range of reference works, textbooks, and handbooks.

Proposals for the series should be sent directly to one of the series editors above, or submitted to:

Chapman & Hall/CRC
23-25 Blades Court
London SW15 2NU
UK

Published Titles

Bayesian Artificial Intelligence
Kevin B. Korb and Ann E. Nicholson

Pattern Recognition Algorithms for Data Mining
Sankar K. Pal and Pabitra Mitra

Exploratory Data Analysis with MATLAB®
Wendy L. Martinez and Angel R. Martinez

Clustering for Data Mining: A Data Recovery Approach
Boris Mirkin

Correspondence Analysis and Data Coding with JAVA and R
Fionn Murtagh

R Graphics
Paul Murrell

Contents

Preface

Clustering is a discipline devoted to finding and describing cohesive or homogeneous chunks in data, the clusters.

Some exemplary clustering problems are:

- Finding common surf patterns in the set of web users;

- Automatically revealing meaningful parts in a digitalized image;

- Partition of a set of documents in groups by similarity of their contents;

- Visual display of the environmental similarity between regions on a country map;

- Monitoring socio-economic development of a system of settlements via a small number of representative settlements;

- Finding protein sequences in a database that are homologous to a query protein sequence;

- Finding anomalous patterns of gene expression data for diagnostic purposes;

- Producing a decision rule for separating potentially bad-debt credit applicants;

- Given a set of preferred vacation places, finding out what features of the places and vacationers attract each other;

- Classifying households according to their furniture purchasing patterns and finding groups' key characteristics to optimize furniture marketing and production.

Clustering is a key area in data mining and knowledge discovery, which are activities oriented towards finding non-trivial or hidden patterns in data collected in databases.

Earlier developments of clustering techniques have been associated, primarily, with three areas of research: factor analysis in psychology [55], numerical taxonomy in biology [122], and unsupervised learning in pattern recognition [21].

Technically speaking, the idea behind clustering is rather simple: introduce a measure of similarity between entities under consideration and combine similar entities into the same clusters while keeping dissimilar entities in different clusters. However, implementing this idea is less than straightforward.

First, too many similarity measures and clustering techniques have been invented with virtually no support to a non-specialist user in selecting among them. The trouble with this is that different similarity measures and/or clustering techniques may, and frequently do, lead to different results. Moreover, the same technique may also lead to different cluster solutions depending on the choice of parameters such as the initial setting or the number of clusters specified. On the other hand, some common data types, such as questionnaires with both quantitative and categorical features, have been left virtually without any substantiated similarity measure.

Second, use and interpretation of cluster structures may become an issue, especially when available data features are not straightforwardly related to the phenomenon under consideration. For instance, certain data on customers available at a bank, such as age and gender, typically are not very helpful in deciding whether to grant a customer a loan or not.

Specialists acknowledge peculiarities of the discipline of clustering. They understand that the clusters to be found in data may very well depend not on only the data but also on the user's goals and degree of granulation. They frequently consider clustering as art rather than science. Indeed, clustering has been dominated by learning from examples rather than theory based instructions. This is especially visible in texts written for inexperienced readers, such as [4], [28] and [115].

The general opinion among specialists is that clustering is a tool to be applied at the very beginning of investigation into the nature of a phenomenon under consideration, to view the data structure and then decide upon applying better suited methodologies. Another opinion of specialists is that methods for finding clusters as such should constitute the core of the discipline; related questions of data pre-processing, such as feature quantization and standardization, definition and computation of similarity, and post-processing, such as interpretation and association with other aspects of the phenomenon, should be left beyond the scope of the discipline because they are motivated by external considerations related to the substance of the phenomenon under investigation. I share the former opinion and argue the latter because it is at odds with the former: in the very first steps of knowledge discovery, substantive considerations are quite shaky, and it is unrealistic to expect that they alone could lead to properly solving the issues of pre- and post-processing.

Such a dissimilar opinion has led me to believe that the discovered clusters must be treated as an "ideal" representation of the data that could be used for recovering the original data back from the ideal format. This is the idea of the data recovery approach: not only use data for finding clusters but also use clusters for recovering the data. In a general situation, the data recovered from aggregate clusters cannot fit the original data exactly, which can be used for evaluation of the quality of clusters: the better the fit, the better the clusters. This perspective would also lead to the addressing of issues in pre- and post-processing, which now becomes possible because parts of the data that are explained by clusters can be separated from those that are not.

The data recovery approach is common in more traditional data mining and statistics areas such as regression, analysis of variance and factor analysis, where it works, to a great extent, due to the Pythagorean decomposition of the data scatter into "explained" and "unexplained" parts. Why not try the same approach in clustering?

In this book, two of the most popular clustering techniques, K-Means for partitioning and Ward's method for hierarchical clustering, are presented in the framework of the data recovery approach. The selection is by no means random: these two methods are well suited because they are based on statistical thinking related to and inspired by the data recovery approach, they minimize the overall within cluster variance of data. This seems to be the reason of the popularity of these methods. However, the traditional focus of research on computational and experimental aspects rather than theoretical ones has contributed to the lack of understanding of clustering methods in general and these two in particular. For instance, no firm relation between these two methods has been established so far, in spite of the fact that they share the same square error criterion.

I have found such a relation, in the format of a Pythagorean decomposition of the data scatter into parts explained and unexplained by the found cluster structure. It follows from the decomposition, quite unexpectedly, that it is the divisive clustering format, rather than the traditional agglomerative format, that better suits the Ward clustering criterion. The decomposition has led to a number of other observations that amount to a theoretical framework for the two methods. Moreover, the framework appears to be well suited for extensions of the methods to different data types such as mixed scale data including continuous, nominal and binary features. In addition, a bunch of both conventional and original interpretation aids have been derived for both partitioning and hierarchical clustering based on contributions of features and categories to clusters and splits. One more strain of clustering techniques, one-by-one clustering which is becoming increasingly popular, naturally emerges within the framework giving rise to intelligent versions of K-Means, mitigating the need for user-defined setting of the number of clusters and their hypothetical prototypes. Most importantly, the framework leads to a set of mathematically proven properties relating classical clustering with other clustering techniques such as conceptual clustering and graph theoretic clustering as well as with other data mining concepts such as decision trees and association in contingency data tables.

These are all presented in this book, which is oriented towards a reader interested in the technical aspects of data mining, be they a theoretician or a practitioner. The book is especially well suited for those who want to learn WHAT clustering is by learning not only HOW the techniques are applied but also WHY. In this way the reader receives knowledge which should allow him not only to apply the methods but also adapt, extend and modify them according to the reader's own ends.

This material is organized in five chapters presenting a unified theory along with computational, interpretational and practical issues of real-world data mining with clustering:

- What is clustering (Chapter 1);
- What is data (Chapter 2);
- What is K-Means (Chapter 3);
- What is Ward clustering (Chapter 4);
- What is the data recovery approach (Chapter 5).

But this is not the end of the story. Two more chapters follow. Chapter 6 presents some other clustering goals and methods such as SOM (self-organizing maps) and EM (expectation-maximization), as well as those for conceptual description of clusters. Chapter 7 takes on "big issues" of data mining: validity and reliability of clusters, missing data, options for data pre-processing and standardization, etc. When convenient, we indicate solutions to the issues following from the theory of the previous chapters. The Conclusion reviews the main points brought up by the data recovery approach to clustering and indicates potential for further developments.

This structure is intended, first, to introduce classical clustering methods and their extensions to modern tasks, according to the data recovery approach, without learning the theory (Chapters 1 through 4), then to describe the theory leading to these and related methods (Chapter 5) and, in addition, see a wider picture in which the theory is but a small part (Chapters 6 and 7).

In fact, my prime intention was to write a text on classical clustering, updated to issues of current interest in data mining such as processing mixed feature scales, incomplete clustering and conceptual interpretation. But then I realized that no such text can appear before the theory is described. When I started describing the theory, I found that there are holes in it, such as a lack of understanding of the relation between K-Means and the Ward method and in fact a lack of a theory for the Ward method at all, misconceptions in quantization of qualitative categories, and a lack of model based interpretation aids. This is how the current version has become a threefold creature oriented toward:

1. Giving an account of the data recovery approach to encompass partitioning, hierarchical and one-by-one clustering methods;

2. Presenting a coherent theory in clustering that addresses such issues as (a) relation between normalizing scales for categorical data and measuring association between categories and clustering, (b) contributions of various elements of cluster structures to data scatter and their use in interpreta-

tion, (c) relevant criteria and methods for clustering differently expressed data, etc.;

3. Providing a text in data mining for teaching and self-learning popular data mining techniques, especially K-Means partitioning and Ward agglomerative and divisive clustering, with emphases on mixed data pre-processing and interpretation aids in practical applications.

At present, there are two types of literature on clustering, one leaning towards providing general knowledge and the other giving more instruction. Books of the former type are Gordon [39] targeting readers with a degree of mathematical background and Everitt et al. [28] that does not require mathematical background. These include a great deal of methods and specific examples but leave rigorous data mining instruction beyond the prime contents. Publications of the latter type are Kaufman and Rousseeuw [62] and chapters in data mining books such as Dunham [23]. They contain selections of some techniques reported in an ad hoc manner, without any concern on relations between them, and provide detailed instruction on algorithms and their parameters.

This book combines features of both approaches. However, it does so in a rather distinct way. The book does contain a number of algorithms with detailed instructions and examples for their settings. But selection of methods is based on their fitting to the data recovery theory rather than just popularity. This leads to the covering of issues in pre- and post-processing matters that are usually left beyond instruction. The book does contain a general knowledge review, but it concerns more of issues rather than specific methods. In doing so, I had to clearly distinguish between four different perspectives: (a) statistics, (b) machine learning, (c) data mining, and (d) knowledge discovery, as those leading to different answers to the same questions. This text obviously pertains to the data mining and knowledge discovery perspectives, though the other two are also referred to, especially with regard to cluster validation.

The book assumes that the reader may have no mathematical background beyond high school: all necessary concepts are defined within the text. However, it does contain some technical stuff needed for shaping and explaining a technical theory. Thus it might be of help if the reader is acquainted with basic notions of calculus, statistics, matrix algebra, graph theory and logics.

To help the reader, the book conventionally includes a list of denotations, in the beginning, and a bibliography and index, in the end. Each individual chapter is preceded by a boxed set of goals and a dictionary of base words. Summarizing overviews are supplied to Chapters 3 through 7. Described methods are accompanied with numbered computational examples showing the working of the methods on relevant data sets from those presented in Chapter 1; there are 58 examples altogether. Computations have been carried out with self-made programs for MATLAB®, the technical computing tool developed by The MathWorks (see its Internet web site www.mathworks.com).

The material has been used in the teaching of data clustering and visualization to MSc CS students in several colleges across Europe. Based on these experiences, different teaching options can be suggested depending on the course objectives, time resources, and students' background.

If the main objective is teaching clustering methods and there are very few hours available, then it would be advisable to first pick up the material on generic K-Means in sections 3.1.1 and 3.1.2, and then review a couple of related methods such as PAM in section 6.1.2, iK-Means in 3.3.1, Ward agglomeration in 4.1 and division in 4.2.1, single linkage in 6.2.1 and SOM in 6.1.6. Given a little more time, a review of cluster validation techniques from 7.6 including examples in 3.3.2 should follow the methods. In a more relaxed regime, issues of interpretation should be brought forward as described in 3.4, 4.2.3, 6.3 and 7.2.

If the main objective is teaching data visualization, then the starting point should be the system of categories described in 1.1.5, followed by material related to these categories: bivariate analysis in section 2.2, regression in 5.1.2, principal component analysis (SVM decomposition) in 5.1.3, K-Means and iK-Means in Chapter 3, Self-organizing maps SOM in 6.1.6 and graph-theoretic structures in 6.2.

Acknowledgments

Too many people contributed to the approach and this book to list them all. However, I would like to mention those researchers whose support was important for channeling my research efforts: Dr. E. Braverman, Dr. V. Vapnik, Prof. Y. Gavrilets, and Prof. S. Aivazian, in Russia; Prof. F. Roberts, Prof. F. McMorris, Prof. P. Arabie, Prof. T. Krauze, and Prof. D. Fisher, in the USA; Prof. E. Diday, Prof. L. Lebart and Prof. B. Burtschy, in France; Prof. H.-H. Bock, Dr. M. Vingron, and Dr. S. Suhai, in Germany. The structure and contents of this book have been influenced by comments of Dr. I. Muchnik (Rutgers University, NJ, USA), Prof. M. Levin (Higher School of Economics, Moscow, Russia), Dr. S. Nascimento (University Nova, Lisbon, Portugal), and Prof. F. Murtagh (Royal Holloway, University of London, UK).

Author

Boris Mirkin is a Professor of Computer Science at the University of London UK. He develops methods for data mining in such areas as social surveys, bioinformatics and text analysis, and teaches computational intelligence and data visualization.

Dr. Mirkin first became known for his work on combinatorial models and methods for data analysis and their application in biological and social sciences. He has published monographs such as “Group Choice” (John Wiley & Sons, 1979) and “Graphs and Genes” (Springer-Verlag, 1984, with S. Rodin). Subsequently, Dr. Mirkin spent almost ten years doing research in scientific centers such as Ecole Nationale Suprieure des Tlcommunications (Paris, France), Deutsches Krebs Forschnung Zentrum (Heidelberg, Germany), and Center for Discrete Mathematics and Theoretical Computer Science DIMACS, Rutgers University (Piscataway, NJ, USA). Building on these experiences, he developed a unified framework for clustering as a data recovery discipline.

List of Denotations

I	Entity set
N	Number of entities
V	Feature set
V_l	Set of categories of a categorical feature l
M	Number of column features
$X = (x_{iv})$	Raw entity-to-feature data table
$Y = (y_{iv})$	Standardized entity-to-feature data table; $y_{iv} = (x_{iv} - a_v)/b_v$ where a_v and b_v denote the shift and scale coefficients, respectively
$y_i = (y_{iv})$	M-dimensional vector corresponding to entity $i \in I$ according to data table Y
$y_i = (y_{iv})$	M-dimensional vector corresponding to entity $i \in I$ according to data table Y
(x, y)	Inner product of two vector points $x = (x_j)$ and $y = (y_j)$, $(x, y) = \sum_j x_j y_j$
$d(x, y)$	Distance (Euclidean squared) between two vector points $x = (x_j)$ and $y = (y_j)$, $d(x, y) = \sum_j (x_j - y_j)^2$
$\{S_1, ..., S_K\}$	Partition of set I in K disjoint classes $S_k \subset I$, $k = 1, ..., K$
K	Number of classes/clusters in a partition $S = \{S_1, ..., S_K\}$ of set I
N_k	Number of entities in class S_k of partition S $(k = 1, ..., K)$
$c_k = (c_{kv})$	Centroid of cluster S_k, $c_{kv} = \sum_{i \in S_k} y_{iv}/N_k$, $v \in V$
S_w, S_{w1}, S_{w2}	Parent-children triple in a cluster hierarchy, $S_w = S_{w1} \cup S_{w2}$
$dw(S_{w1}, S_{w2})$	Ward distance between clusters S_{w1}, with centroid c_1, and S_{w2}, with centroid c_2, $dw(S_{w1}, S_{w2}) = \frac{N_{w1}N_{w2}}{N_{w1}+N_{w2}} d(c_{w1}, c_{w2})$
N_{kv}	Number of entities in class S_k of partition S $(k = 1, ..., K)$ that fall in category $v \in V$; an entry in the contingency table between partition S and categorical feature l with set of categories V_l

N_{k+}	Marginal distribution: Number of entities in class S_k of partition S $(k = 1, ..., K)$ as related to a contingency table between partition S and another categorical feature
N_{+v}	Marginal distribution: Number of entities falling in category $v \in V_l$ of categorical feature l as related to a contingency table between partition S and categorical feature l
p_{kv}	Frequency N_{kv}/N
p_{k+}	N_{k+}/N
p_{+v}	N_{+v}/N
q_{kv}	Relative Quetelet coefficient, $q_{kv} = \frac{p_{kv}}{p_{k+}p_{+v}} - 1$
$T(Y)$	Data scatter, $T(Y) = \sum_{i \in I} \sum_{v \in V} y_{iv}^2$
$W(S_k, c_k)$	Cluster's square error, $W(S_k, c_k) = \sum_{i \in S_k} d(y_i, c_k)$
$W(S, c)$	K-Means square error criterion equal to the sum of $W(S_k, c_k)$, $k = 1, ...K$
$\beta(i, S_k)$	Attraction of $i \in I$ to cluster S_k

Introduction: Historical Remarks

Clustering is a discipline aimed at revealing groups, or clusters, of similar entities in data. The existence of clustering activities can be traced a hundred years back, in different disciplines in different countries.

One of the first was the discipline of ecology. A question the scientists were trying to address was of the territorial structure of the settlement of bird species and its determinants. They did field sampling to count numbers of various species at observation spots; similarity measures between spots were defined, and a method of analysis of the structure of similarity dubbed Wrozlaw taxonomy was developed in Poland between WWI and WWII (see publication of a later time [32]). This method survives, in an altered form, in diverse computational schemes such as single-linkage clustering and minimum spanning tree (see section 6.2.1).

Simultaneously, phenomenal activities in differential psychology initiated in the United Kingdom by the thrust of F. Galton (1822-1911) and supported by the mathematical genius of K. Pearson (1855-1936) in trying to prove that human talent is not a random gift but inherited, led to developing a body of multivariate statistics including the discipline of factor analysis (primarily, for measuring talent) and, as its offshoot, cluster analysis. Take, for example, a list of high school students and their marks at various disciplines such as maths, English, history, etc. If one believes that the marks are exterior manifestations of an inner quality, or factor, of talent, then one can assign a student i with a hidden factor score of his talent, z_i. Then marks x_{il} of student i at different disciplines l can be modeled, up to an error, by the product $c_l z_i$ so that $x_{il} \approx c_l z_i$ where factor c_l reflects the impact of the discipline l over students. The problem is to find the unknown z_i and c_l, given a set of students' marks over a set of disciplines. This was the idea behind a method proposed by K. Pearson in 1901 [106] that became the ground for later developments in Principal Component Analysis (PCA), see further explanation in section 5.1.3. To do the job of measuring hidden factors, F. Galton hired C. Spearman who developed a rather distinct method for factor analysis based on the assumption that no unique talent can explain various human abilities, but there are different, and independent, dimensions of talent such as linguistic or spatial ones. Each of these hidden dimensions must be presented by a corresponding independent factor so that the mark can be thought of as the total of factor scores weighted by their loadings. This idea proved fruitful in developing various personality theories and related psychological tests. However, methods for factor analysis developed between WWI and WWII were computationally intensive since they used the operation of inversion of a matrix of discipline-to-discipline similarity coefficients (covariances, to be exact). The operation of matrix inversion still can be a challenging task when the matrix size grows into thousands, and it was a nightmare before the electronic computer era even with a matrix size of a dozen. It was noted then that variables (in this case, disciplines) related to the same factor are highly correlated among themselves, which led to the idea of catching "clusters" of highly correlated variables as proxies for factors, without computing the inverse matrix, an activity which was referred to once as "factor analysis for the poor." The very first book on cluster analysis, within this framework, was published in 1939 [131], see also [55].

In the 50s and 60s of the 20th century, with computer powers made available at universities, cluster analysis research grew fast in many disciplines simultaneously. Three of these seem especially important for the development of cluster analysis as a scientific discipline.

First, machine learning of groups of entities (pattern recognition) sprang up to involve both supervised and unsupervised learning, the latter being synonymous to cluster analysis [21].

Second, the discipline of numerical taxonomy emerged in biology claiming that a biological taxon, as a rule, could not be defined in the Aristotelian way, with a conjunction of features: a taxon thus was supposed to be such a set of organisms in which a majority shared a majority of attributes with each other [122]. Hierarchical agglomerative and divisive clustering algorithms were supposed to formalize this. They were being "polythetic" by the very mechanism of their action in contrast to classical "monothetic" approaches in which every divergence of taxa was to be explained by a single character. (It should be noted that the appeal of numerical taxonomists left some biologists unimpressed; there even exists the so-called "cladistics" discipline that claims that a single feature ought always to be responsible for any evolutionary divergence.)

Third, in the social sciences, an opposite stance of building a divisive decision tree at which every split is made over a single feature emerged in the work of Sonquist and Morgan (see a later reference [124]). This work led to the development of decision tree techniques that became a highly popular part of machine learning and data mining. Decision trees actually cover three methods, conceptual clustering, classification trees and regression trees, that are usually considered different because they employ different criteria of homogeneity [58]. In a conceptual clustering tree, split parts must be as homogeneous as possible with regard to all participating features. In contrast, a classification tree or regression tree achieves homogeneity with regard to only one, so-called target, feature. Still, we consider that all these techniques belong in cluster analysis because they all produce split parts consisting of similar entities; however, this does not prevent them also being part of other disciplines such as machine learning or pattern recognition.

A number of books reflecting these developments were published in the 70s describing the great opportunities opened in many areas of human activity by algorithms for finding "coherent" clusters in a data "cloud" placed in geometrical space (see, for example, Benzécri 1973, Bock 1974, Clifford and Stephenson 1975, Duda and Hart 1973, Duran and Odell 1974, Everitt 1974, Hartigan 1975, Sneath and Sokal 1973, Sonquist, Baker, and Morgan 1973, Van Ryzin 1977, Zagoruyko 1972). In the next decade, some of these developments have been further advanced and presented in such books as Breiman et al. [11], Jain and Dubes [58] and McLachlan and Basford [82]. Still the common view is that clustering is an art rather than a science because determining clusters may depend more on the user's goals than on a theory. Accordingly, clustering is viewed as a set of diverse and ad hoc procedures rather than a consistent theory.

The last decade saw the emergence of data mining, the discipline combining issues of handling and maintaining data with approaches from statistics and machine learning for discovering patterns in data. In contrast to the statistical approach, which tries to find and fit objective regularities in data, data mining is oriented towards the end user. That means that data mining considers the problem of useful knowledge discovery in its entire range, starting from database acquisition to data preprocessing to finding patterns to drawing conclusions. In particular, the concept of an interesting pattern as something which is unusual or far from normal or anomalous has been introduced into data mining [29]. Obviously, an anomalous cluster is one that is further away from the grand mean or any other point of reference – an approach which is adapted in this text.

A number of computer programs for carrying out data mining tasks, clustering included, have been successfully exploited, both in science and industry; a review of them can be found in [23]. There are a number of general purpose statistical packages which have made it through from earlier times: those with some cluster analysis applications such as SAS [119] and SPSS[42] or those entirely devoted to clustering such as CLUSTAN [140]. There are data mining tools which include clustering, such as Clementine [14]. Still, these programs are far from sufficient in advising a user on what method to select, how to pre-process data and, especially, what sense to make of the clusters.

Another feature of this more recent period is that a number of application areas have emerged in which clustering is a key issue. In many application areas that began much earlier – such as image analysis, machine vision or robot planning – clustering is a rather small part of a very complex task such that the quality of clustering does not much matter to the overall performance; as any reasonable heuristic would do, these areas do not require the discipline of clustering to theoretically develop and mature.

This is not so in Bio-informatics, the discipline which tries to make sense of interrelation between structure, function and evolution of biomolecular objects. Its primary entities, DNA and protein sequences, are complex enough to have their similarity modeled as homology, that is, inheritance from a common ancestor. More advanced structural data such as protein folds and their contact maps are being constantly added to existing depositories. Gene expression technologies add to this an invaluable next step - a wealth of data on biomolecular function. Clustering is one of the major tools in the analysis of bioinformatics data. The very nature of the problem here makes researchers see clustering as a tool not only for finding cohesive groupings in data but also for relating the aspects of structure, function and evolution to each other. In this way, clustering is more and more becoming part of an emerging area of computer classification. It models the major functions of classification in the sciences: the structuring of a phenomenon and associating its different aspects. (Though, in data mining, the term 'classification' is almost exclusively used in its partial meaning as merely a diagnostic tool.) Theoretical and practical research in clustering is thriving in this area.

Another area of booming clustering research is information retrieval and text document mining. With the growth of the Internet and the World Wide Web, text has become one of the most important mediums of mass communication. The terabytes of text that exist must be summarized effectively, which involves a great deal of clustering in such key stages as natural language processing, feature extraction, categorization, annotation and summarization. In author's view, clustering will become even more important as the systems for acquiring and understanding knowledge from texts evolve, which is likely to occur soon. There are already web sites providing web search results with clustering them according to automatically found key phrases (see, for instance, [134]).

This book is mostly devoted to explaining and extending two clustering techniques, K-Means for partitioning and Ward for hierarchical clustering. The choice is far from random. First, they present the most popular clustering formats, hierarchies and partitions, and can be extended to other interesting formats such as single clusters. Second, many other clustering and statistical techniques, such as conceptual clustering, self-organizing maps (SOM), and contingency association measures, appear to be closely related to these. Third, both methods involve the same criterion, the minimum within cluster variance, which can be treated within the same theoretical framework. Fourth, many data mining issues of current interest, such as analysis of mixed data, incomplete clustering, and conceptual description of clusters, can be treated with extended versions of these methods. In fact, the book contents go far beyond these methods: the two last chapters, accounting for one third of the material, are devoted to the "big issues" in clustering and data mining that are not limited to specific methods.

The present account of the methods is based on a specific approach to cluster analysis, which can be referred to as the *data recovery clustering*. In this approach, clusters are not only found in data but they also feed back into the data: a cluster structure is used to generate data in the format of the data table which has been analyzed with clustering. The data generated by a cluster structure are, in a sense, "ideal" as they reproduce only the cluster structure lying behind their generation. The observed data can then be considered a noisy version of the ideal cluster-generated data; the extent of noise can be measured by the difference between the ideal and observed data. The smaller the difference the better the fit. This idea is not particularly new; it is, in fact, the backbone of many quantitative methods of multivariate statistics, such as regression and factor analysis. Moreover, it has been applied in clustering from the very beginning; in particular, Ward [135] developed his method of agglomerative clustering with implicitly this view of data analysis. Some methods were consciously constructed along the data recovery approach: see, for instance, work of Hartigan [46] at which the single linkage method was developed to approximate the data with an ultrametric matrix, an ideal data type corresponding to a cluster hierarchy. Even more appealing in this capacity is a later work by Hartigan [47].

However, this approach has never been applied in full. The sheer idea, following from models presented in this book, that classical clustering is but a constrained analogue to the principal component model has not achieved any popularity so far, though it has been around for quite a while [89], [90]. The unifying capability of the data recovery clustering is grounded on convenient relations which exist between data approximation problems and geometrically explicit classical clustering. Firm mathematical relations found between different parts of cluster solutions and data lead not only to explanation of the classical algorithms but also to development of a number of other algorithms for both finding and describing clusters. Among the former, principal-component-like algorithms for finding anomalous clusters and divisive clustering should be pointed out. Among the latter, a set of simple but efficient interpretation tools, that are absent from the multiple programs implementing classical clustering methods, should be mentioned.

Chapter 1

What Is Clustering

After reading this chapter the reader will have a general understanding of:

1. What clustering is and its basic elements.
2. Clustering goals.
3. Quantitative and categorical features.
4. Main cluster structures: partition, hierarchy, and single cluster.
5. Different perspectives at clustering coming from statistics, machine learning, data mining, and knowledge discovery.

A set of small but real-world clustering problems will be presented.

Base words

Association Finding interrelations between different aspects of a phenomenon by matching cluster descriptions in the feature spaces corresponding to the aspects.

Classification An actual or ideal arrangement of entities under consideration in classes to shape and keep knowledge, capture the structure of phenomena, and relate different aspects of a phenomenon in question to each other. This term is also used in a narrow sense referring to any activities in assigning entities to prespecified classes.

Cluster A set of similar data entities found by a clustering algorithm.

Cluster representative An element of a cluster to represent its "typical" properties. This is used for cluster description in domains knowledge of which is poor.

Cluster structure A representation of an entity set I as a set of clusters that form either a partition of I or hierarchy on I or an incomplete clustering of I.

Cluster tendency A description of a cluster in terms of the average values of relevant features.

Clustering An activity of finding and/or describing cluster structures in a data set.

Clustering goal Types of problems of data analysis to which clustering can be applied: associating, structuring, describing, generalizing and visualizing.

Clustering criterion A formal definition or scoring function that can be used in computational algorithms for clustering.

Conceptual description A logical statement characterizing a cluster or cluster structure in terms of relevant features.

Data A set of entities characterized by values of quantitative or categorical features. Sometimes data may characterize relations between entities such as similarity coefficients or transaction flows.

Data mining perspective In data mining, clustering is a tool for finding patterns and regularities within the data.

Generalization Making general statements about data and, potentially, about the phenomenon the data relate to.

Knowledge discovery perspective In knowledge discovery, clustering is a tool for updating, correcting and extending the existing knowledge. In this regard, clustering is but empirical classification.

Machine learning perspective In machine learning, clustering is a tool for prediction.

Statistics perspective In statistics, clustering is a method to fit a prespecified probabilistic model of the data generating mechanism.

Structuring Representing data with a cluster structure.

Visualization Mapping data onto a known "ground" image such as the coordinate plane or a genealogy tree – in such a way that properties of the data are reflected in the structure of the ground image.

1.1 Exemplary problems

Clustering is a discipline devoted to revealing and describing homogeneous groups of entities, that is, clusters, in data sets. Why would one need this? Here is a list of potentially overlapping objectives for clustering.

1. **Structuring**, that is, representing data as a set of groups of similar objects.

2. **Description** of clusters in terms of features, not necessarily involved in finding the clusters.

3. **Association**, that is, finding interrelations between different aspects of a phenomenon by matching cluster descriptions in spaces corresponding to the aspects.

4. **Generalization**, that is, making general statements about data and, potentially, the phenomena the data relate to.

5. **Visualization**, that is, representing cluster structures as visual images.

These categories are not mutually exclusive, nor do they cover the entire range of clustering goals but rather reflect the author's opinion on the main applications of clustering. In the remainder of this section we provide real-world examples of data and the related clustering problems for each of these goals. For illustrative purposes, small data sets are used in order to provide the reader with the opportunity of directly observing further processing with the naked eye.

1.1.1 Structuring

Structuring is the main goal of many clustering applications, which is to find principal groups of entities in their specifics. The cluster structure of an entity set can be looked at through different glasses. One user may wish to aggregate the set in a system of nonoverlapping classes; another user may prefer to develop a taxonomy as a hierarchy of more and more abstract concepts; yet another user may wish to focus on a cluster of "core" entities considering the rest as merely a nuisance. These are conceptualized in different types of cluster structures, such as a partition, a hierarchy, or a single subset.

Market towns

Table 1.1 represents a small portion of a list of thirteen hundred English market towns characterized by the population and services provided in each listed in the following box.

Market town features:	
P	Population resident in 1991 Census
PS	Primary Schools
Do	Doctor Surgeries
Ho	Hospitals
Ba	Banks and Building Societies
SM	National Chain Supermarkets
Pe	Petrol Stations
DIY	Do-It-Yourself Shops
SP	Public Swimming Pools
PO	Post Offices
CA	Citizen's Advice Bureaux (cheap legal advice)
FM	Farmers' Markets

For the purposes of social monitoring, the set of all market towns should be partitioned into similarity clusters in such a way that a representative from each of the clusters may be utilized as a unit of observation. Those characteristics of the clusters that separate them from the others should be used to properly select representative towns.

As further computations will show, the numbers of services on average follow the town sizes, so that the found clusters can be described mainly in terms of the population size. This set, as well as the complete set of almost thirteen hundred English market towns, consists of seven clusters that can be described as belonging to four tiers of population: large towns of about 17-20,000 inhabitants, two clusters of medium sized towns (8-10,000 inhabitants), three clusters of small towns (about 5,000 inhabitants) and a cluster of very small settlements with about 2,500 inhabitants. The difference between clusters in the same population tier is caused by the presence or absence of some service features. For instance, each of the three small town clusters is characterized by the presence of a facility, which is absent in two others: a Farm market, a Hospital and a Swimming pool, respectively. The number of clusters is determined in the process of computations (see sections 3.3, 3.4.2).

This data set is analyzed on pp. 52, 56, 68, 92, 94, 97, 99, 100, 101, 108.

Primates and Human origin

In Table 1.2, the data on genetic distances between Human and three genera of great apes are presented; the Rhesus monkey is added as a distant relative to certify the starting divergence event. It is well established that humans diverged from a common ancestor with chimpanzees approximately 5 million years ago, after a divergence from other great apes. Let us see how compatible with this conclusion the results of cluster analysis are.

Table 1.1: **Market towns:** Market towns in the West Country, England.

Town	P	PS	Do	Ho	Ba	SM	Pe	DIY	SP	PO	CA	FM
Ashburton	3660	1	0	1	2	1	2	0	1	1	1	0
Bere Alston	2362	1	0	0	1	1	0	0	0	1	0	0
Bodmin	12553	5	2	1	6	3	5	1	1	2	1	0
Brixham	15865	7	3	1	5	5	3	0	2	5	1	0
Buckfastleigh	2786	2	1	0	1	2	2	0	1	1	1	1
Bugle/Stenalees	2695	2	0	0	0	0	1	0	0	2	0	0
Callington	3511	1	1	0	3	1	1	0	1	1	0	0
Dartmouth	5676	2	0	0	4	4	1	0	0	2	1	1
Falmouth	20297	6	4	1	11	3	2	0	1	9	1	0
Gunnislake	2236	2	1	0	1	0	1	0	0	3	0	0
Hayle	7034	4	0	1	2	2	2	0	0	2	1	0
Helston	8505	3	1	1	7	2	3	0	1	1	1	1
Horrabridge/Yel	3609	1	1	0	2	1	1	0	0	2	0	0
Ipplepen	2275	1	1	0	0	0	1	0	0	1	0	0
Ivybridge	9179	5	1	0	3	1	4	0	0	1	1	0
Kingsbridge	5258	2	1	1	7	1	2	0	0	1	1	1
Kingskerswell	3672	1	0	0	0	1	2	0	0	1	0	0
Launceston	6466	4	1	0	8	4	4	0	1	3	1	0
Liskeard	7044	2	2	2	6	2	3	0	1	2	2	0
Looe	5022	1	1	0	2	1	1	0	1	3	1	0
Lostwithiel	2452	2	1	0	2	0	1	0	0	1	0	1
Mevagissey	2272	1	1	0	1	0	0	0	0	1	0	0
Mullion	2040	1	0	0	2	0	1	0	0	1	0	0
Nanpean/Foxhole	2230	2	1	0	0	0	0	0	0	2	0	0
Newquay	17390	4	4	1	12	5	4	0	1	5	1	0
Newton Abbot	23801	13	4	1	13	4	7	1	1	7	2	0
Padstow	2460	1	0	0	3	0	0	0	0	1	1	0
Penryn	7027	3	1	0	2	4	1	0	0	3	1	0
Penzance	19709	10	4	1	12	7	5	1	1	7	2	0
Perranporth	2611	1	1	0	1	1	2	0	0	2	0	0
Porthleven	3123	1	0	0	1	1	0	0	0	1	0	0
Saltash	14139	4	2	1	4	2	3	1	1	3	1	0
South Brent	2087	1	1	0	1	1	0	0	0	1	0	0
St Agnes	2899	1	1	0	2	1	1	0	0	2	0	0
St Austell	21622	7	4	2	14	6	4	3	1	8	1	1
St Blazey/Par	8837	5	2	0	1	1	4	0	0	4	0	0
St Columb Major	2119	1	0	0	2	1	1	0	0	1	1	0
St Columb Road	2458	1	0	0	0	1	3	0	0	2	0	0
St Ives	10092	4	3	0	7	2	2	0	0	4	1	0
St Just	2092	1	0	0	2	1	1	0	0	1	0	0
Tavistock	10222	5	3	1	7	3	3	1	2	3	1	1
Torpoint	8238	2	3	0	3	2	1	0	0	2	1	0
Totnes	6929	2	1	1	7	2	1	0	1	4	0	1
Truro	18966	9	3	1	19	4	5	2	2	7	1	1
Wadebridge	5291	1	1	0	5	3	1	0	1	1	1	0

Table 1.2: **Primates:** Distances between four Primate species and Rhesus monkey.

Genus	Human	Chimpanzee	Gorilla	Orangutan
Chimpanzee	1.45			
Gorilla	1.51	1.57		
Orangutan	2.98	2.94	3.04	
Rhesus monkey	7.51	7.55	7.39	7.10

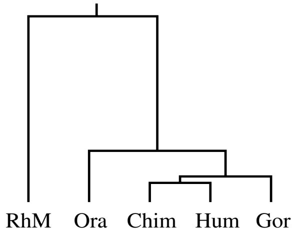


Figure 1.1: A tree representing pair-wise distances between the primate species from Table 1.2.

The data is a square matrix of the dissimilarity values between the species from Table 1.2 as cited in [90], p. 30. (Only sub-diagonal distances are shown since the table is symmetric.) An example of analysis of the structure of this matrix is given on p. 192.

The query: what species belongs to the same cluster as Humans? This obviously can be treated as a single cluster problem: one needs only one cluster to address the issue. The structure of the data is so simple that the cluster of chimpanzee, gorilla and human can be separated without any theory: distances within this subset are similar, all about the average 1.51, and by far less than other distances.

In biology, this problem is traditionally addressed through evolutionary trees, which are analogues to genealogy trees except that species play the role of relatives. An evolutionary tree built from the data in Table 1.2 is shown in Figure 1.1. The closest relationship between human and chimpanzee is obvious, with gorilla branching off next. The subject of human evolution is treated in depth with data mining methods in [13].

Gene presence-absence profiles

Evolutionary analysis is an important tool not only for understanding evolution but also for analysis of gene functions in humans and other organisms including medically and industrially important ones. The major assumption underlying the analysis is that all species are descendants of the same ancestor species, so that subsequent evolution can be depicted in terms of divergence only, as in the evolutionary tree in Figure 1.1.

The terminal nodes, so-called leaves, correspond to the species under consideration, and the root denotes the common ancestor. The other interior nodes represent other ancestral species, each being the last common ancestor to the set of organisms in the leaves of the sub-tree rooted in the given node. Recently, this line of research has been supplemented by data on the gene content of multiple species as exemplified in Table 1.3. Here, the columns correspond to 18 simple, unicellular organisms, bacteria and archaea (collectively called

Table 1.3: **Gene profiles**: Presence-absence profiles of 30 COGs in a set of 18 genomes.

No	COG	Species																	
		y	a	o	m	p	k	z	q	v	d	r	b	c	e	f	g	s	j
1	COG0090	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	COG0091	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	COG2511	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
4	COG0290	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
5	COG0215	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	COG2147	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
7	COG1746	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
8	COG1093	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
9	COG2263	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
10	COG0847	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1
11	COG1599	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
12	COG3066	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
13	COG3293	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1
14	COG3432	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
15	COG3620	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
16	COG1709	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0
17	COG1405	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
18	COG3064	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
19	COG2853	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
20	COG2951	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
21	COG3114	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
22	COG3073	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
23	COG3026	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
24	COG3006	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
25	COG3115	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
26	COG2414	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0
27	COG3029	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0
28	COG3107	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
29	COG3429	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
30	COG1950	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0

Table 1.4: **Species**: List of eighteen species (one eukaryota, then six archaea and then eleven bacteria) represented in Table 1.3.

Species	Code	Species	Code
Saccharomyces cerevisiae	y	*Deinococcus radiodurans*	d
Archaeoglobus fulgidus	a	*Mycobacterium tuberculosis*	r
Halobacterium sp.NRC-1	o	*Bacillus subtilis*	b
Methanococcus jannaschii	m	*Synechocystis*	c
Pyrococcus horikoshii	k	*Escherichia coli*	e
Thermoplasma acidophilum	p	*Pseudomonas aeruginosa*	f
Aeropyrum pernix	z	*Vibrio cholera*	g
Aquifex aeolicus	q	*Xylella fastidiosa*	s
Thermotoga maritima	v	*Caulobacter crescentus*	j

Table 1.5: COG names and functions.

Code	Name
COG0090	Ribosomal protein L2
COG0091	Ribosomal protein L22
COG2511	Archaeal Glu-tRNAGln
COG0290	Translation initiation factor IF3
COG0215	Cysteinyl-tRNA synthetase
COG2147	Ribosomal protein L19E
COG1746	tRNA nucleotidyltransferase (CCA-adding enzyme)
COG1093	Translation initiation factor eIF2alpha
COG2263	Predicted RNA methylase
COG0847	DNA polymerase III epsilon
COG1599	Replication factor A large subunit
COG3066	DNA mismatch repair protein
COG3293	Predicted transposase
COG3432	Predicted transcriptional regulator
COG3620	Predicted transcriptional regulator with C-terminal CBS domains
COG1709	Predicted transcriptional regulators
COG1405	Transcription initiation factor IIB
COG3064	Membrane protein involved
COG2853	Surface lipoprotein
COG2951	Membrane-bound lytic murein transglycosylase B
COG3114	Heme exporter protein D
COG3073	Negative regulator of sigma E
COG3026	Negative regulator of sigma E
COG3006	Uncharacterized protein involved in chromosome partitioning
COG3115	Cell division protein
COG2414	Aldehyde:ferredoxin oxidoreductase
COG3029	Fumarate reductase subunit C
COG3107	Putative lipoprotein
COG3429	Uncharacterized BCR, stimulates glucose-6-P dehydrogenase activity
COG1950	Predicted membrane protein

prokaryotes), and a simple eukaryote, yeast *Saccharomyces cerevisiae*. The list of species along with their one-letter codes is given in Table 1.4.

The rows in Table 1.3 correspond to individual genes represented by the so-called Clusters of Orthologous Groups (COGs) which are supposed to include genes originating from the same ancestral gene in the common ancestor of the respective species [68]. COG names which reflect the functions of the respective genes in the cell are given in Table 1.5. These tables present but a small part of the publicly available COG database currently including 66 species and 4857 COGs posted in the web site www.ncbi.nlm.nih.gov/COG.

The pattern of presence-absence of a COG in the analyzed species is shown in Table 1.3, with zeros and ones standing for absence and presence, respectively. This way, a COG can be considered a character (attribute) that is either present or absent in a species. Two of the COGs, in the top two rows, are present at each of the 18 genomes, whereas the others cover only some of the species.

An evolutionary tree must be consistent with the presence-absence patterns.

Specifically, if a COG is present in two species, then it should be present in their last common ancestor and, thus, in all other descendants of the last common ancestor. This would be in accord with the natural process of inheritance. However, in most cases, the presence-absence pattern of a COG in extant species is far from the "natural" one: many genes are dispersed over several subtrees. According to comparative genomics, this may happen because of multiple loss and horizontal transfer of genes [68]. The hierarchy should be constructed in such a way that the number of inconsistencies is minimized.

The so-called principle of Maximum Parsimony (MP) is a straightforward formalization of this idea. Unfortunately, MP does not always lead to appropriate solutions because of intrinsic and computational problems. A number of other approaches have been proposed including hierarchical cluster analysis (see [105]).

Especially appealing in this regard is divisive cluster analysis. It begins by splitting the entire data set into two parts, thus imitating the divergence of the last universal common ancestor (LUCA) into two descendants. The same process then applies to each of the split parts until a stop-criterion is reached to halt the division process. In contrast to other methods for building evolutionary trees, divisive clustering imitates the process of evolutionary divergence. Further approximation of the real evolutionary process can be achieved if the characters on which divergence is based are discarded immediately after the division of the respective cluster [96]. Gene profiles data are analyzed on p. 121 and p. 131.

After an evolutionary tree is built, it can be utilized for reconstructing gene histories by mapping events of emergence, inheritance, loss and horizontal transfer of individual COGs on the tree according to the principle of Maximum Parsimony (see p. 126). These histories of individual genes can be helpful in advancing our understanding of biological functions and drug design.

1.1.2 Description

The problem of description is that of automatically deriving a conceptual description of clusters found by a clustering algorithm or supplied from a different source. The problem of cluster description belongs in cluster analysis because this is part of the interpretation and understanding of clusters. A good conceptual description can be used for better understanding and/or better predicting. The latter because we can check whether an object in question satisfies the description or not: the more the object satisfies the description the better the chances that it belongs to the cluster described. This is why conceptual description tools, such as decision trees [11, 23], have been conveniently used and developed mostly for the purposes of prediction.

Describing Iris genera

Table 1.6 presents probably the most popular data set in the machine learning research community: 150 Iris specimens, each measured on four morphological variables: sepal length (w1), sepal width (w2), petal length (w3), and petal width (w4), as collected by botanist E. Anderson and published in a founding paper of celebrated British statistician R. Fisher in 1936 [7]. It is said that there are three species in the table, I *Iris setosa* (diploid), II *Iris versicolor* (tetraploid), and III *Iris virginica* (hexaploid), each represented by 50 consecutive entities in the corresponding column.

The classes are defined by the genome (genotype); the features are of the appearance (phenotype). Can the classes be described in terms of the features in Table 1.6? It is well known from previous studies that classes II and III are not well separated in the variable space (for example, specimens 28, 33 and 44 from class II are more similar to specimens 18, 26, and 33 from class III than to specimens of the same species, see Figure 1.10 on p. 25). This leads to the problem of deriving new features from those that have been measured on spot to provide for better descriptions of the classes. These new features could be then utilized for the clustering of additional specimens.

Some non-linear machine learning techniques such as Neural Nets [51] and Support Vector Machines [128] can tackle the problem and produce a decent decision rule involving non-linear transformation of the features. Unfortunately, rules that can be derived with currently available methods are not comprehensible to the human mind and, thus, cannot be used for interpretation and description. The human mind needs somewhat less artificial logics that can reproduce and extend such botanists' observations as that the petal area roughly expressed by the product of w3 and w4 provides for much better resolution than the original linear sizes. A method for building cluster descriptions of this type, referred to as APPCOD, will be described in section 7.2.

The Iris data set is analyzed on pp. 87, 211, 212, 213.

Body mass

Table 1.7 presents data on the height and weight of 22 males of which individuals p13-p22 are considered overweight and p1-p12 normal. As Figure 1.2 clearly shows, a line of best fit separating these two sets should run along the elongated cloud formed by entity points. The groups have been defined according to the so-called body mass index, bmi: those individuals whose bmi is 25 or over are considered overweight. The body mass index is defined as the ratio of the weight, in kilograms, to the squared height, in meters. The problem is to make a computer automatically transform the current height-weight feature space into such a format that would allow one to clearly distinguish between the overweight and normally-built individuals.

Table 1.6: **Iris:** Anderson-Fisher data on 150 Iris specimens.

Entity in	Class I				Class II				Class III			
a Class	Iris setosa				Iris versicolor				Iris virginica			
	w1	w2	w3	w4	w1	w2	w3	w4	w1	w2	w3	w4
1	5.1	3.5	1.4	0.3	6.4	3.2	4.5	1.5	6.3	3.3	6.0	2.5
2	4.4	3.2	1.3	0.2	5.5	2.4	3.8	1.1	6.7	3.3	5.7	2.1
3	4.4	3.0	1.3	0.2	5.7	2.9	4.2	1.3	7.2	3.6	6.1	2.5
4	5.0	3.5	1.6	0.6	5.7	3.0	4.2	1.2	7.7	3.8	6.7	2.2
5	5.1	3.8	1.6	0.2	5.6	2.9	3.6	1.3	7.2	3.0	5.8	1.6
6	4.9	3.1	1.5	0.2	7.0	3.2	4.7	1.4	7.4	2.8	6.1	1.9
7	5.0	3.2	1.2	0.2	6.8	2.8	4.8	1.4	7.6	3.0	6.6	2.1
8	4.6	3.2	1.4	0.2	6.1	2.8	4.7	1.2	7.7	2.8	6.7	2.0
9	5.0	3.3	1.4	0.2	4.9	2.4	3.3	1.0	6.2	3.4	5.4	2.3
10	4.8	3.4	1.9	0.2	5.8	2.7	3.9	1.2	7.7	3.0	6.1	2.3
11	4.8	3.0	1.4	0.1	5.8	2.6	4.0	1.2	6.8	3.0	5.5	2.1
12	5.0	3.5	1.3	0.3	5.5	2.4	3.7	1.0	6.4	2.7	5.3	1.9
13	5.1	3.3	1.7	0.5	6.7	3.0	5.0	1.7	5.7	2.5	5.0	2.0
14	5.0	3.4	1.5	0.2	5.7	2.8	4.1	1.3	6.9	3.1	5.1	2.3
15	5.1	3.8	1.9	0.4	6.7	3.1	4.4	1.4	5.9	3.0	5.1	1.8
16	4.9	3.0	1.4	0.2	5.5	2.3	4.0	1.3	6.3	3.4	5.6	2.4
17	5.3	3.7	1.5	0.2	5.1	2.5	3.0	1.1	5.8	2.7	5.1	1.9
18	4.3	3.0	1.1	0.1	6.6	2.9	4.6	1.3	6.3	2.7	4.9	1.8
19	5.5	3.5	1.3	0.2	5.0	2.3	3.3	1.0	6.0	3.0	4.8	1.8
20	4.8	3.4	1.6	0.2	6.9	3.1	4.9	1.5	7.2	3.2	6.0	1.8
21	5.2	3.4	1.4	0.2	5.0	2.0	3.5	1.0	6.2	2.8	4.8	1.8
22	4.8	3.1	1.6	0.2	5.6	3.0	4.5	1.5	6.9	3.1	5.4	2.1
23	4.9	3.6	1.4	0.1	5.6	3.0	4.1	1.3	6.7	3.1	5.6	2.4
24	4.6	3.1	1.5	0.2	5.8	2.7	4.1	1.0	6.4	3.1	5.5	1.8
25	5.7	4.4	1.5	0.4	6.3	2.3	4.4	1.3	5.8	2.7	5.1	1.9
26	5.7	3.8	1.7	0.3	6.1	3.0	4.6	1.4	6.1	3.0	4.9	1.8
27	4.8	3.0	1.4	0.3	5.9	3.0	4.2	1.5	6.0	2.2	5.0	1.5
28	5.2	4.1	1.5	0.1	6.0	2.7	5.1	1.6	6.4	3.2	5.3	2.3
29	4.7	3.2	1.6	0.2	5.6	2.5	3.9	1.1	5.8	2.8	5.1	2.4
30	4.5	2.3	1.3	0.3	6.7	3.1	4.7	1.5	6.9	3.2	5.7	2.3
31	5.4	3.4	1.7	0.2	6.2	2.2	4.5	1.5	6.7	3.0	5.2	2.3
32	5.0	3.0	1.6	0.2	5.9	3.2	4.8	1.8	7.7	2.6	6.9	2.3
33	4.6	3.4	1.4	0.3	6.3	2.5	4.9	1.5	6.3	2.8	5.1	1.5
34	5.4	3.9	1.3	0.4	6.0	2.9	4.5	1.5	6.5	3.0	5.2	2.0
35	5.0	3.6	1.4	0.2	5.6	2.7	4.2	1.3	7.9	3.8	6.4	2.0
36	5.4	3.9	1.7	0.4	6.2	2.9	4.3	1.3	6.1	2.6	5.6	1.4
37	4.6	3.6	1.0	0.2	6.0	3.4	4.5	1.6	6.4	2.8	5.6	2.1
38	5.1	3.8	1.5	0.3	6.5	2.8	4.6	1.5	6.3	2.5	5.0	1.9
39	5.8	4.0	1.2	0.2	5.7	2.8	4.5	1.3	4.9	2.5	4.5	1.7
40	5.4	3.7	1.5	0.2	6.1	2.9	4.7	1.4	6.8	3.2	5.9	2.3
41	5.0	3.4	1.6	0.4	5.5	2.5	4.0	1.3	7.1	3.0	5.9	2.1
42	5.4	3.4	1.5	0.4	5.5	2.6	4.4	1.2	6.7	3.3	5.7	2.5
43	5.1	3.7	1.5	0.4	5.4	3.0	4.5	1.5	6.3	2.9	5.6	1.8
44	4.4	2.9	1.4	0.2	6.3	3.3	4.7	1.6	6.5	3.0	5.5	1.8
45	5.5	4.2	1.4	0.2	5.2	2.7	3.9	1.4	6.5	3.0	5.8	2.2
46	5.1	3.4	1.5	0.2	6.4	2.9	4.3	1.3	7.3	2.9	6.3	1.8
47	4.7	3.2	1.3	0.2	6.6	3.0	4.4	1.4	6.7	2.5	5.8	1.8
48	4.9	3.1	1.5	0.1	5.7	2.6	3.5	1.0	5.6	2.8	4.9	2.0
49	5.2	3.5	1.5	0.2	6.1	2.8	4.0	1.3	6.4	2.8	5.6	2.2
50	5.1	3.5	1.4	0.2	6.0	2.2	4.0	1.0	6.5	3.2	5.1	2.0

Table 1.7: **Body mass**: Height and weight of twenty-two individuals.

Individual	Height, cm	Weight, kg
p1	160	63
p2	160	61
p3	165	64
p4	165	67
p5	164	65
p6	164	62
p7	157	60
p8	158	60
p9	175	75
p10	173	74
p11	180	79
p12	185	84
p13	160	67
p14	160	71
p15	170	73
p16	170	76
p17	180	82
p18	180	85
p19	175	78
p20	174	77
p21	171	75
p22	170	75

The best thing would be if a computer could derive the bmi based decision rule itself, which may not be necessarily the case since the bmi is defined universally whereas only a very limited data set is presented here. One would obviously have to consider whether a linear description could be derived such as the following existing rule of thumb: a man is overwheight if the difference between his height in cm and weight in kg is greater than one hundred. A man 175 cm in height should normally weigh 75 kg or less according to this rule.

Once again it should be pointed out that non-linear transformations supplied by machine learning tools for better prediction may be not necessarily usable for the purposes of description.

The Body mass data set is analyzed on pp. 205, 213, 242.

1.1.3 Association

Revealing associations between different aspects of phenomena is one of the most important goals of classification. Clustering as a classification of empirical data also can do the job. A relation between different aspects of a phenomenon in question can be established if the same clusters are well described twice,

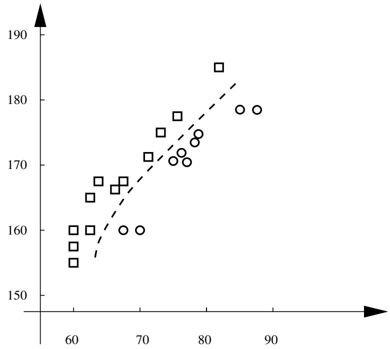


Figure 1.2: Twenty-two individuals at the height-weight plane.

each description related to one of the aspects. Different descriptions of the same cluster are then obviously linked as those referring to the same contents, though possibly with different errors.

Digits and patterns of confusion between them

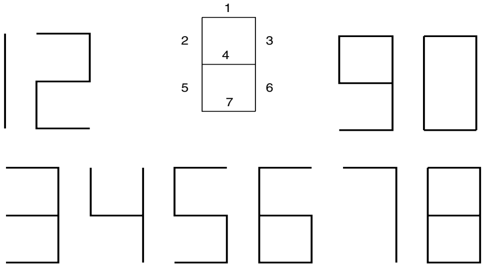


Figure 1.3: Styled digits formed by segments of the rectangle.

The rectangle in the upper part of Figure 1.3 is used to draw numeral digits around it in a styled manner of the kind used in digital electronic devices. Seven binary presence/absence variables e1, e2,..., e7 in Table 1.8 correspond to the numbered segments on the rectangle in Figure 1.3.

Although the digit character images may seem arbitrary, finding patterns of similarity in them may be of interest in training operators dealing with digital numbers.

Table 1.8: **Digits:** Segmented numerals presented with seven binary variables corresponding to presence/absence of the corresponding edge in Figure 1.3.

Digit	e1	e2	e3	e4.	e5	e6	e7
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1
0	1	1	1	0	1	1	1

Table 1.9: **Confusion:** Confusion between the segmented numeral digits.

Stimulus	Response									
	1	2	3	4	5	6	7	8	9	0
1	877	7	7	22	4	15	60	0	4	4
2	14	782	47	4	36	47	14	29	7	18
3	29	29	681	7	18	0	40	29	152	15
4	149	22	4	732	4	11	30	7	41	0
5	14	26	43	14	669	79	7	7	126	14
6	25	14	7	11	97	633	4	155	11	43
7	269	4	21	21	7	0	667	0	4	7
8	11	28	28	18	18	70	11	577	67	172
9	25	29	111	46	82	11	21	82	550	43
0	18	4	7	11	7	18	25	71	21	818

Results of a psychological experiment on confusion between the segmented numerals are in Table 1.9. A digit appeared on a screen for a very short time (stimulus), and an individual was asked to report what was the digit (response). The response frequencies of digits versus shown stimuli stand in the rows of Table 1.9 [90].

The problem is to find general patterns in confusion and to interpret them in terms of the segment presence-absence variables in Digits data Table 1.8. If the found interpretation can be put in a theoretical framework, the patterns can be considered as empirical reflections of theoretically substantiated classes. Patterns of confusion would show the structure of the phenomenon. Interpretation of the clusters in terms of the drawings, if successful, would allow us to see what relation may exist between the patterns of drawing and confusion.

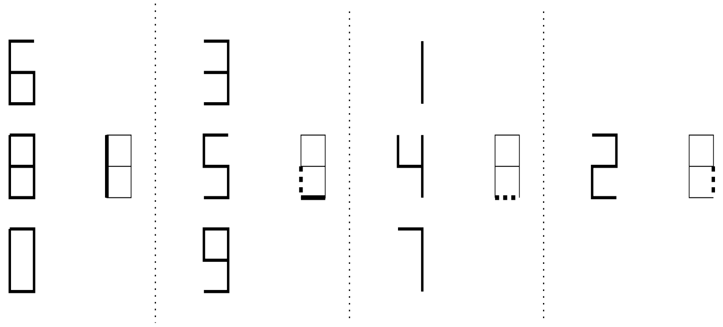


Figure 1.4: Visual representation of four Digits confusion clusters: solid and dotted lines over the rectangle show distinctive features that must be present in or absent from all entities in the cluster.

Indeed, four major confusion clusters can be distinguished in the Digits data, as will be found in section 4.4.2 and described in section 6.3 (see pp. 73, 129, 133 and 134 for computations on these data). On Figure 1.4 these four clusters are presented with distinctive features shown with segments defining the drawing of digits. We can see that all relevant features are concentrated on the left and down the rectangle. It remains to be seen if there is any physio-psychological mechanism behind this and how it can be utilized.

Moreover, it appears the attributes in Table 1.8 are quite relevant on their own, pinpointing the same patterns that have been identified as those of confusion. This can be clearly seen in Figure 1.5, which illustrates a classification tree for Digits found using an algorithm for conceptual clustering presented in section 4.3. On this tree, clusters are the terminal boxes and interior nodes are labeled by the features involved in classification. The coincidence of the drawing clusters with confusion patterns indicates that the confusion is caused by the segment features participating in the tree. These appear to be the same features in both Figure 1.4 and Figure 1.5.

Literary masterpieces

The data in Table 1.10 reflect the language and style features of eight novels by three great writers of the nineteenth century. Two language features are:

1) LenSent - Average length of (number of words in) sentences;

2) LenDial - Average length of (number of sentences in) dialogues. (It is assumed that longer dialogues are needed if the author uses dialogue as a device to convey information or ideas to the reader.)

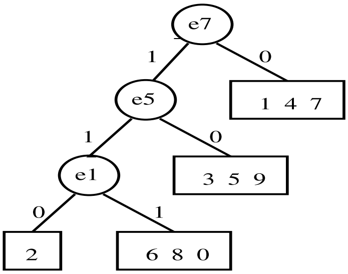


Figure 1.5: The conceptual tree of Digits.

Table 1.10: **Masterpieces:** Masterpieces of 19th century: the first three by Charles Dickens (1812–1870), the next three by Mark Twain (1835–1910), and the last two by Leo Tolstoy (1828–1910).

Title	LenSent	LenDial	NChar	SCon	Narrative
Oliver Twist	19.0	43.7	2	No	Objective
Dombey and Son	29.4	36.0	3	No	Objective
Great Expectations	23.9	38.0	3	No	Personal
Tom Sawyer	18.4	27.9	2	Yes	Objective
Huckleberry Finn	25.7	22.3	3	Yes	Personal
Yankee at King Arthur	12.1	16.9	2	Yes	Personal
War and Peace	23.9	30.2	4	Yes	Direct
Anna Karenina	27.2	58.0	5	Yes	Direct

Features of style:

3) NChar - Number of principal characters (the larger the number the more themes raised);

4) SCon - Yes or No depending on the usage of the stream of conscience techniques;

5) Narrative - The narrative style is a qualitative feature categorized as: (a) Personal (if the narrative comes from the mouth of a character such as Pip in “Great Expectations” by Charles Dickens), or (b) Objective (if the subject develops mainly through the behavior of the characters and other indirect means), or (c) Direct (if the author prefers to directly intervene with the comments and explanations).

As we have seen already with the Digits data, features are not necessarily quantitative. They also can be categorical, such as SCon, a binary variable, or Narrative, a nominal variable.

The data in Table 1.10 can be utilized to advance two of the clustering goals:

1. **Structurization:** To cluster the set of masterpieces and intensionally describe clusters in terms of the features. We expect the clusters to accord to the three authors and convey features of their style.

2. **Association:** To analyze interrelations between two aspects of prose writing: (a) linguistic (presented by LenSent and LenD), and (b) the author's narrative style (the other three variables). For instance, we may find clusters in the linguistic features space and conceptually describe them in terms of the narrative style features. The number of entities that do not satisfy the description will score the extent of correlation. We expect, in this particular case, to have a high correlation between these aspects, since both must depend on the same cause (the author) which is absent from the feature list (see page 104).

This data set is used for illustration of many concepts and methods described further on; see pp. 61, 62, 78, 79, 80, 81, 84, 89, 104, 105, 162, 182, 193, 195, 197.

1.1.4 Generalization

Generalization, or overview, of data is a (set of) statement(s) about properties of the phenomenon reflected in the data under consideration. To make a generalization with clustering, one may need to do a multistage analysis: at first, structure the entity set; second, describe clusters; third, find associations between different aspects.

Probably one of the most exciting applications of this type can be found in the newly emerging area of text mining [139]. With the abundance of text information flooding every Internet user, the discipline of text mining is flourishing. A traditional paradigm in text mining is underpinned by the concept of the key word. The key word is a string of symbols (typically corresponding to a language word or phrase) that is considered important for the analysis of a pre-specified collection of texts. Thus, first comes a collection of texts defined by a meaningful query such as "recent mergers among insurance companies" or "medieval Britain." (Keywords can be produced by human experts in the domain or from statistical analyses of the collection.) Then a virtual or real text-to-keyword table can be created with keywords treated as features. Each of the texts (entities) can be represented by the number of occurrences of each of the keywords. Clustering of such a table may lead to finding subsets of texts covering different aspects of the subject.

This approach is being pursued by a number of research and industrial groups, some of which have built clustering engines on top of Internet search engines: given a query, such a clustering engine singles out several dozen of the most relevant web pages, resulting from a search by a search engine such as

Table 1.11: List of eleven features I-XI and their categories with respect to five aspects of a Bribery situation.

Actor	Service	Interaction	Environment
I. Office	**III. Type of service**	**V. Initiator**	**IX. Condition**
1. Enterprise	1. Obstr. of justice	1. Client	1. Regular routine
2. City	2. Favors	2. Official	2. Monitoring
3. Region	3. Extortion		3. Sloppy regulations
4. Federal	4. Category change		4. Irregular
	5. Cover-up		
II. Client	**IV. Occurrence**	**VI. Bribe level**	**X. Branch**
1. Individual	1. Once	1. $10K or less	1. Government
2. Business	2. Multiple	2. Up to $100K	2. Law enforcement
		3. ≥$100K	3. Other
		VII. Type	**XI. Punishment**
		1. Infringement	1. None
		2. Extortion	2. Administrative
			3. Arrest
		VIII. Network	4. Arrest followed by release
		1. None	5. Arrest with imprisonment
		2. Within office	
		3. Between offices	
		4. Clients	

Google or Yahoo, finds keywords or phrases in the corresponding texts, clusters web pages according to the keywords used as features, and then describes clusters in terms of the most relevant keywords or phrases. Two top web sites which have been found from searching for “clustering engines” with Google on 29 June 2004 in London are Vivisimo at hhtp://vivisimo.com and iBoogie at http://iboogie.tv. The former is built on top of ten popular search engines and can be used for partitioning web pages from several different sources such as “Web” or “Top stories,” the latter maintains several dozen languages and presents a hierarchical classification of selected web pages. In response to the query “clustering” Vivisimo produced 232 web pages in a “Web” category and 117 in a “Top news” category. Among top news the most populated clusters were “Linux” (16 items), “Stars” (12), and “Bombs” (11). Among general web sites the most numerous were “Linux” (25), “Search, Engine” (21), “Computing” (22), etc. More or less random web sites devoted to individual papers or scientists or scientific centers or commercial companies have been listed under categories “Visualization” (12), “Methods” (7), “Clustering” (7), etc. Such categories as “White papers” contained pages devoted to both computing clusters and cluster analysis. Similar results, though somewhat more favourable towards clustering as data mining, have been produced with iBoogie. Its cluster “Cluster” (51) was further divided into categories such as “computer” (10) and “analysis” (5). Such categories as “software for clustering” and “data clustering" have been presented too to refer to a random mix of 24 and 20 web sites respectively.

The activity of generalization so far mainly relies on human experts who supply understanding of a substantive area behind the text corpus. Human experts develop a text-to-feature data table that can be further utilized for generalization. Such is a collection of 55 articles on Bribery cases from central Russian newspapers 1999-2000 presented in Table 1.12 according to [97]. The features reflect the following fivefold structure of bribery situations: two interacting sides - the office and the client, their interaction, the corrupt service rendered, and the environment in which it all occurs.

These structural aspects can be characterized by eleven features that can be recovered from the newspaper articles; they are presented in Table 1.11.

To show how these features can be applied to a newspaper article, let us quote an article that appeared in a newspaper called "Kommersant" on 20 March 1999 (translated from Russian):

> **Mayor of a coal town under arrest**
> Thursday this week, Mr Evgeny Parshukov, Mayor of town Belovo near Kemerovo, was arrested under a warrant issued by the region attorney, Mr. Valentin Simuchenkov. The mayor is accused of receiving a bribe and abusing his powers for wrongdoing. Before having been elected to the mayoral post in June 1997, he received a credit of 62,000 roubles from Belovo Division of KUZBASS Transport Bank to support his election campaign. The Bank then cleared up both the money and interest on it, allegedly because after his election Mr. Parshukov ordered the Finance Department of the town administration, as well as all municipal organisations in Belovo, to move their accounts into the Transport Bank. Also, the attorney office claims that in 1998 Mr. Parshukov misspent 700,000 roubles from the town budget. The money came from the Ministry of Energy specifically aimed at creating new jobs for mine workers made redundant because their mines were getting closed. However, Mr. Parshukov ordered to lend the money at a high interest rate to the Municipal Transport agency. Mr. Parshukov doesn't deny the facts. He claims however that his actions involve no crime.

A possible coding of the eleven features in this case constitutes the contents of row 29 in Table 1.12. The table presents 55 cases that could be more or less unambiguously coded (from the original 66 cases [98]).

The prime problem here is similar to those in the Market towns and Digits data: to see if there are any patterns at all. To generalize, one has to make sense of patterns in terms of the features. In other words, we are interested in getting a synoptic description of the data in terms of clusters which are to be found and described.

On the first glance, no structure exists in the data. Nor could the scientists

Table 1.12: **Bribery:** data with features from Table 1.11.

Case	Of	Cl	Serv	Occ	Ini	Br	Typ	Net	Con	Branch	Pun
1	2	2	2	2	1	2	1	3	3	1	5
2	2	1	5	2	1	1	1	3	2	1	5
3	2	2	2	1	1	3	1	1	3	1	4
4	1	2	3	1	2	1	2	1	3	1	3
5	1	1	4	2	2	1	1	4	3	3	3
6	1	2	3	1	2	2	2	1	3	1	5
7	3	2	1	1	2	3	2	3	2	2	5
8	2	2	4	2	1	1	1	2	1	2	5
9	1	2	3	1	2	1	2	1	1	1	5
10	1	2	3	1	2	1	2	1	1	1	5
11	2	2	5	1	2	3	2	2	2	1	5
12	2	2	1	1	2	2	1	1	4	2	5
13	3	2	1	1	1	3	1	1	4	2	2
14	2	1	4	1	2	1	1	2	1	2	5
15	3	2	2	1	2	1	1	2	3	1	5
16	2	1	4	2	2	1	1	1	1	3	3
17	4	2	2	1	1	2	1	1	4	1	5
18	2	2	5	1	1	2	2	2	3	2	5
19	2	2	5	1	2	1	2	1	3	2	5
20	2	2	1	1	1	2	1	4	3	2	5
21	1	2	3	1	2	2	2	1	3	1	5
22	1	2	2	2	2	2	1	1	4	1	3
23	1	1	4	1	2	1	1	1	1	3	5
24	3	2	5	1	2	1	1	2	2	2	2
25	2	1	2	2	2	1	2	1	3	3	5
26	1	2	5	1	2	2	1	1	3	1	3
27	1	1	4	2	1	2	1	2	4	3	5
28	1	1	5	1	2	1	1	1	2	3	5
29	2	2	2	1	2	2	1	1	3	1	5
30	2	2	3	1	2	2	2	3	3	1	5
31	2	2	3	1	2	1	2	3	3	1	5
32	4	2	2	1	2	1	1	1	3	1	5
33	3	1	1	1	2	1	1	1	4	2	3
34	3	2	1	2	1	1	1	2	4	2	5
35	3	1	3	2	2	1	2	3	4	2	3
36	2	1	3	1	2	2	1	3	1	2	5
37	2	2	5	1	2	3	1	1	2	2	5
38	2	1	1	1	2	2	1	3	4	2	4
39	2	2	1	1	1	1	1	4	4	2	5
40	1	2	3	2	2	1	2	1	3	2	5
41	1	1	4	2	2	1	2	1	2	3	5
42	1	1	4	2	2	1	1	1	3	3	5
43	2	1	1	1	1	1	1	4	4	2	5
44	3	2	5	2	2	1	1	1	2	2	5
45	2	1	1	1	1	1	1	2	4	2	5
46	3	2	5	2	2	1	1	2	2	2	5
47	2	1	1	1	2	1	2	2	4	2	1
48	2	1	1	1	2	1	1	1	4	2	5
49	3	1	2	1	1	2	1	3	4	1	5
50	3	1	1	1	2	1	1	1	4	2	5
51	2	1	2	1	2	1	1	1	4	2	5
52	3	2	1	1	2	1	2	1	2	2	5
53	2	2	5	1	2	2	2	2	2	2	5
54	2	2	5	2	2	2	1	2	2	2	5
55	2	2	5	1	2	3	2	1	2	2	2

specializing in the research of corruption see any. However, after applying an intelligent version of the algorithm K-Means as described later in example 3.20, section 3.3, a rather simple core structure could be found that is defined by just two features and determines all other aspects. The results provide for a really short generalization: "It is the branch of government that determines which of the five types of corrupt services are rendered: Local government → Favors or Extortion; Law enforcement → Obstruction of Justice or Cover-Up; and Other → Category Change." A detailed discussion is given in examples on pp. 95, 106 and 147.

1.1.5 Visualization of data structure

Visualization is considered a rather vague area involving psychology, cognitive sciences and other disciplines, which is rapidly developing. In the current thinking, the subject of data visualization is defined as creation of mental images to gain insight and understanding [125]. This, however, seems too wide and includes too many non-operational images such as realistic and surrealistic paintings. In our presentation, we take on a more operational view and consider that data visualization is an activity related to mapping data onto a known ground image such as a coordinate plane, geography map, or a genealogy tree in such a way that properties of the data are reflected in the structure of the ground image.

Among ground images, the following are the most popular: geographical maps, networks, 2D displays of one-dimensional objects such as graphs or pie-charts or histograms, 2D displays of two-dimensional objects, and block structures. Sometimes, the very nature of the data suggests what ground image should be used. All of these can be used with clustering, and we are going to review most of them except for geographical maps.

One-dimensional data

One-dimensional data over pre-specified groups or found clusters can be of two types: (a) the distribution of entities over groups and (b) values of a feature within clusters. Accordingly, there can be two types of visual support for these.

Consider, for instance, groups of the Market town data defined by the population. According to Table 1.1 the population ranges approximately between 2000 and 24000 habitants. Let us divide the range in five equal intervals, bins, that are defined thus to have size (24000-2000)/5=4400 and bound points 6400, 10800, 15200, and 19600.

In Table 1.13 the data of the groups are displayed: their absolute and relative sizes and also the average numbers of Banks and the standard deviations within them. For the definitions of the average and standard deviation see section 2.1.2.

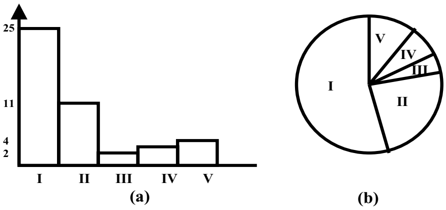


Figure 1.6: Histogram (a) and pie-chart (b) presenting the distribution of Population over five equally sized bins in Market data.

Figure 1.6 shows two traditional displays for the distribution: a *histogram* (part (a) on the left) in which bars are proportional to the group sizes and a *pie-chart* in which a pie is partitioned into slices proportional to the cluster sizes (part (b) on the right). These two point to different features of the distribution. The histogram positions the categories along the horizontal axis, thus providing for a possibility to see the distribution's shape, which can be quite useful when the categories have been created as interval bins of a quantitative feature, as is this case. The pie-chart points to the fact that the group sizes sum up to the total so that one can see what portions of the pie account for different categories.

One-dimensional data within groups

To visualize a quantitative feature within pre-specified groups, *box-plots* and *stick-plots* are utilized. They show within-cluster central values and their dis-

Table 1.13: **Population groups:** Data of the distribution of population groups and numbers of banks within them.

Group	Size	Frequency, %	Banks	Std Banks
I	25	55.6	1.80	1.62
II	11	24.4	4.82	2.48
III	2	4.4	5.00	1.00
IV	3	6.7	12.00	5.72
V	4	8.9	12.50	1.12
Total	45	100	4.31	4.35

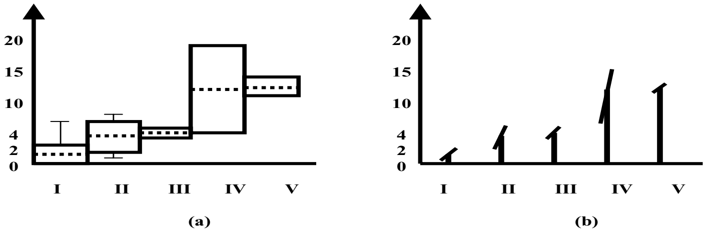


Figure 1.7: Box-plot (a) and stick-plot (b) presenting the feature Bank over the five bins in Market data.

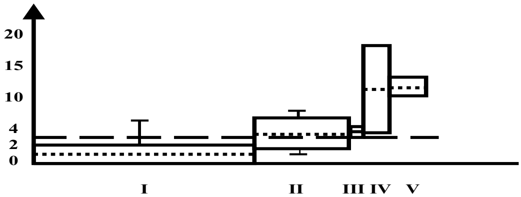


Figure 1.8: Box-plot presenting the feature Bank over five bins in Market data along with bin sizes.

persion, which can be done in different ways. Figure 1.7 presents a box-plot (a) and stick-plot (b) of feature Bank within the five groups defined above as Population bins. The box-plot on Figure 1.7 (a) represents each group as a box bounded by its 10% percentile values separating extreme 10% cases both on the top and bottom of the feature Bank range. The real within group ranges are shown by "whiskers" that can be seen above and below the boxes of groups I and II; the other groups have no whiskers because of too few entities in each of them. A line within each box shows the within-group average. The stick-plot on Figure 1.7 (b) represents the within-group averages by "sticks," with their "whiskers" proportional to the standard deviations.

Since the displays are in fact two-dimensional, both features and distributions can be shown on a box-plot simultaneously. Figure 1.8 presents a box-plot of the feature Bank over the five bins with the box widths made proportional to the group sizes. This time, the grand mean is also shown by the horizontal dashed line.

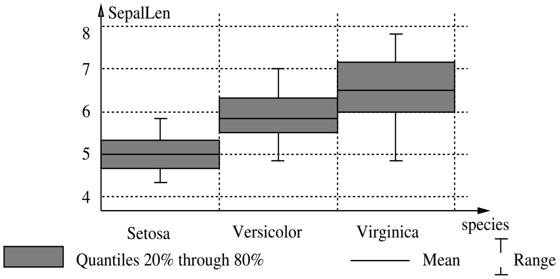


Figure 1.9: Box-plot of three classes of Iris specimens from Table 1.6 over the sepal length w1; the classes are presented by both the percentile boxes and within cluster range whiskers; the choice of percentiles can be adjusted by the user.

A similar box-plot for the three genera in the Iris data is presented in Figure 1.9. This time the percentiles are taken at 20%.

Two-dimensional display

A traditional two-dimensional display of this type is the so-called *scatter-plot*, representing all the entity points in a plane generated by two of the variables or linear combinations of the variables such as principal components (for a definition of principal components see section 5.1.3). A scatter-plot at the plane of two variables can be seen in Figure 1.2 for the Body mass data on page 13. A scatter-plot in the space of two first principal components is presented in Figure 1.10: the Iris specimens are labelled by the class number (1, 2, or 3); centroids are gray circles; the most deviate entities (30 in class 1, 32 in class 2, and 39 in class 3) are shown in boxes. For an explanation of the principal components see section 5.1.3. The scatter-plot illustrates that two of the classes are somewhat interwoven.

Block-structure

A block-structure is a representation of the data table as organized in larger blocks of a specific pattern with transpositions of rows and/or columns. In principle one can imagine various block patterns [47], of which the most common is a pattern formed by the largest entry values.

Figure 1.11 presents an illustrative example (described in [125]). In part A, results of seven treatments (denoted by letters from a to g) applied to each of ten crops denoted by numerals are presented: gray represents a success and blank space failure. The pattern of gray seems rather chaotic in table A. However, it becomes very much orderly when appropriate rearrangements of rows and

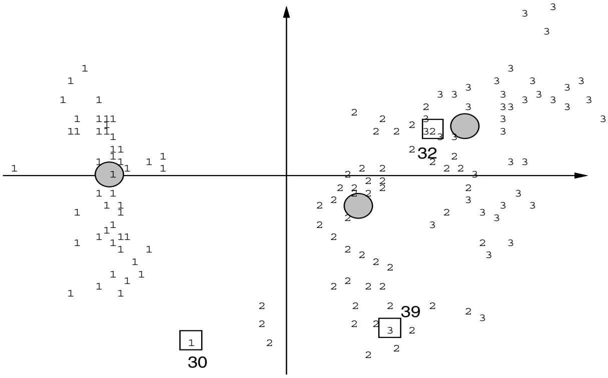
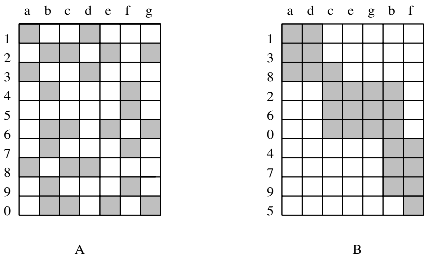


Figure 1.10: Scatter-plot of Iris specimens in the plane of the first two principal components.



	a	b	c	d	e	f	g
1	▨			▨			
2		▨	▨		▨		▨
3	▨			▨			
4		▨				▨	
5						▨	
6		▨	▨		▨		▨
7		▨				▨	
8	▨		▨	▨			
9		▨				▨	
0		▨	▨		▨		▨

A

	a	d	c	e	g	b	f
1	▨	▨					
3	▨	▨					
8	▨	▨	▨				
2			▨	▨	▨	▨	
6			▨	▨	▨	▨	
0			▨	▨	▨	▨	
4						▨	▨
7						▨	▨
9						▨	▨
5							▨

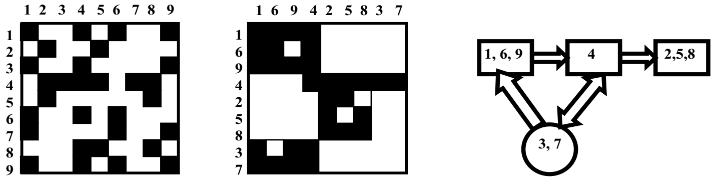
B

Figure 1.11: Visibility of the matrix block structure with a rearrangement of rows and columns.

columns are performed. Part B of the Figure clearly demonstrates a visible block structure in the matrix, that can be interpreted as mapping specific sets of treatments to different sets of crops, which can be exploited, for instance, in specifying adjacent locations for crops.

Visualization of block structures by reordering rows and/or columns is popular in the analysis of gene expression data [20] and ecology [53].

A somewhat more realistic example is shown in Figure 1.12 (a) representing a matrix of value transferred between nine industries during a year: the (i,j)-th entry is gray if the transfer from industry i to industry j is greater than a specified threshold and blank otherwise. Figure 1.12 (b) shows a block structure pattern that becomes visible when the order of industries from 1 to 9 changes for the order 1-6-9-4-2-5-8-3-7, which is achieved with the reordering of both rows and columns of the matrix. The reordering is made simultaneously on both rows and columns because both represent the same industries, both as sources (rows) and targets (columns) of the value transfer. We can discern four blocks of different patterns (1-6-9, 4, 2-5-8, 3-7) in Figure 1.12 (b). The structure of the transfers between the blocks can be captured in a graph presented in Figure 1.12 (c).



(a) (b) (c)

Figure 1.12: Value transfer matrix presented with only entries greater than a threshold (a); the same matrix, with rows and columns simultaneously reordered, is in (b); in (c), the structure is represented as a graph.

Structure

A simple structure such as a chain or a tree or just a small graph, whose vertices (nodes) correspond to clusters and edges to associations between them, is a frequent tool in data visualization.

Two examples are presented in Figure 1.13: a tree structure over clusters reflecting common origin is shown in part (a) and a graph corresponding to the block structure of Figure 1.12 (c) is shown in part (b) to reflect links between clusters of industries in the production process.

A similar tree structure is presented on Figure 1.5 on page 16 illustrating a classification tree for Digits. Tree leaves, the terminal boxes, show clusters as entity sets; the features are shown along corresponding branches; the entire structure illustrates the relation between clusters in such a way that any combination of the segments can be immediately identified and placed into a corresponding cluster or not identified at all if it is not shown on the tree.

Visualization using an inherent topology

In many cases the entities come from an image themselves – such as in the cases of analysis of satellite images or topographic objects. For example, consider the Digit data set: all the integer symbols are associated with segments of the generating rectangle on Figure 1.3, page 13. Clusters of such entities can be visualized with the generating image.

Figure 1.4 visualizes clusters of digits along with their defining features resulting from analyses conducted later in example 4.43 (page 134) as parts of the generating rectangle. There are four major confusion clusters in the

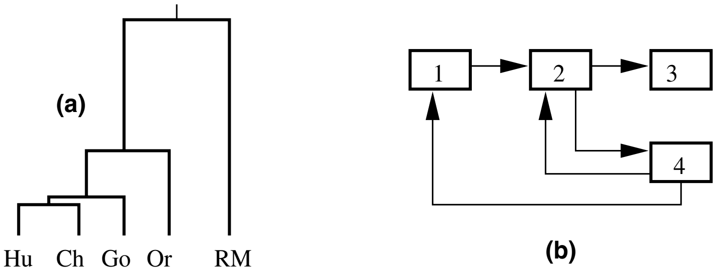


Figure 1.13: Visual representation of relations between clusters: (a) the evolutionary structure of the Primates genera according to distances in Table 1.2; (b) interrelation between clusters of industries according to Figure 1.12 (c).

Digits data of Figure 1.3 that are presented with distinctive features shown with segments defining the drawing of digits.

1.2 Bird's-eye view

This section contains general remarks on clustering and can be skipped on the first reading.

1.2.1 Definition: data and cluster structure

After looking through the series of exemplary problems in the previous section, we can give a more formal definition of clustering than that in the Preface: Clustering is a discipline devoted to revealing and describing cluster structures in data sets.

To animate this definition, one needs to specify the four concepts involved:
(a) **data**,
(b) **cluster structure**,
(c) **revealing a cluster structure**,
(d) **describing a cluster structure**.

Data

The concept of data refers to any recorded and stored information such as satellite images or time series of prices of certain stocks or survey questionnaires filled in by respondents. Two types of information are associated with data: the data entries themselves, e.g., recorded prices or answers to questions, and meta-data, that is, legends to rows and columns giving meaning to entries. The aspect of developing and maintaining databases of records, taking into account the relations stored in metadata, is very important for data mining [23, 29, 44].

In this text, for the sake of linearity of presentation, we concentrate on a generic data format only, the so-called entity-to-variable table whose entries represent values of pre-specified variables at pre-specified entities.

The variables are synonymously called attributes, features, characteristics, characters and parameters. Such words as case, object, observation, instance, record are in use as synonyms to the term, entity, accepted here.

The data table format of data often arises directly from experiments or observations, from surveys, and from industrial or governmental statistics. This also is a conventional form for presenting database records. Other data types such as signals or images can be modelled in this format, too, via digitalized representation. However, a digital representation, typically, involves much more information than can be kept in a data table format. Especially important is the spatial arrangement of pixels, which is not, typically, maintained in the concept of data table. The contents of a data table are assumed to be invariant under permutations of rows and columns and corresponding metadata.

Another data type traditionally considered in clustering is the similarity or dissimilarity between entities (or features). The concept of similarity is most important in clustering: similar objects are to be put into the same cluster and dissimilar into different clusters. There have been invented dozens of (dis)similarity indices. Some of them nicely fit into theoretical frameworks and will be considered further in the text.

One more data type considered in this text is co-occurrence or flow tables that represent the same substance distributed between different categories such as Confusion data in Table 1.9 in which the substance is the scores of individuals. An important property of this type of data is that any part of the data table, referred to a subset of rows and /or a subset of columns, can be meaningfully aggregated by summing the part of the total flow within the subset of rows (and/or the subset of columns). The sums represent the total flow to, from, and within the subset(s). Thus, problems of clustering and aggregating are naturally linked here. Until recently, this type of data appeared as a result of data analysis rather than input to it. Currently it has become one of the major data formats. Examples are: distributions of households purchasing various commodities or services across postal districts or other regional units, counts of telephone calls across areas, and counts of visits in various categories of web-sites.

Cluster structure

The concept of **cluster** typically refers to a set of entities that is cohesive in such a way that entities within are more similar to each other than to the outer entities.

Three major types of cluster structures are: (a) a single cluster considered against the rest or whole of the data, (b) a partition of the entity set in a set of clusters, and (c) a (nested) hierarchy of clusters.

Of these three, partition is the most conventional, probably because it is relevant to both science and management, the major forces behind scientific developments. A scientist, as well as a manager, wants unequivocal control over the entire universe under consideration. This is why they may wish to partition the entity set into a set of nonoverlapping clusters.

In some situations there is no need for total clustering. The user may be quite satisfied with getting just a single (or few) cluster(s) and leaving the rest completely unclustered. Examples:

(1) a bank manager wants to learn how to discern potential fraudsters from other clients or

(2) a marketing researcher separates a segment of customers prone to purchase a particular product or

(3) a bioinformatician seeks a set of proteins homologous to a query protein sequence.

Incomplete clustering is a recently recognized addition to the body of clustering approaches, very suitable not only at the situations above but also as a tool for conventional partitioning via cluster-by-cluster procedures such as those described in section 5.5.

The hierarchy is the oldest and probably least understood of the cluster structures. To see how important it is, it should suffice to recall that the Aristotelian approach to classification encapsulated in library classifications and biological taxonomies is always based on hierarchies. Moreover, hierarchy underlies most advanced data processing tools such as wavelets and quadtrees. It is ironic then that as a cluster structure in its own right, the concept of hierarchy rarely features in clustering, especially when clustering is confined to the cohesive partitioning of geometric points.

1.2.2 Criteria for revealing a cluster structure

To **reveal a cluster structure** in a data table means to find such clusters that allow the individual characteristics of entities to be substituted by aggregate characteristics of clusters. A cluster structure is revealed by a *method* according to a *criterion* of how well the data are represented by clusters. Criteria and methods are, to an extent, independent from each other so that the same method such as agglomeration or splitting can be used with different criteria.

Criteria usually are formulated in terms of (dis)similarity between entities. This helps in formalizing the major idea that entities within clusters should be similar to each other and those between clusters dissimilar. Dozens of similarity based criteria developed so far can be categorized in three broad classes:

(1) Definition-based,

(2) Index-based, and

(3) Computation-based.

The first category comprises methods for finding clusters according to an explicit definition of a cluster. An example: A cluster is a subset S of entities such that for all i, j in S the similarity between i and j is greater than the similarities between these and any k outside S. Such a property must hold for all entities with no exceptions, which means that well isolated clusters are rather rare in real world data. However, when the definition of cluster is relaxed to include less isolated clusters, too many may then appear. This is why definition-based methods are not popular in practical clustering.

A criterion in the next category involves an index, that is, a numerical function that scores different cluster structures and, in this way, may guide the process of choosing the best. However, not all indices are suitable for obtaining reasonable clusters. Those derived from certain model-based considerations tend to be computationally hard to optimize. Optimizing methods are thus bound to be local and, therefore, heavily reliant on the initial settings, which involve, in the case of K-Means clustering, pre-specifying the number of clusters and the location of their central points. Accordingly, the found cluster structure may be rather far from the global optimum and, thus, must be *validated.* Cluster validation may be done according to internal criteria such as that involved in the optimization process or external criteria comparing the clusters found with those known from external considerations or according to its stability with respect to randomly resampling entities/features. These will be outlined in section 7.5 and exemplified in section 3.3.2.

The third category comprises computation methods involving various heuristics for individual entities to be added to or removed from clusters, for merging or splitting clusters, and so on. Since operations of this type are necessarily local, they resemble local search optimization algorithms, though, typically, have no unique guiding scoring index to follow, thus, can include various tricks making them flexible. However, such flexibility is associated with an increase in the number of ad hoc parameters such as various similarity thresholds and, in this way, turning clustering from a reproducible activity into a kind of magic. Validation of a cluster structure found with a heuristic-based algorithm becomes a necessity.

In this book, we adhere to an index-based principle, which scores a cluster structure against the data from which it has been built. The cluster structure here is used as a device for reconstructing the original data table; the closer the reconstructed data are to the original ones, the better the structure. It is this principle that is called the *data recovery approach* in this book. Many index-based and computation-based clustering methods can be reinterpreted according to the principle, which allows us to see interrelations between different methods and concepts for revealing and analyzing clustering structures. New methods can be derived from the principle too (see Chapter 5, especially sections 5.4-5.6). It should be noted, though, that we will use only the most straightforward rules for reconstructing the data from cluster structures.

1.2.3 Three types of cluster description

Cluster descriptions help in understanding, explaining and predicting clusters. These may come in different formats of which the most popular are the following three: (a) Representative, (b) Tendency, (c) Conceptual description.

A *representative*, or a *prototype*, is an object such as a literary character or a sort of wine or mineral, representing the most typical features of a cluster. This format is useful in giving a meaning to entities that are easily available empirically but difficult to conceptually describe. There is evidence that some aggregate language constructs, such as “fruit,” are mentally maintained via prototypes, such as “apple” [74]. In clustering, the representative is usually the most central entity in a cluster.

A *tendency* expresses a cluster’s most likely features such as its way of behavior or pattern. It is usually related to the center of gravity of the cluster and its differences from the average. In this respect, the tendency models the concept of type in classification studies.

A *conceptual description* may come in the form of a classification tree built for predicting a class or partition. Another form of conceptual description is an *association*, or *production, rule*, stating that if an object belongs to a cluster then it must have such and such features. Or, vice versa, if an object satisfies the premise, then it belongs in the cluster. The simplest conceptual description of a cluster is a statement of the form “the cluster is characterized by the feature A being between values a1 and a2.” The existence of a feature A, which alone is sufficient to distinctively describe a cluster is a rare occurrence of luck in data mining. Typically, features in data are rather superficial and do not express essential properties of entities and thus cannot be the basis of straightforward descriptions.

The subject of cluster description overlaps that of supervised machine learning and pattern recognition. Indeed, given a cluster, having its description may allow one to predict, for new objects, whether they belong to the cluster or not, depending on how much they satisfy the description. On the other hand, a decision rule obtained with a machine learning procedure, especially, for example, a classsification tree, can be considered a cluster description usable for the interpretation purposes. Still the goals are different: interpretation in clustering and prediction in machine learning. However, cluster description is as important in clustering as cluster finding.

1.2.4 Stages of a clustering application

Typically, clustering as a data mining activity involves the following five stages:

A. Developing a data set.

B. Data pre-processing and standardizing.

C. Finding clusters in data.

D. Interpretation of clusters.

E. Drawing conclusions.

To develop a data set one needs to define a substantive problem or issue, however vague it may be, and then determine what data set related to the issue can be collected from an existing database or set of experiments or survey, etc.

Data pre-processing is the stage of preparing data processing by a clustering algorithm; typically, it includes developing a uniform data set, frequently called a 'flat' file, from a database, checking for missing and unreliable entries, rescaling and standardizing variables, deriving a unified similarity measure, etc.

The cluster finding stage involves application of a clustering algorithm and results in a (series of) cluster structure(s) to be presented, along with interpretation aids, to substantive specialists for an expert judgement and interpretation in terms of features, both those utilized for clustering (internal features) and those not utilized (external features). At this stage, the expert may see no relevance in the results and suggest a modification of the data by adding/removing features and/or entities. The modified data is subject to the same processing procedure. The final stage is the drawing of conclusions, with respect to the issue in question, from the interpretation of the results. The more focussed are the regularities implied by the findings, the better the quality of conclusions.

There is a commonly held opinion among specialists in data analysis that the discipline of clustering concerns only the proper clustering stage C while the other four are the concern of specialists in the substance of the particular issue for which clustering is performed. Indeed, typically, clustering results can not and are not supposed to solve the entire substantive problem, but rather relate to an aspect of it.

On the other hand, clustering algorithms are supposedly most applicable to situations and issues in which the user's knowledge of the domain is more superficial than profound. What are the choices regarding data pre-processing, initial settings in clustering and interpretation of results – facing the laymen user who has an embryonic knowledge of the domain? More studies and experiments? In most cases, this is not practical advice. Sometimes a more viable strategy would be to better utilize properties of the clustering methods at hand.

At this stage, no model-based recommendations can be made about the initial and final stages, A and E. However, the data recovery approach does allow us to use the same formalisms for tackling not stage C only, but also B and D; see sections 2.4, 4.3 and 6.3 for related prescriptions and discussions.

1.2.5 Clustering and other disciplines

The concepts involved make clustering a multidisciplinary activity on its own, regardless of its many applications. In particular,

1. **Data** relates to database, data structure, measurement, similarity and dissimilarity, statistics, matrix theory, metric and linear spaces, graphs, data analysis, data mining, etc.

2. **Cluster structure** relates to discrete mathematics, abstract algebra, cognitive science, graph theory, etc.

3. **Revealing** cluster structures relates to algorithms, matrix analysis, optimization, computational geometry, etc.

4. **Describing** clusters relates to machine learning, pattern recognition, mathematical logic, knowledge discovery, etc.

1.2.6 Different perspectives of clustering

Clustering is a discipline on the intersection of different fields and can be viewed from different angles, which may be sometimes confusing because different perspectives may contradict each other. A question such as, “How many clusters are out there?,” which is legitimate in one perspective, can be meaningless in the other. Similarly, the issue of validation of clusters may have different solutions in different frameworks. The author finds it useful to distinguish between the perspectives supplied by statistics, machine learning, data mining and classification.

Statistics perspective

Statistics tends to view any data table as a sample from a probability distribution whose properties or parameters are to be estimated with the data. In the case of clustering, clusters are supposed to be associated with different probabilistic distributions which are intermixed in the data and should be recovered from it.

Within this approach, such questions as “How many clusters are out there?” and “How to preprocess the data?” are well substantiated and can be dealt with according to the assumptions of the underlying model.

In many cases the statistical paradigm suits quite well and should be applied as the one corresponding most to what is called the scientific method: make a hypothesis of the phenomenon in question, then look for relevant data and check how the hypothesis fits them.

A trouble with this approach is that in most cases clustering is applied to phenomena of which almost nothing is known, not only of their underlying mechanisms but of the very features measured or to be measured. Then any modelling assumptions of the data generation would be necessarily rather arbitrary and so too conclusions based on them.

Moreover in many cases the set of entities is rather unique and cannot be considered a sample from a larger population, such as the set of European countries or single malt whisky brands.

Sometimes the very concept of a cluster as a probabilistic distribution seems to not fit into a clustering goal. Look, for example, at a bell-shaped Gaussian distribution which is considered a good approximation for such variables as the height or weight of young male individuals of the same ethnicity so that they form a cluster corresponding to the distribution. However, when confronted with the practical issue of dividing people, for example, according to their fighting capabilities (such as in military conscription or in the sport of boxing), the set cannot be considered a homogeneous cluster anymore and must be further partitioned into more homogeneous strata. Some say that there must be a boundary between "natural" clusters and clusters to be drawn on purpose; that a bell-shape distribution corresponds to a natural cluster and a boxing weight category to an artificial one. However, it is not always easy to distinguish which situation is which. There will always be situations when a cluster of potentially weak fighters (or bad customers, or homologous proteins) must be cut out from the rest.

Machine learning perspective

Machine learning tends to view the data as a device for learning how to predict pre-specified or newly created categories. The entities are considered as coming one at a time so that the machine can learn adaptively in a supervised manner. To theorize, the flow of data must be assumed to come from a probabilistic population, an assumption which has much in common with the statistics approach. However, it is prediction rather than model fitting which is the central issue in machine learning.

Such a shift in the perspective has led to the development of strategies for predicting categories such as decision trees and support vector machines as well as resampling methods such as the bootstrap and cross-validation for dealing with limited data sets.

Data mining perspective

Data mining is not much interested in reflection on where the data have come from nor how they have been collected. It is assumed that a data set or database has been collected already and, however bad or well it reflects the properties of the phenomenon in question, the major concern is in finding patterns and regularities within the data as they are. Machine learning and statistics methods are welcome here – for their capacity to do the job.

This view, started as early as in the sixties and seventies in many countries including France, Russia and Japan in such subjects as analysis of questionnaires or of inter-industrial transfers, was becoming more and more visible, but it did not make it into prominence until the nineties. By that time, big warehouse databases became available, which led to the discovery of patterns of transactions with the so-called association search methods. The patterns proved themselves correct when superstores increased profits by accommodating to them.

Data mining is a huge activity on the intersection of databases and data analysis methods. Clustering is a recognized part of it. The data recovery approach which is maintained in this book obviously fits within data mining very well, because it is based only on the data available.

It should be added that the change of the paradigm from modeling of mechanisms of data generation to data mining has drastically changed requirements to methods and programs. According to the statistics approach, the user must know the models and methods he uses; if a method is applied wrongly, the results can be wrong too. Thus, application of statistical methods is limited within a small circle of experts. In data mining, it is the patterns not methods that matter. This shifts the focus of computer programs from statistics to the user's substantive area and makes them user-friendly.

Similarly, the validation objectives seem to diverge here: in statistics and machine learning the stress goes on the consistency of the algorithms, which is not quite so important in data mining, in which it is the consistency of patterns, not algorithms, which matters the most.

Classification/knowledge-discovery perspective

The classification perspective is rarely discussed indeed. In data mining the term "classification" is usually referred to in a very limited sense: as an activity of assigning prespecified categories (classes) to entities, in contrast to clustering which assigns entities with newly created categories (clusters).

According to its genuine meaning, classification is an actual or ideal arrangement of entities under consideration in classes to:

(1) shape and keep knowledge;

(2) capture the structure of phenomena; and

(3) relate different aspects of a phenomenon in question to each other.

These make the concept of classification a specific mechanism for knowledge discovery and maintenance. Consider, for instance, the Periodic Chart of chemical elements. Its rows correspond to numbers of electron shells in the atoms, and its columns to the numbers of electrons in the external shell thus capturing the structure of the phenomenon. These also relate to most important physical properties and chemical activities of the elements thus associating different aspects of the phenomenon. And this is a compact form of representing the knowledge; moreover, historically it is this form itself, developed rather empirically, that made possible rather fast progress to the current theories of the matter.

In spite of the fact that the notion of classification as part of scientific knowledge was introduced by the ancient Greeks (Aristotle and the like) the very term "classification" seems a missing item in the vocabulary of current scientific discourse. This may have happened because in traditional sciences, classifications are defined within well developed substantive theories according to variables which are defined as such within the theories. Thus, there has been no need in specific theories for classification.

Clustering should be considered as classification based on empirical data in a situation when clear theoretical concepts and definitions are absent and the regularities are unknown. Thus, the clustering goals should relate to the classification goals above. This brings one more aspect to clustering. Consider, for example, how one can judge whether a clustering is good or bad? According to the classification/knowledge-discovery view, this is easy and has nothing to do with statistics: just look at how well clusters fit within the existing knowledge, how well they allow updating, correcting and extending.

Somewhat simplistically, one might say that two of the points stressed in this book, that of the data recovery approach and the need to not only find, but describe clusters, fit well into the two perspectives, the former into data mining and the latter into classification as knowledge discovery.

Chapter 2

What Is Data

After reading through this chapter, the reader will know of:

1. Three types of data tables: (a) feature-to-entity, (b) similarity/dissimilarity and (c) contingency/flow tables, and ways to standardize them.
2. Quantitative, categorical and mixed data, and ways to pre-process and standardize them.
3. Characteristics of feature spread and centrality.
4. Bi-variate distributions over mixed data, correlation and association, and their characteristics.
5. Visualization of association in contingency tables with Quetelet coefficients.
6. Multidimensional concepts of distance and inner product.
7. The concept of data scatter.

Base words

Average The average value of a feature over a subset of entities. If the feature is binary and corresponds to a category, the average is the category frequency in the subset. The average over the entire entity set is referred to as a grand mean.

Contingency coefficient A summary index of statistical association between two sets of categories in a contingency table. The greater it is, the closer the association to a conceptual one.

Contingency table Given two sets of categories corresponding to rows and columns, respectively, this table presents counts of entities co-occurring at the intersection of each pair of categories from the two sets. When categories within each of the sets are mutually disjoint, the contingency table can be aggregated by summing up relevant entries.

Correlation The shape of a scatter-plot showing the extent to which two features can be considered mutually related. The (product-moment) correlation coefficient captures the extent at which one of the features can be expressed as a linear function of the other.

Data scatter The sum of squared entries of the data matrix; it is equal to the sum of feature contributions or the summary distance from entities to zero.

Data table Also referred to as *flat file* (in databases) or *vector space data* (in information retrieval), this is a two-dimensional array whose rows correspond to entities, columns to features, and entries to feature values at entities.

Distance Given two vectors of the same size, the (Euclidean squared) distance is the sum of squared differences of corresponding components, $d(x,y) = \sum_i (x_i - y_i)^2$. It is closely related to the inner product: $d(x,y) = (x-y, x-y)$.

Entity Also referred to as *observation* (in statistics) or *case* (in social sciences) or *instance* (in artificial intelligence) or *object*, this is the main item of clustering corresponding to a data table row.

Feature Also referred to as *variable* (in statistics) or *character* (in biology) or *attribute* (in logic), this is another major data item corresponding to a data table column. It is assumed that feature values can be compared to each other, at least, whether they coincide or not (categorical features), or even averaged over any subset of entities (quantitative feature case).

Inner product Given two vectors of the same size, the inner product is the sum of products of corresponding components, $(x,y) = \sum_i x_i y_i$. It is closely related to the distance: $d(x,y) = (x,x) + (y,y) - 2(x,y)$.

Quetelet index In contingency tables: A value showing the change in frequency of a row category when a column category becomes known. The greater the value, the greater the association between the column and row categories. It is a basic concept in contingency table analysis.

Range The interval in which a feature takes its values; the difference between the feature maximum and minimum over a data set.

Scatter plot A graph presenting entities as points on the plane formed by two quantitative features.

Variance The average of squared deviations of feature values from the average.

2.1 Feature characteristics

2.1.1 Feature scale types

The Masterpieces data in Table 1.10 will be used to illustrate data handling concepts in this section. For the reader's convenience, the table is reprinted here as Table 2.1.

A data table of this type represents a unity of the set of rows, always denoted as I further on, the set of columns denoted by V and the table contents X, the set of values x_{iv} in rows $i \in I$ and columns $v \in V$. The number of rows, or cardinality of I, $|I|$, will be denoted by N, and the number of columns, the cardinality of V, $|V|$, by M. Rows will always correspond to entities, columns to features. Whatever metadata of entities may be known, are all to be put as the features, except for names, that may be maintained as a list associated with I. As to the features $v \in V$, it is assumed that each has a measurement scale assigned to it, and of course a name.

All within-column entries are supposed to have been measured in the same scale and thus comparable within the scale; this is not so over rows in Y. Three different types of scales that are present in Table 2.1 and will be dealt with in the remainder are quantitative (LenSent, LenDial, and NChar), nominal (Narrative) and binary (SCon). Let us elaborate on these scale types:

Table 2.1: **Masterpieces:** Masterpieces of 19th century: the first three by Charles Dickens (1812–1870), the next three by Mark Twain (1835–1910), and the last two by Leo Tolstoy (1828–1910).

Title	LenSent	LenDial	NChar	SCon	Narrative
Oliver Twist	19.0	43.7	2	No	Objective
Dombey and Son	29.4	36.0	3	No	Objective
Great Expectations	23.9	38.0	3	No	Personal
Tom Sawyer	18.4	27.9	2	Yes	Objective
Huckleberry Finn	25.7	22.3	3	Yes	Personal
Yankee at King Arthur	12.1	16.9	2	Yes	Personal
War and Peace	23.9	30.2	4	Yes	Direct
Anna Karenina	27.2	58.0	5	Yes	Direct

1. **Quantitative**: A feature is quantitative if the operation of taking its average is meaningful.

   It is quite meaningful to compare the average values of feature LenS or LenD for different authors in Table 2.1. Somewhat less convincing is the case of NumC which must be an integer; some authors even consider such "counting" features a different scale type. Still, we can safely say that on average Tolstoy's novels have larger numbers of principal characters than those by Dickens or Twain. This is why counting features are also considered quantitative in this text.

2. **Nominal**: A categorical feature is said to be nominal if its categories are (i) disjoint, that is, no entity can fall in more than one of them, and (ii) not ordered, that is, they only can be compared with respect to whether they coincide or not. Narrative, in Table 2.1, is such a feature.

   Categorical features maintaining (ii) but not (i) are referred to as multi-choice variables. For instance, Masterpieces data might include a feature that presents a list of social themes raised in a novel, which may contain more than one element. That would produce a one-to-many mapping of the entities to the categories, that is, social themes. There is no problem in treating this type of data within the framework described here. For instance, the Digit data table may be treated as that representing the only, multi-choice, variable "Segment" which has the set of seven segments as its categories.

   Categorical features that maintain (i) but have their categories ordered are called rank variables. Variable Bribe level in the Bribery data of Tables 1.11 and 1.12 is rank: its three categories are obviously ordered according to the bribe size. Traditionally, it is assumed for the rank variables that only the order of categories matters and intervals between them are irrelevant. That is, rank categories may accept any quantitative coding which is compatible with their ordering. This makes rank features difficult to deal with in the context of mixed data tables. We maintain a different view, going back to C. Spearman: the ranks are treated as numerical values and the rank variables are considered thus quantitative and processed accordingly. In particular, seven of the eleven variables in Bribery data (II. Client, IV. Occurrence, V. Initiator, VI. Bribe, VII. Type, VIII. Network, and XI. Punishment) will be considered ranked with ranks assigned in Table 1.11 and treated as quantitative values.

   There are two approaches to the issue of involving qualitative features into analysis. According to one, more traditional, approach, categorical variables are considered non-treatable quantitatively. The only quantitative operation admitted for categories is counting the number or frequency of its occurrences at various subsets. To conduct cluster analysis, categorical data, according to this view, can only be utilized for deriving an entity-to-entity (dis)similarity measure. Then this measure can be used for finding clusters.

   A different approach is maintained and further developed here: a category defines a quantitative zero-one variable on entities, with one corresponding to its presence and zero absence, which is treated then as such. We will see later that this view, in fact, does not contradict the former one but rather fits into it with geometrically and statistically sound specifications.

3. **Binary**: A qualitative feature is said to be binary if it has two categories which can be thought of as Yes or No answer to a question such as feature SCon in Table 2.1. A two-category feature can be considered either a nominal or binary one, depending on the context. For instance, feature “Gender” of a human should be considered a nominal feature, whereas the question “Are you female?” a binary feature, because the latter assumes that it is the “female,” not “male,” category which is of interest. Operationally, the difference between these two types will amount to how many binary features should be introduced to represent the feature under consideration in full. Feature “Gender” cannot be represented by one column with Yes or No categories: two are needed, one for “Female” and one for “Male.”

2.1.2 Quantitative case

As mentioned, we consider that the meaningfulness of taking the average is a defining property of a quantitative variable. Given a feature $v \in V$ whose values y_{iv}, $i \in I$, constitute a column in the data table, its *average* over entity subset $S \subseteq I$ is defined by the formula

$$c_v(S) = (1/N_S)\sum_{i \in S} y_{iv} \tag{2.1}$$

where N_S is the number of entities in S.

The average $c_v = c_v(I)$ of $v \in V$ over the entire set I is sometimes referred to as *grand mean*. After grand mean c_v of $v \in V$ has been subtracted from all elements of the column-feature $v \in V$, the grand mean of v becomes zero. Such a variable is referred to as *centered*.

It should be mentioned that usually the quantitative scale is defined somewhat differently, not in terms of the average but the so-called admissible transformations $y = \phi(x)$. The scale type is claimed to depend on the set of transformations ϕ which are considered admissible, that is, do not change the scale contents. For the quantitative feature scales, those that are admissible are transformations such as $y = ax + b$ converting all x values into y values by changing the scale factor a times and shifting the scale origin at b. Transformations $y = \phi(x)$ of this type, with $\phi(x) = ax+b$ for some real a and b, are referred to as affine transformations. For instance, the temperature Celsius scale x is transformed into the temperature Fahrenheit scale with $\phi(x) = 1.8x+32$. Standardizations of data with affine transformations are at the heart of our approach to clustering.

Our definition is compatible with the one given above. Indeed, if a feature x admits affine transformations, it is meaningful to compare its average values over various entity sets. Let x_J and x_K be the averages of sets $\{x_j : j \in J\}$ and $\{x_k : k \in K\}$ respectively, and, say, $x_J \leq x_K$. Does the same automatically hold for the averages of $y = ax + b$ over J and K? To answer this question, we consider values $y_j = ax_j + b$, $j \in J$, and $y_k = ax_k + b$, $k \in K$ and calculate their averages, y_J and y_K. It is easy to prove that $y_K = ax_K + b$ and $y_J = ax_J + b$ so that any relation between x_J and x_K remains the same for y_J and y_K, up to the obvious reversal when a is negative (which means that rescaling involves change of the direction of the scale).

Other indices of "centrality" have been considered too; the most popular of them are:

i Midrange, point in the middle of the range, that is, equi-distant from the minimum and maximum values of the feature.

ii Median, the middle item in the series of elements of column v sorted in ascending (or descending) order.

iii Mode, "the most likely" value, which is operationally defined by partitioning the feature range in a number of bins (intervals of the same size) and determining at which of the bins the number of observations is maximum: the center of this bin is the mode, up to the error related to the bin size.

Each of these has its advantages and drawbacks as a centrality measure. The median is the most stable with regard to change in the sample and, especially, to the presence of outliers. Outliers can drastically change the average, and they do not affect the median at all. However, the calculation of the median requires sorting the entity set, which sometimes may be costly. Midrange is insensitive to the shape of the distribution and is highly sensitive to outliers. The mode is of interest when distribution of the feature is far from uniform.

These may give a hint with respect to what measure should be used in a specific situation. For example, if the data to be analyzed have no specific properties at all, the average should be utilized. When outliers or data errors are expected, the median would be a better bet.

The average, median and midrange all fit within the following approximation model which is at the heart of the data recovery approach. Given a number of reals, x_1, x_2,..., x_N, find a unique real a that can be used as their aggregate substitute so that for each i, a approximates x_i up to a residual ϵ_i: $x_i = a + \epsilon_i$, $i \in I$. The smaller the residuals the better the aggregate. To minimize the residuals $\epsilon_i = x_i - a$, they should be combined into a scalar criterion such

Table 2.2: Summary characteristics of the Market town data.

	P	PS	Do	Ho	Ba	Su	Pe	DIY	SP	PO	CAB	FM
Mean	7351.4	3.0	1.4	0.4	4.3	1.9	2.0	0.2	0.5	2.6	0.6	0.2
Std	6193.2	2.7	1.3	0.6	4.4	1.7	1.6	0.6	0.6	2.1	0.6	0.4
Range	21761.0	12.0	4.0	2.0	19.0	7.0	7.0	3.0	2.0	8.0	2.0	1.0

as $L_1 = \sum_i |x_i - a|$, $L_\infty = \max_i |x_i - a|$, or $L_2 = \sum_i |x_i - a|^2$. It appears, L_1 is minimized by the median, L_∞ by midrange and L_2 by the average. The average fits best because it solves the least squares approximation problem, and the least-square criterion is the basis of all further developments in Chapter 5.

A number of characteristics have been defined to measure the features' dispersion or spread. Probably the simplest of them is the variable's *range*, the difference between its maximal and minimal values, that has been mentioned above already. This measure should be used cautiously as it may be overly sensitive to changes in the entity set. For instance, removal of Truro from the set of entities in the Market town data immediately reduces the range of variable Banks to 14 from 19. Further removal of St Blazey/Par further reduces the range to 13. Moreover, the range of variable DIY shrinks to 1 from 3, with these two towns removed. Obviously, no such drastic changes emerge when all thirteen hundred of the English Market towns are present.

A somewhat more elaborate characteristic of dispersion is the so-called (empirical) variance of $v \in V$ which is defined as

$$s_v^2 = \sum_{i \in I} (y_{iv} - c_v)^2 / N \tag{2.2}$$

where c_v is the grand mean. That is, s_v^2 is the average squared deviation L_2 of y_{iv} from c_v.

The standard deviation of $v \in V$ is defined as just $s_v = \sqrt{s_v^2}$ which has also a statistical meaning as the square-average deviation of the variable's values from its grand mean. The standard deviation is zero, $s = 0$, if and only if the variable is constant, that is, all the entries are equal to each other.

In some packages, especially statistical ones, denominator $N - 1$ is used instead of N in definition (2.2) because of probabilistic consistency considerations (see any text on mathematical statistics). This shouldn't much affect results because N is assumed constant here and, moreover, $1/N$ and $1/(N-1)$ do not much differ when N is large.

For the Market town data, with $N = 45$ and $n = 12$, the summary characteristics are in Table 2.2.

The standard deviations in Table 2.2 are at least as twice as small as the ranges, which is true for all data tables (see Statement 2.2.).

The values of the variance s_v^2 and standard deviation s_v obviously depend on the variable's spread measured by its range. Multiplying the column $v \in V$

by $\alpha > 0$ obviously multiplies its range and standard deviation by α, and the variance by α^2.

The quadratic index of spread, s_v^2, depends not only on the scale but also on the character of the feature's distribution within its range. Can we see how?

Let as consider all quantitative variables defined on N entities $i \in I$ and ranged between 0 and 1 inclusive, and analyze at what distributions the variance attains its maximum and minimum values.

It is not difficult to see that any feature v that minimizes the variance s_v^2 is equal to 0 at one of the entities, 1 at another entity, and $y_{iv} = c_v = 1/2$ at all other entities $i \in I$. The minimum variance s_v^2 is $\frac{1}{2N}$ then.

Among the distributions under consideration, the maximum value of s_v^2 is reached at a feature v which is binary, that is, has only boundary points, 0 and 1, as its values. Indeed, if v has any other value at an entity i, then the variance will only increase if we redefine v in such a way that it becomes 0 or 1 at i depending on whether y_{iv} is smaller or greater than v's average c_v. For a binary v, let us specify proportion p of values y_{iv} at which the feature is larger than its grand mean, $y_{iv} > c_v$. Then, obviously, the average $c_v = 0 * (1-p) + 1 * p = p$ and, thus, $s_v^2 = (0-p)^2 * (1-p) + (1-p)^2 * p = p(1-p)$.

The choice of the left and right bounds of the range, 0 and 1, does have an effect on the values attained by the extremal variable but not on the conclusion of its binariness. That means that the following is proven.

Statement 2.1. *With the range and proportion p of values smaller than the average prespecified, the distribution at which the variance reaches its maximum is the distribution of a binary feature having p values at the left bound and $1-p$ values at the right bound of the range.*

Among the binary features, the maximum variance is reached at $p = 1/2$, the maximum uncertainty. This implies one more property.

Statement 2.2. *For any feature, its standard deviation is at least as twice as small as its range.*

Proof: Indeed, with the range being unity between 0 and 1, the maximum variance is $p(1-p) = 1/4$ at $p = 1/2$ leading to the maximum standard deviation of just half of the unity range, q.e.d.

From the intuitive point of view, the range being the same, the greater the variance the better the variable suits the task of clustering.

2.1.3 Categorical case

Let us first consider binary features and then nominal ones.

To quantitatively recode a binary feature, its Yes category is converted into 1 and No into 0. The grand mean of the obtained zero/one variable will be

p_v, the proportion of entities falling in the category Yes. Its variance will be $s_v^2 = p(1-p)$.

In statistics, two types of probabilistic mechanisms for generating zero/one binary variables are considered, Bernoulli/binomial and Poisson. Each relies on having the proportion of ones, p, fixed. However, the binomial distribution assumes that every single entry has the same probability p of being unity, whereas Poisson distribution does not care about individual entries: just that the proportion p of entries randomly thrown into a column must be unity. This subtle difference makes the variance of the Poisson distribution greater: the variance of the binomial distribution is equal to $s_v^2 = p(1-p)$ and the variance of the Poisson distribution is equal to $\pi_v = p$. Thus, the variance of a one-zero feature considered as a quantitative feature corresponds to the statistical model of binomial distribution.

Turning to the case of nominal variables, let us denote the set of categories of a nominal variable l by V_l. Any category $v \in V_l$ is conventionally characterized by its frequency, the number of entities, N_v, falling in it. The sum of frequencies is equal to the total number of entities in I, $\sum_{v \in V_l} N_v = N$. The *relative frequencies*, $p_v = N_v/N$, sum up to unity. The vector $p = (p_v)$, $v \in V_l$ is referred to as the *distribution* of l (over I). A category with the largest frequency is referred to as the distribution's mode. The dispersion of a nominal variable l is frequently measured by the so-called *Gini coefficient*, or qualitative variance:

$$G = \sum_{v \in V_l} p_v(1-p_v) = 1 - \sum_{v \in V_l} p_v^2 \tag{2.3}$$

This is zero if all entities fall in one of the categories only. G is maximum when the distribution is *uniform*, that is, when all category frequencies are the same, $p_v = 1/|V_l|$ for all $v \in V_l$.

A similar measure referred to as *entropy* and defined as

$$H = -\sum_{v \in V_l} p_v \log p_v \tag{2.4}$$

with the logarithm's base 2 is also quite popular. This measure is related to so-called information theory [12]. Entropy reaches its minimum and maximum values at the same distributions as the Gini coefficient. Moreover, the Gini coefficient can be thought of as a linearized version of entropy since $1 - p_v$ linearly approximates $\log p_v$ at p_v close to 1. In fact, both can be considered averaged information measures, just that one uses $-\log p_v$ and the other $1 - p_v$ to express the information contents.

There exists a general formula to express the diversity of a nominal variable as $S_q = (1 - \sum_{v \in V_l} p_v^q)/(q-1)$, $q > 0$ [132]. The entropy and Gini index are special cases of S_q since $S_2 = G$ and $S_1 = H$ assuming S_1 to be the limit of S_q when q tends to 1.

A nominal variable l can be converted into a quantitative format by assigning a zero/one feature to each of its categories $v \in V_l$ coded by 1 or 0 depending on whether an entity falls into the category or not. These binary features are referred to sometimes as dummy variables.

Unlike a binary feature, a two-category nominal feature such as "Gender" is converted into two columns, each corresponding to one of the categories, "Male" and "Female" of "Gender." This way of quantization is quite convenient within the data recovery approach as will be seen further in section 4.3 and others. However, it is also compatible with the traditional view of quantitative measurement scales as expressed in terms of admissible transformations. Indeed, for a nominal scale x, any one-to-one mapping $y = \phi(x)$ is considered admissible. When there are only two categories, x_1 and x_2, they can be recoded into any y_1 and y_2 with an appropriate rescaling factor a and shift b so that the transformation of x to y can be considered an affine one, $y = ax + b$. It is not difficult to prove that $a = (y_1 - y_2)/(x_1 - x_2)$ and $b = (x_1 y_2 - x_2 y_1)/(x_1 - x_2)$ will do the recoding. In other words, nominal features with two categories can be considered quantitative. The binary features, in this context, are those with category Yes coded by 1 and No by 0 for which transformation $y = ax + b$ is meaningful only when $a > 0$.

The vector of averages of the dummy category features, p_v, $v \in V_l$, is nothing but the distribution of l. Moreover, the Gini coefficient appears to be but the summary Bernoullian variance of the dummy category features, $G = \sum_{v \in V_l} p_v(1 - p_v)$. In the case when l has only two categories, this becomes just the variance of any of them doubled. Thus, the transformation of a nominal variable into a bunch of zero-one dummies conveniently converts it into a quantitative format which is compatible with the traditional treatment of nominal features.

2.2 Bivariate analysis

Statistical science in the pre-computer era developed a number of tools for the analysis of interrelations between variables, which will be useful in the sequel. In the remainder of this section, a review is given of the three cases emerging from the pair-wise considerations, with emphasis on the measurement scales: (a) quantitative-to-quantitative, (b) categorical-to-quantitative, and (c) categorical-to-categorical variables. The discussion of the latter case follows that in [93].

2.2.1 Two quantitative variables

Mutual interrelations between two quantitative features can be caught with a scatter plot such as in Figure 1.10, page 24. Two indices for measuring

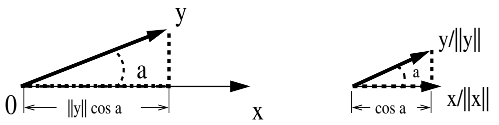


Figure 2.1: Geometrical meaning of the inner product and correlation coefficient.

association between quantitative variables have attracted considerable attention in statistics and data mining: those of covariance and correlation.

The covariance coefficient between the variables x and y considered as columns in a data table, $x = (x_i)$ and $y = (y_i)$, $i \in I$, can be defined as

$$cov(x, y) = (1/N) \sum_{i \in I} (x_i - \bar{x})(y_i - \bar{y}) \tag{2.5}$$

where $\bar{x}$ and $\bar{y}$ are the average values of x and y, respectively.

Obviously, $cov(x, x) = s^2(x)$, the variance of x defined in section 2.1.3.

The covariance coefficient changes proportionally when the variable scales are changed. A scale-invariant version of the coefficient is the correlation coefficient (sometimes referred to as the Pearson product-moment correlation coefficient) which is the covariance coefficient normalized by the standard deviations:

$$r(x, y) = cov(x, y)/(s(x)s(y)) \tag{2.6}$$

A somewhat simpler formula for the correlation coefficient can be obtained if the data are first standardized by subtracting their average and dividing the results by the standard deviation: $r(x, y) = cov(x', y') = (x', y')/N$ where $x'_i = (x_i - \bar{x})/s(x)$, $y'_i = (y_i - \bar{y})/s(y)$, $i \in I$. Thus, the correlation coefficient is but the mean of the component-to-component, that is, inner, product of feature vectors when both of the scales are standardized as above.

The coefficient of correlation can be substantiated in different theoretic frameworks. These require some preliminary knowledge of mathematics and can be omitted at first reading, which is reflected in using a smaller font for explaining them.

1. **Cosine.** A geometric approach, relying on concepts introduced later in section 2.3.2, offers the view that the covariance coefficient as the inner product of feature column-vectors is related to the angle between the vectors so that $(x, y) = ||x||||y|| \cos(x, y)$. This can be illustrated with Fig. 2.1; norms $||x||$ and $||y||$ are Euclidean lengths of intervals from 0 to x and y, respectively. The correlation coefficient is the inner product of the corresponding normalized variables, that is, the cosine of the angle between the vectors.

2. **Linear slope.** The data recovery approach suggests that one of the features is modeled as a linear function of the other, say, y as $ax+b$ where a and b are chosen to minimize the norm of the difference, $||y-ax-b||$. It appears, the optimal slope a is proportional to $r(x,y)$ and, moreover, the square $r(x,y)^2$ expresses that part of the variance of y that is taken into account by $ax+b$ (see details in section 5.1.2).
3. **Parameter in Gaussian distribution.** The correlation coefficient has a very clear meaning in the framework of probabilistic bivariate distributions. Consider, for the sake of simplicity, features x and y normalized so that the variance of each is unity. Denote the matrix formed by the two features by $z=(x,y)$ and assume a unimodal distribution over z, controlled by the so-called Gaussian, or normal, density function (see section 6.1.5) which is proportional to the exponent of $-z^T\Sigma^{-1}z/2$ where Σ is a 2×2 matrix equal to $\Sigma = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$. The parameter r determines the distance between the foci of the ellipse $z^T\Sigma^{-1}z=1$: the greater r the greater the distance. At $r=0$ the distance is zero so that the ellipsis is a circle and at r tending to 1 or -1 the distance tends to the infinity so that the ellipse degenerates into a straight line. It appears $r(x,y)$ is a sample based estimate of this parameter.

These three frameworks capture different pieces of the "elephant." That of cosine is the most universal framework: one may always take that measure to see to what extent two features go in concert, that is, to what extent their highs and lows co-occur. As any cosine, the correlation coefficient is between –1 and 1, the boundary values corresponding to the coincidence of the normalized variables or to a linear relation between the original features. The correlation coefficient being zero corresponds to the right angle between the vectors: the features are not correlated! Does that mean they must be independent in this case? Not necessarily. The linear slope approach allows one to see how this may happen: just the slope of the line best fitting the scatter-plot must be horizontal. According to this approach, the square of the correlation coefficient shows to what extent the relation between variables, as observed, is owed to linearity. The Gaussian distribution view is the most demanding: it requires a properly defined distribution function, a unimodal one, if not normal.

Three examples of scatter-plots on Figure 2.2 illustrate some potential cases of correlation: (a) strong positive, (b) strong negative, and (c) zero correlation. Yet be reminded: in contrast to what is claimed in popular web sites, the the correlation coefficient cannot gather up all the cases in which variables are related; it does capture only those of linear relation and those close enough to that.

2.2.2 Nominal and quantitative variables

How should one measure association between a nominal feature and a quantitative feature? By looking at whether specifying a category can lead to a better

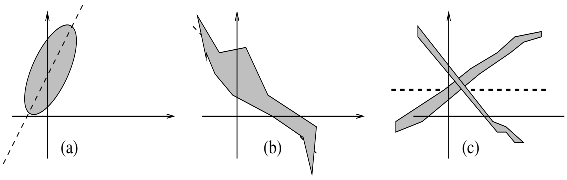


Figure 2.2: Three cases of a scatter-plot: (a) positive correlation, (b) negative correlation, (c) no correlation. The shaded area is supposed to be randomly covered by entity points.

prediction of the quantitative feature or not.

Let us denote the partition of the entity set I corresponding to categories of the nominal variable by $S = \{S_1, ..., S_m\}$; subset S_k consists of N_k entities falling in k-th category of the variable. The quantitative variable will be denoted by y with its values y_i for $i \in I$. The box-plot such as in Figure 1.7 on page 23 is a visual representation of the relationship between the nominal variable represented by the grouping and the quantitative variable represented by the boxes and whiskers.

Let us introduce the framework for prediction of y values. Let the predicted y value for any entity be the grand mean $\bar{y}$ if no other information is supplied, or $\bar{y}_k = \sum_{i \in S_k} y_i/N_k$, the within-class average, if the entity is known to belong to S_k. The average error of these predictions can be estimated in terms of the variances. To do this, one should relate within-class variances of y to its total variance s^2: the greater the change, the lesser the error and the closer the relation between S and y.

An index, referred to as the correlation ratio, measures the proportion of total feature variance that falls within classes. Let us denote the within class variance of variable y by

$$s_k^2 = \sum_{i \in S_k} (y_i - \bar{y}_k)^2/N_k \tag{2.7}$$

where N_k is the number of entities in S_k and $\bar{y}_k$ the feature's within cluster average. Let us denote the proportion of S_k in I by p_k so that $p_k = N_k/N$. Then the average variance within partition $S = \{S_1, ..., S_m\}$ will be $\sum_{k=1}^{K} p_k s_k^2$. This can be proven to never be greater than the total variance s^2.

However, the average within partition variance can be as small as zero – when values y_i all coincide with $\bar{y}_k$ within each category S_k, that is, when y is piece-wise constant across S. In other words, all cluster boxes of a box-plot of y over classes of S degenerate into straight lines in this case. In such a situation partition S is said to perfectly match y. The smaller the difference between

Table 2.3: Cross-classification of 8 masterpieces according to the author and Narrative in the format Count/Proportion.

	Narrative			
Author	Objective	Personal	Direct	Total
Dickens	2/0.250	1/0.125	0/0	3/0.375
Twain	1/0.125	2/0.250	0/0	3/0.375
Tolstoy	0/0	0/0	2/0.250	2/0.250
Total	3/0.375	3/0.375	2/0.375	8/1.000

the average within-class variance and s^2, the worse the match between S and y. The relative value of the difference,

$$\eta^2 = \frac{s^2 - \sum_{k=1}^{K} p_k s_k^2}{s^2} \qquad (2.8)$$

is referred to as the correlation ratio.

The correlation ratio is between 0 and 1, the latter corresponding to the perfect match case. The greater the within-category variances, the smaller the correlation ratio. The minimum value, zero, is reached when all within class variances coincide with the total variance.

2.2.3 Two nominal variables cross-classified

Interrelation between two nominal variables is represented with the so-called contingency table. A contingency, or cross-classification, data table corresponds to two sets of disjoint categories, such as authorship and narrative style in the Masterpieces data, which respectively form rows and columns of Table 2.3.

Entries of the contingency table are co-occurrences of row and column categories, that is, counts of numbers of entities that fall simultaneously in the corresponding row and column categories such as in Table 2.3.

In a general case, with the row categories denoted by $t \in T$ and column categories by $u \in U$, the co-occurrence counts are denoted by N_{tu}. The frequencies of row and column categories usually are called marginals (since they are presented on margins of contingency tables as in Table 2.3) and denoted by N_{t+} and N_{+u} since, when the categories within each of the two sets do not overlap, they are sums of co-occurrence entries, N_{tu}, in rows, t, and columns, u, respectively. The proportions, $p_{tu} = N_{tu}/N$, $p_{t+} = N_{t+}/N$, and $p_{+u} = N_{+u}/N$ are also frequently used as contingency table entries. The general contingency table is presented in Table 2.4.

Contingency tables can be considered for quantitative features too, if they are preliminarily categorized as demonstrated in the following example.

Table 2.4: A contingency table or cross classification of two sets of categories, $t \in T$ and $u \in U$ on the entity set I.

Category	1	2	...	$\|U\|$	Total
1	N_{11}	N_{12}	...	$N_{1\|U\|}$	N_{1+}
2	N_{21}	N_{22}	...	$N_{2\|U\|}$	N_{2+}
...	...	...	....	...	...
$\|T\|$	$N_{\|T\|1}$	$N_{\|T\|2}$	...	$N_{\|T\|\|U\|}$	$N_{\|T\|+}$
Total	N_{+1}	N_{+2}	...	$N_{+\|U\|}$	N

Table 2.5: Cross classification of the Bank related partition with FM feature at Market towns.

	Number of banks				
FMarket	10+	4+	2+	1-	Total
Yes	2	5	1	1	9
No	4	7	13	12	36
Total	6	12	14	13	45

Table 2.6: Frequencies, per cent, in the bivariate distribution of the Bank related partition and FM at Market towns.

	Number of banks				
FMarket	10+	4+	2+	1-	Total
Yes	4.44	11.11	2.22	2.22	20.00
No	8.89	15.56	28.89	26.67	80.00
Total	13.33	26.67	31.11	28.89	100.00

Example 2.1. A cross classification of Market towns

Let us partition the Market town set in four classes according to the number of Banks and Building Societies (feature Ba): class T_1 to include towns with Ba equal to 10 or more; T_2 with Ba equal to 4 or more, but less than 10; T_3 with Ba equal to 2 or 3; and T_4 to consist of towns with one or no bank at all. Let us cross classify partition $T = \{T_v\}$ with feature FM, presence or absence of a Farmers' market in the town. That means that we draw a table whose columns correspond to classes T_v, rows to presence or absence of Farmers' markets, and entries to their overlaps (see Table 2.5).

The matrix of frequencies, or proportions, $p_{tu} = N_{tu}/N$ for Table 2.5 can be found by dividing all its entries by $N = 45$ (see Table 2.6). □

A contingency table gives a picture of interrelation between two categorical features, or partitions corresponding to them, which is not quite clear. Let us make the picture sharper by removing thirteen towns from the sample, those

Table 2.7: Cross classification of the Bank related partition with FM feature at a "cleaned" subsample of Market towns.

	Number of banks				
FMarket	10+	4+	2+	1-	Total
Yes	2	5	0	0	7
No	0	0	13	12	25
Total	2	5	13	12	32

falling in the less populated cells of Table 2.5 (see Table 2.7). Table 2.7 shows a very clear association between two features on the "cleaned" subsample: the Farmers' markets are present only in towns in which the number of banks is 4 or greater. A somewhat subtler relation: the medium numbers of banks are more closly associated with the presence of a Farmers' market than the higher ones.

This clear picture is somewhat blurred in the original sample in Table 2.5 and, moreover, maybe does not hold at all.

Thus, the issue of relating two features to each other can be addressed by looking at mismatches. For instance, Table 2.3 shows that Narrative style is quite close to authorship, though they do not completely match: there are two mismatching entities, one by Dickens and the other by Twain. Similarly, there are 13 mismatches in Table 2.5 removed in Table 2.7. The sheer numbers of mismatching entities measure the differences between category sets rather well when the distribution of entities within each category is rather uniform as it is in Table 2.3. When the proportions of entities in different categories drastically differ, as in Table 2.5, to measure association between category sets more properly, the numbers of mismatching entities should be weighted according to the frequencies of corresponding categories. Can we discover the relation in Table 2.5 without removing entities?

To measure association between categories according to a contingency table, a founding father of the science of statistics, A. Quetelet, proposed utilizing the relative or absolute change of the conditional probability of a category. The conditional probability $p(u/t) = N_{tu}/N_{t+} = p_{tu}/p_{t+}$ measures the proportion of category u in category t. Quetelet coefficients measure the difference between $p(u/t)$ and the average rate p_{+u} of $u \in U$. The Quetelet absolute probability change is defined as

$$g_{tu} = p(u/t) - p_{+u} = (p_{tu} - p_{t+}p_{+u})/p_{t+}, \tag{2.9}$$

and the Quetelet relative change

$$q_{tu} = g_{tu}/p_{+u} = p_{tu}/(p_{t+}p_{+u}) - 1 = (p_{tu} - p_{t+}p_{+u})/(p_{t+}p_{+u}) \tag{2.10}$$

If, for instance, t is an illness risk factor such as “exposure to certain allergens” and u is an allergic reaction such as asthma, and $p_{tu} = 0.001, pt+ = 0.01, p_{+u} = 0.02$, that means that ten per cent of those people who have been exposed to the allergens, $p(u/t) = p_{tu}/p_{t+} = 0.001/0.1 = 0.1$, contract the disease while only two per cent on average have the disease. Thus, the exposure to the allergens multiplies risk of the disease fivefold or increases the probability of contracting it by 400 %. This is exactly the value of $q_{tu} = 0.001/0.0002 - 1 = 4$. The value of g_{tu} expresses the absolute difference between $p(u/t) = 0.1$ and $p_{+u} = 0.02$; it is not that dramatic, just 0.08.

The summary Quetelet coefficients (weighted by the co-occurrence values) can be considered as summary measures of association between two category sets especially when distributions are far from uniform:

$$G^2 = \sum_{t\in T}\sum_{u\in U} p_{tu} g_{tu} = \sum_{t\in T}\sum_{u\in U} \frac{p_{tu}^2}{p_{t+}} - \sum_{u\in U} p_{+w}^2 \tag{2.11}$$

and

$$Q^2 = \sum_{t\in T}\sum_{u\in U} p_{tu} q_{tu} = \sum_{t\in T}\sum_{u\in U} \frac{p_{tu}^2}{p_{t+}p_{+u}} - 1 \tag{2.12}$$

The right-hand 1 in (2.12) comes as $\sum_{t\in T}\sum_{u\in U} p_{tu}$ when the categories t are mutually exclusive and cover the entire set I as well as categories u. In this case Q^2 is equal to the well known Pearson chi-squared coefficient X^2 defined by a different formula:

$$X^2 = \sum_{t\in T}\sum_{u\in U} \frac{(p_{tu} - p_{t+}p_{+u})^2}{p_{t+}p_{+u}} \tag{2.13}$$

The fact that $Q^2 = X^2$ can be proven with simple algebraic manipulations. Indeed, take the numerator in X^2: $(p_{tu} - p_{t+}p_{+u})^2 = p_{tu}^2 - 2p_{tu}p_{t+}p_{+u} + p_{t+}^2 p_{+u}^2$. Divided by the denominator $p_{t+}p_{+u}$, this becomes $p_{tu}^2/p_{t+}p_{+u} - 2p_{tu} + p_{t+}p_{+u}$. Summing up the first item over all u and t leads to $Q^2 + 1$. The second item sums up to -2 and the third item to 1, which proves the statement.

This coefficient is by far the most popular association coefficient. There is a probability-based theory describing what values of NX^2 can be explained by fluctuations of the random sampling from the population.

The difference between equivalent expressions (2.12) and (2.13) for the relative Quetelet coefficient Q^2 reflects deep epistemological differences. In fact, Pearson chi-squared coefficient has been introduced in the format of NX^2 with X^2 in (2.13) to measure the deviation of the bivariate distribution in an observed contingency table from the model of statistical independence. Two partitions (categorical variables) are referred to as statistically independent if any entry in their relative contingency table is equal to the product of corresponding marginal proportions; that is, in our notation,

$$p_{tu} = p_{t+}p_{+u} \tag{2.14}$$

for all $t \in T$ and $u \in U$.

Expression (2.13) for X^2 shows that it is a quadratic measure of deviation of the contingency table entries from the model of statistical independence. This shows that (2.13) is good at testing the hypothesis of statistical independence when I is an independent random sample: the statistical distribution of NX^2 has been proven to converge, when N tends to infinity, to the chi-squared distribution with $(|T| - 1)(|U| - 1)$ degrees of freedom.

Statistics texts and manuals claim that, without relating the observed contingency counts to the model of statistical independence, there is no point in considering X^2. This claim sets a very restrictive condition for using X^2 as an association measure. In particular, it is quite cumbersome to substantiate presence of zero entries (such as in Tables 2.3 and 2.7) in a contingency table under this condition. However, expression (2.12) for Q^2 sets a very different framework that has nothing to do with the statistical independence. In this framework, X^2 is Q^2, the average relative change of the probability of a category u when category t becomes known. There is no restriction on using $X^2 = Q^2$ in this framework.

It is not difficult to prove that the summary coefficient Q^2 reaches its maximum value

$$\max X^2 = \min(|U|, |T|) - 1 \tag{2.15}$$

in tables with the structure of Table 2.7, at which only one element is not zero in every column (or row, if the number of rows is greater than the number of columns)[93]. Such a structure suggests a conceptual relation between categories of the two features, which means that the coefficient is good in measuring association indeed. For instance, according to Table 2.7, Farmers' markets are present if and only if the number of banks or building societies is 4 or greater, and $Q^2 = 1$ in this table.

The minimum value of Q^2 is reached in the case of statistical independence between the features, which obviously follows from the "all squared" form of the coefficient in (2.13).

Formula (2.12) suggests a way for visualization of dependencies in a "blurred" contingency table by putting the constituent items $p_{tu}q_{tu}$ as (t, u) entries of a show-case table. The proportional but greater values $Np_{tu}q_{tu} = N_{tu}q_{tu}$ can be used as well, since they sum up to NX^2 used in the probabilistic framework.

Table 2.8: Relative Quetelet coefficients, per cent, for Table 2.5.

	Number of banks			
FMarket	10+	4+	2+	1-
Yes	**66.67**	**108.33**	-64.29	-61.54
No	-16.67	-27.08	**16.07**	**15.38**

Table 2.9: Items summed up in the chi-square contingency coefficient (times N) in the Quetelet format (2.12) for Table 2.5.

	Number of banks				
FMarket	10+	4+	2+	1-	Total
Yes	**1.33**	**5.41**	-0.64	-0.62	5.48
No	-0.67	-1.90	**2.09**	**1.85**	1.37
Total	0.67	3.51	1.45	1.23	6.86

Example 2.2. Highlighting positive contributions to the total association

The table of the relative Quetelet coefficients q_{tu} for Table 2.5 is presented in Table 2.8 and that of items $N_{tu}q_{tu}$ in Table 2.9.

It is easy to see that the highlighted positive entries in both of the tables express the same pattern as in Table 2.7 but without removing entities from the table.

Table 2.9 demonstrates one more property of the items $p_{tu}q_{tu}$ summed up in the chi-square coefficient: their within-row or within-column sums are always positive. □

Highlighting the positive entries $p_{tu}q_{tu} > 0$ (or $q_{tu} > 0$) can be used for visualization of the pattern of association between any categorical features [94].

A similar to (2.13), though asymmetric, expression can be derived for G^2:

$$G^2 = \sum_{t\in T}\sum_{u\in U}\frac{(p_{tu} - p_{t+}p_{+u})^2}{p_{t+}} \qquad (2.16)$$

Though it also can be considered a measure of deviation of the contingency table from the model of statistical independence, G^2 has been always considered in the literature as a measure of association. A corresponding definition involves the Gini coefficient defined in section 2.1.3, $G(U) = 1 - \sum_{u\in U} p_{+u}^2$. Within a category t, the variation is equal to $G(U/t) = 1 - \sum_{u\in U}(p_{tu}/p_{t+})^2$, which makes, on average, the qualitative variation that cannot be explained by T: $G(U/T) = \sum_{t\in T} p_{t+}G(U/t) = 1 - \sum_{t\in T}\sum_{u\in U} p_{tu}^2/p_{t+}$.

The difference $G(U) - G(U/T)$ represents that part of $G(U)$ that is explained by T, and this is exactly G^2 in (2.11).

2.2.4 Relation between correlation and contingency

Let us elaborate on the interrelation between the correlation and contingency. K. Pearson tackled the issue by proving that, given two quantitative features whose ranges have been divided into a number of equal intervals, under some standard mathematical assumptions, the value of $\sqrt{X^2/(1+X^2)}$ converges to the correlation coefficient when the number of intervals tends to infinity [63].

To define a framework for experimentally exploring the issue in the context of a mixed scale pair of features, let us consider a quantitative feature A and a nominal variable A_t obtained by partitioning the range of A into t qualitative categories, with respect to a pre-specified partition $S = \{S_k\}$. The relation between S and A can be captured by comparing the correlation ratio $\eta^2(S, A)$ with corresponding values of contingency coefficients $G^2(S, A_t)$ and $Q^2(S, A_t)$. The choice of the coefficients is not random. As proven in section 5.2.3, $\eta^2(S, A)$ is equal to the contribution of A and clustering S to the data scatter. In the case of A_t, analogous roles are played by coefficients G^2 and $X^2 = Q^2$.

Relations between η^2, G^2 and X^2 can be quite complex depending on the bivariate distribution of A and S. However, when the distribution is organized in such a way that all the within-class variances of A are smaller than its overall variance, the pattern of association expressed in G^2 and X^2 generally follows that expressed in η^2.

To illustrate this, let us set an experiment according to the data in Table 2.10: within each of four classes, S_1, S_2, S_3, and S_4, a prespecified number of observations is randomly generated with pre-specified mean and variance. The totality of 2300 generated observations constitutes the quantitative feature A for which the correlation ratio $\eta^2(S, A)$ is calculated. Then, the range of A is divided in $t = 5$ equally-spaced intervals (i.e., not necessarily intervals with an equal number of data) constituting categories of the corresponding attribute A_t, which is cross-classified with S to calculate G^2 and X^2. This setting follows that described in [94].

The initial within-class means are not much different with respect to the corresponding variances. Multiplying each of the initial means by the same factor value, $f = 1, 2, ..., 20$, the means are step by step diverged in such a way that the within-class samples become more and more distinguishable from each other, thus increasing the association between S and A. The final means in Table 2.10 correspond to $f = 20$.

This is reflected in Figure 2.3 where the horizontal axis corresponds to the divergence factor, f, and the vertical axis represents values of the three coefficients for the case when the within class distribution of A is uniform (on the left) or Gaussian, or normal (on the right). We can see that the patterns follow each other rather closely in the case of a uniform distribution. There are small diversions from this in the case of a normal distribution. The product-moment correlation between G^2 and X^2 is always about 0.98-0.99 whereas they both

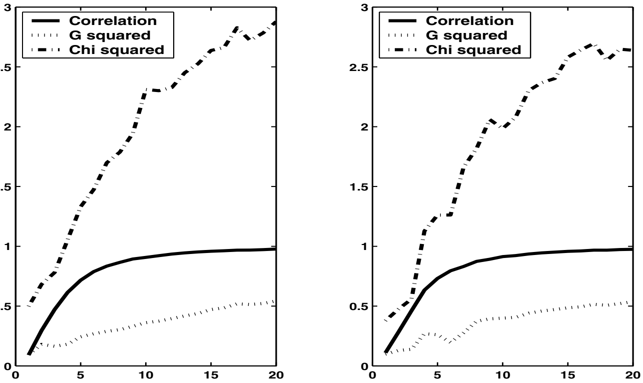


Figure 2.3: Typical change of the correlation ratio (solid line), G squared (dotted line) and chi-square (dashdotted line) with increase of the class divergence factor in the case of uniform (left) and normal (right) within class distribution of the quantitative variable A.

Table 2.10: Setting of the experiment.

Class	S_1	S_2	S_3	S_4
Number of observations	200	100	1000	1000
Variance	1.0	1.0	4.0	0.1
Initial mean	0.5	1.0	1.5	2.0
Final mean	10	20	30	40

correlate with η^2 on the level of 0.90. The difference in values of G^2, X^2 and η^2 is caused by two factors: first, by the coarse qualitative nature of A_t versus the fine-grained quantitative character of A, and, second, by the difference in their contributions to the data scatter. The second factor scales G^2 down and X^2 up, to the maximum value 3 according to (2.15).

2.2.5 Meaning of correlation

Correlation is a phenomenon which may be observed between two features co-occurring in the same observations: the features are co-related in such a way that change in one of them accords with a corresponding change in the other.

These are frequently asked questions: Given a high correlation or association, is there any causal relation behind? Given a low correlation, are the features involved independent? If there is a causal relation, should it translate into a higher correlation?

The answer to each is: no, not necessarily.

To make our point less formal, let us refer to a typical statistics news nugget brought to life by newspapers and BBC Ceefax 25 June 2004: “Children whose

Table 2.11: Association between mother's fish eating (A) and her baby's language skills (B) and health (C).

Feature	B	$\bar{B}$	C	$\bar{C}$	Total
A	520	280	560	240	800
$\bar{A}$	480	720	840	360	1200
Total	1000	1000	1400	600	2000

mothers eat fish regularly during pregnancy develop better language and communication skills. The findings are based on analysis of eating habits of 7400 mothers ... by the University of North Carolina, published in the journal Epidemiology."

At face value, the claim is simple: eat more fish while pregnant and your baby will be better off in the contest of language and communication skills. The real value behind it is a cross classification of a mother's eating habits and her baby's skills over the set of 7400 mother-baby couples at which the cell combining "regular fish eating" with "better language skills" has accounted for a considerably greater number of observations than it would be expected under statistical independence, that is, the corresponding Quetelet coefficient q is positive. So what? Could it be just because of the fish? Very possibly: some say that the phosphorus which is abundant in fish is a building material for the brain. Yet some say that the phosphorus diet has nothing to do with brain development. They think that the correlation is just a manifestation of a deeper relation between family income, not accounted for in the data, and the two features: in richer families it is both the case that mothers eat more expensive food, fish included, and babies have better language skills. The conclusion: more research is needed to see which of these two explanations is correct. And more research may bring further unaccounted for and unforeseen factors and observations.

Example 2.3. False correlation and independence

To illustrate the emergence of "false" correlations and non-correlations, let us dwell on the mother-baby example above involving the following binary features: A – "more fish eating mother," B – "baby's better language skills," and C – "healthier baby." Table 2.11 presents artificial data on two thousand mother-baby couples relating A with B and C.

According to Table 2.11, the baby's language skills (B) are indeed positively related to mother's fish eating (A): 520 observations at cell AB rather than 400 expected if A and B were independent, which is supported by a positive Quetelet coefficient $q(B/A) = 30\%$. In contrast, no relation is observed between fish eating (A) and a baby's health (C): all A/C cross classifying entries on the right of Table 2.11 are proportional to the products of marginal frequencies. For instance, with $p(A) = 0.4$ and $p(C) = 0.7$ their product $p(A)p(C) = 0.28$ accounts for 28% of 2000 observations,

Table 2.12: Association between mother's fish eating (A) and baby's language skills (B) and health (C) with income (D) taken into account.

Feature D	Feature A	B	$\bar{B}$	C	$\bar{C}$	Total
D	A	480	120	520	80	600
	$\bar{A}$	320	80	300	100	400
	Total	800	200	820	180	1000
$\bar{D}$	A	40	160	40	160	200
	$\bar{A}$	160	640	540	260	800
	Total	200	800	580	420	1000
Total		1000	1000	1400	600	2000

that is, 560, which is exactly the entry at cell AC.

However, if we take into account one more binary feature, D, which assigns Yes to better off families, and break down the sample according to D, the data may show a different picture (see Table 2.12). All turned upside down in Table 2.12: what was independent in Table 2.11, A and C, became associated within both D and not-D categories, and what was correlated in Table 2.11, A and B, became independent within both D and not-D categories!

Specifically, with these artificial data, one can see that A accounts for 600 within D category and 200 within not-D category. Similarly, B accounts for 800 within D and only 200 within not-D. Independence between A and B within either strata brings the numbers of AB to 480 in D and only 40 in not-D. This way, the mutually independent A and B within each stratum become correlated in the combined sample, because both A and B are concentrated mostly within D.

Similar though opposite effects are at play with association between A and C: they are negatively related in not-D and positively related in D, so that combining these two strata brings the mutual dependence to zero. □

A high correlation/association is just a pointer to the user, researcher or manager alike, to look at what is behind. The data on their own cannot prove any causal relations, especially when no timing is involved, as is the case in all our exemplary problems. A causal relation can be established only with a mechanism explaining the process in question theoretically, to which the data may or may not add credibility.

2.3 Feature space and data scatter

2.3.1 Data matrix

A quantitative data table is usually referred to as a data matrix. Its rows correspond to entities and columns to variables. Moreover, in most clustering computations, all metadata are left aside so that a feature and entity are represented by the corresponding column and row only, under the assumption that the labels of entities and variables do not change.

A data table with mixed scales such as Table 2.1 will be transformed to a quantitative format. According to rules described in the next section, this is achieved by pre-processing each of the categories into a dummy variable by assigning 1 to an entity that falls in it and 0 otherwise.

Example 2.4. Pre-processing Masterpieces data

Let us convert the Masterpieces data in Table 2.1 to the quantitative format. The binary feature SCon is converted by substituting Yes by 1 and No by zero. A somewhat more complex transformation is performed at the three categories of feature Narrative: each is assigned with a corresponding zero/one vector so that the original column Narrative is converted into three (see Table 2.13). □

Table 2.13: Quantitative representation of the Masterpieces data as an 8 × 7 entity-to-attribute matrix.

Entity	LenSent	LenDial	NumCh	SCon	Objective	Personal	Direct
1	19.0	43.7	2	0	1	0	0
2	29.4	36.0	3	0	1	0	0
3	23.9	38.0	3	0	0	1	0
4	18.4	27.9	2	1	1	0	0
5	25.7	22.3	3	1	0	1	0
6	12.1	16.9	2	1	0	1	0
7	23.9	30.2	4	1	0	0	1
8	27.2	58.0	5	1	0	0	1

A data matrix row corresponding to an entity $i \in I$ constitutes what is called an *M-dimensional point* or *vector* $y_i = (y_{i1}, ..., y_{iM})$ whose components are the row entries. For instance, Masterpieces data in Table 2.13 is a 8 × 7 matrix, and the first row in it constitutes vector $y_1 = (19.0, 43.7, 2, 0, 0, 1, 0)$ each component of which corresponds to a specific feature and, thus, cannot change its position without changing the feature's position in the feature list.

Similarly, a data matrix column corresponds to a feature or category with its elements corresponding to different entities. This is an N-dimensional vector.

Matrix and vector terminology is not just fancy language but part of a well developed mathematical discipline of linear algebra, which is used throughout all data mining disciplines. Some of it will be used in Chapter 5.

2.3.2 Feature space: distance and inner product

Any M-dimensional vector $y = (y_1, ..., y_M)$ pertains to the corresponding combination of feature values. Thus, the set of all M-dimensional vectors y is referred to as the feature space. This space is provided with interrelated distance and similarity measures.

The distance between two M-dimensional vectors, $x = (x_1, x_2, ..., x_M)$ and

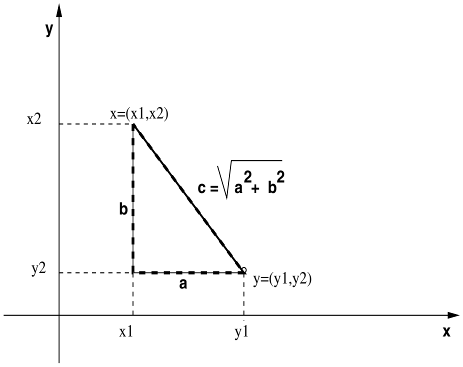


Figure 2.4: Interval between points x and y is the hypotenuse of the highlighted triangle, which explains the distance between x and y.

$y = (y_1, y_2, ..., y_M)$, will be defined as the sum of the component-wise differences squared:

$$d(x,y) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_M - y_M)^2 \tag{2.17}$$

This difference-based quadratic measure is what mathematicians call Euclidean distance squared. It generalizes the basic property of plane geometry, the so-called Pythagoras' theorem as presented in Figure 2.4. Indeed, distance $d(x,y)$ in it is c^2 and c is the Euclidean distance between x and y.

Example 2.5. Distance between entities
Let us consider three novels, 1 and 2 by Dickens, and one, 7 by Tolstoy, as row-points of the matrix in Table 2.13 as presented in the upper half of Table 2.14. The mutual distances between them are calculated in the lower half. The differences in the first two variables, LenS and LenD, predefine the result however different the other features are, because their scales prevail. This way we get a counter-intuitive conclusion that a novel by Dickens is closer to that of Tolstoy than to the other by Dickens because $d(2,7) = 67.89 < d(1,2) = 168.45$. Therefore, feature scales must be rescaled to give greater weights to the other variables.

□

The concept of M-dimensional feature space comprises not only all M-series of reals, $y = (y_1, ..., y_M)$, but also two mathematical operations with them: the component-wise summation defined by the rule $x+y = (x_1+y_1, ..., x_n+y_n)$ and multiplication of a vector by a number defined as $\lambda x = (\lambda x_1, ..., \lambda x_n)$ for any real λ. These operations naturally generalize the operations with real numbers and have similar properties.

With such operations, variance s_v^2 also can be expressed as the distance, between column $v \in V$ and column vector $c_v e$, whose components are all equal to the column's average, c_v, divided by N, $s_v^2 = d(x_v, c_v e)/N$. Vector $c_v e$ is the

Table 2.14: Computation of distances between three masterpieces according to Table 2.13. Squared differences of values in the upper part are in the lower part of the matrix, column-wise; they are summed up in the column "Distance" on the right.

Item	LenS	LenD	NumC	SCon	Obje	Pers	Dire	Distance
1	19.0	43.7	2	0	1	0	0	
2	29.4	36.0	3	0	1	0	0	
7	23.9	30.2	4	1	0	0	1	
Distance								Total
d(1,2)	108.16	59.29	1	0	0	0	0	168.45
d(1,7)	24.01	182.25	4	1	1	0	1	213.26
d(2,7)	30.25	33.64	1	1	1	0	1	67.89

result of multiplication of vector $e = (1, ..., 1)$ whose components are all unities by c_v.

Another important operation is the so-called inner, or scalar, product. For any two M-dimensional vectors x, y their inner product is a number denoted by (x, y) and defined as the sum of component-wise products, $(x, y) = x_1 y_1 + x_2 y_2 + ... + x_M y_M$.

The inner product and distance are closely related. It is not difficult to see, just from the definitions, that for any vectors/points x, y: $d(x, 0) = (x, x) = \sum_{v \in V} x_v^2$ and $d(y, 0) = (y, y)$ and, moreover, $d(x, y) = (x-y, x-y)$. The symbol 0 refers here to a vector with all components equal to zero. The distance $d(y, 0)$ will be referred to as the scatter of y. The square root of the scatter $d(y, 0)$ is referred to as the (Euclidean) norm of y and denoted by $||y|| = \sqrt{(y, y)} = \sqrt{\sum_{i \in I} y_i^2}$. It expresses the length of y.

In general, for any M-dimensional x, y, the following equation holds:

$$d(x, y) = (x - y, x - y) = (x, x) + (y, y) - 2(x, y) = d(0, x) + d(0, y) - 2(x, y). \tag{2.18}$$

This equation becomes especially simple when $(x, y) = 0$. In this case, vectors x, y are referred to as mutually *orthogonal.* When x and y are mutually orthogonal, $d(0, x - y) = d(0, x + y) = d(0, x) + d(0, y)$, that is, the scatters of $x - y$ and $x + y$ are equal to each other and the sum of scatters of x and y. This is a multidimensional analogue to the Pythagoras theorem and the base for decompositions of the data scatter employed in many statistical theories including the theory for clustering presented in Chapter 5.

The inner product of two vectors has a simple geometric interpretation (see Figure 2.1 on page 48), $(x, y) = ||x|| ||y|| \cos \alpha$ where α is the "angle" between x and y (at 0). This conforms to the concept of orthogonality above: vectors are orthogonal when the angle between them is a right angle.

2.3.3 Data scatter

The summary scatter of all row-vectors in data matrix Y is referred to as the data scatter of Y and denoted by

$$T(Y) = \sum_{i \in I} d(0, y_i) = \sum_{i \in I} \sum_{v \in V} y_{iv}^2 \tag{2.19}$$

Equation (2.19) means that $T(Y)$ is the total of all Y entries squared.

An important characteristic of feature $v \in V$ is its *contribution to the data scatter* defined as

$$T_v = \sum_{i \in I} y_{iv}^2, \tag{2.20}$$

the distance of the N-dimensional column from the zero column. Data scatter is obviously the sum of contributions of all variables, $T(Y) = \sum_{v \in V} T_v$.

If feature v is centered, then its contribution to the data scatter is proportional to its variance:

$$T_v = N s_v^2 \tag{2.21}$$

Indeed, $c_v = 0$ since v is centered. Thus, $s_v^2 = \sum_{i \in I} (y_{iv} - 0)^2 / N = T_v / N$.

The relative contribution $T_v / T(Y)$ is a characteristic playing an important role in data standardization issues as explained in the next section.

2.4 Pre-processing and standardizing mixed data

The data pre-processing stage is to transform the raw entity-to-feature table into a quantitative matrix for further analysis. To do this, one needs first to convert all categorical data to a numerical format. We will do this by using a dummy zero-one variable for each category. Then variables are standardized by shifting their origins and rescaling. This operation can be clearly substantiated from a statistics perspective, typically, by assuming that entities have been randomly sampled from an underlying Gaussian distribution. In data mining, substantiation may come from the data geometry. By shifting all the origins to feature means, entities become scattered around the center of gravity so that clusters can be more easily "seen" from that point. With feature rescaling, feature scales become balanced according to the principle of equal importance of each feature brought into the data table.

To implement these general principles, we are going to rely on the following three-stage procedure. The stages are: (1) enveloping qualitative categories, (2) standardization, and (3) rescaling, as follows:

1. **Quantitatively enveloping categories**: This stage is to convert a mixed scale data table into a quantitative matrix by treating every qualitative category as a separate dummy variable coded by 1 or 0 depending on whether an entity falls into the category or not. Binary features are coded similarly except that no additional columns are created. Quantitative features are left as they are. The converted data table will be denoted by $X = (x_{iv})$, $i \in I, v \in V$.

2. **Standardization**: This stage aims at transforming feature-columns of the data matrix to make them comparable by shifting their origins to a_v and rescaling them by b_v, $v \in V$, thus to create standardized matrix $Y = (y_{iv})$:

   $$y_{iv} = \frac{x_{iv} - a_v}{b_v},\ i \in I, v \in V. \tag{2.22}$$

   In this text, the shift coefficient a_v always will be the grand mean. In particular, the dummy variable corresponding to category $v \in V_l$ has its mean $c_v = p_v$, the proportion of entities falling in the category.

   The scale factor b_v can be either the standard deviation or range or other quantity reflecting the variable's spread. In particular, for a category $v \in V_l$, the standard deviation can be either $\sqrt{p_v(1-p_v)}$ (Bernoulli distribution) or $\sqrt{p_v}$ (Poisson distribution), see page 46. The range of a dummy variable is always 1.

   Using the standard deviation is popular in data mining probably because it is used in classical statistics relying on the theory of Gaussian distribution which is characterized by the mean and standard deviation. Thus standardized, contributions of all features to data scatter become equal to each other because of the proportionality of contributions and standard deviations. On first glance this seems an attractive property guaranteeing equal contributions of all features to the results, an opinion to which the current author once also subscribed [90]. However, this is not so. Two different factors contribute to the value of standard deviation: the feature scale and the shape of its distribution. As shown in section 2.1.2, within the same range scale the standard deviation may greatly vary from the minimum, at the peak unimodal distribution, to the maximum, at the peak bimodal distribution. By standardizing with standard deviations, we deliberately bias data in favor of unimodal distributions, although obviously it is the bimodal distribution that should contribute to clustering most. This is why the range, not the standard deviation, is used here as the scaling factor b_v. In the case when there can be outliers in data, which may highly affect the range, another more stable range-like scaling factor can be chosen, such as the difference between percentiles, that does not

Table 2.15: Std standardized Masterpieces matrix; Mean is grand mean, Std the standard deviation and Cntr the relative contribution of a variable.

	LS	LD	SC	NC	Ob	Pe	Di
1	-0.6	0.7	-0.9	-1.2	1.2	-0.7	-0.5
2	1.2	0.1	0.0	-1.2	1.2	-0.7	-0.5
3	0.3	0.3	0.0	-1.2	-0.7	1.2	-0.5
4	-0.7	-0.5	-0.9	0.7	1.2	-0.7	-0.5
5	0.6	-0.9	0.0	0.7	-0.7	1.2	-0.5
6	-1.8	-1.3	-0.9	0.7	-0.7	1.2	-0.5
7	0.3	-0.3	0.9	0.7	-0.7	-0.7	1.6
8	0.8	1.8	1.9	0.7	-0.7	-0.7	1.6
Mean	22.4	34.1	3.0	0.6	0.4	0.4	0.3
Std	5.6	12.9	1.1	0.5	0.5	0.5	0.5
Cntr, %	14.3	14.3	14.3	14.3	14.3	14.3	14.3

much depend on the distribution shape. The range based scaling option has been supported experimentally in [87].

3. **Rescaling**: This stage rescales column-features v, which come from the same categorical variable l, back by further dividing y_{iv} with supplementary rescaling coefficients b'_v to restore the original weighting of raw variables. The major assumption in clustering is that all raw variables are supposed to be of equal weight. Having its categories enveloped, the "weight" of a nominal variable l becomes equal to the summary contribution $T_l = \sum_{v \in V_l} T_v$ to the data scatter where V_l is the set of categories belonging to l. Therefore, to restore the original "equal weighting" of l, the total contribution T_l must be related to $|V_l|$, which is achieved by taking $b'_v = \sqrt{|V_l|}$ for $v \in V_l$.

   For a quantitative $v \in V$, b'_v is, typically, unity.

   Sometimes, there can be available an expert evaluation of the relative weights of the original variables l. If such is the case, rescaling coefficients b'_v should be redefined with the square roots of the expert supplied relative weights. This option may be applied to both quantitative and qualitative features.

Note that two of the three steps above refer to categorical features.

Example 2.6. Effects of different scaling options

Table 2.15 presents the Masterpieces data in Table 2.13 standardized according to the most popular transformation of feature scales, the so-called z-scoring, when the scales are shifted to their mean values and then normalized by the standard deviations. Table 2.16 presents the same data matrix range standardized. All feature contributions are different in this table except for those of NC, Ob and Pe which are the same. Why the same? Because they have the same variance $p(1-p)$ corresponding to p or $1-p$ equal to $3/8$.

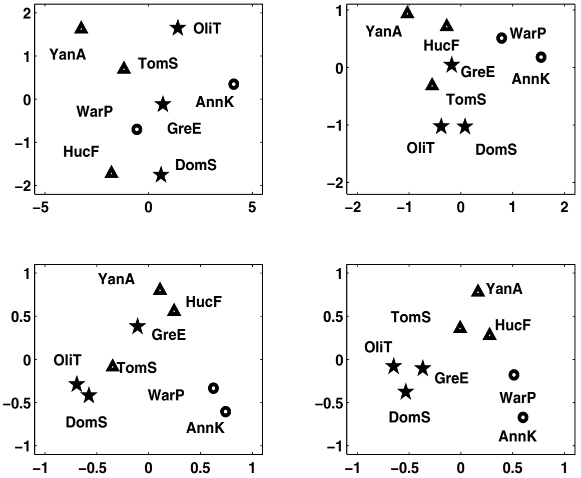


Figure 2.5: Masterpieces on the plane of two first principal components at four different standardizations: no scaling (top left), scaling with standard deviations (top right), range scaling (bottom left), and range scaling with the follow-up rescaling (bottom right).

We can see how overrated the summary contribution of the qualitative variable Narrative becomes: three dummy columns on the right in Table 2.16 take into account 55.7% of the data scatter and thus highly affect any further results. This is why further rescaling of these three variables by the $\sqrt{3}$ is needed to decrease their total contribution 3 times. Table 2.17 presents results of this operation applied to data in Table 2.16.

Note that the total Narrative's contribution per cent has not changed as much as we would expect: about two times, yet not three times!

Figure 2.5 shows how important the scaling can be for clustering results. It displays mutual locations of the eight masterpieces on the plane of the first two principal components of the data in Table 2.13 at different scaling factors: (a) left top: no scaling at all; (b) right top: scaling by the standard deviations, see Table 2.15; (c) left bottom: scaling by ranges; (d) right bottom: scaling by ranges with the follow-up rescaling of the three dummy variables for categories of Narrative by taking into account that they come from the same nominal feature; the scale shifting parameter is always the variable's mean.

The left top scatter-plot displays no relation to the novels' authorship. Probably no clustering algorithm can properly identify the author classes with this standardization of the data (Table 2.15). On the contrary, the authorship pattern is clearly displayed on the bottom right figure, and it is likely that any reasonable clustering algorithm will capture them with this standardization.

We can clearly see on Figure 2.5 that, in spite of the unidimensional nature of transformation (2.22), its combination of shifts and scales can be quite powerful in changing the geometry of the data. □

Table 2.16: Range standardized Masterpieces matrix; Mean is grand mean, Range the range and Cntr the relative contribution of a variable.

	LS	LD	SC	NC	Ob	Pe	Di
1	-0.2	0.2	-0.3	-0.6	0.6	-0.4	-0.3
2	0.4	0.0	0.0	-0.6	0.6	-0.4	-0.3
3	0.1	0.1	0.0	-0.6	-0.4	0.6	-0.3
4	-0.2	-0.2	-0.3	0.4	0.6	-0.4	-0.3
5	0.2	-0.3	0.0	0.4	-0.4	0.6	-0.3
6	-0.6	-0.4	-0.3	0.4	-0.4	0.6	-0.3
7	0.1	-0.1	0.3	0.4	-0.4	-0.4	0.8
8	0.3	0.6	0.7	0.4	-0.4	-0.4	0.8
Mean	22.4	34.1	3.0	0.6	0.4	0.4	0.3
Range	17.3	41.1	3.0	1.0	1.0	1.0	1.0
Cntr, %	7.8	7.3	9.4	19.9	19.9	19.9	15.9

Table 2.17: Range standardized Masterpieces matrix with the additionally rescaled nominal feature attributes; Mean is grand mean, Range the range and Cntr the relative contribution of a variable.

	LS	LD	SC	NC	Ob	Pe	Di
1	-0.20	0.23	-0.33	-0.63	0.36	-0.22	-0.14
2	0.40	0.05	0.00	-0.63	0.36	-0.22	-0.14
3	0.08	0.09	0.00	-0.63	-0.22	0.36	-0.14
4	-0.23	-0.15	-0.33	0.38	0.36	-0.22	-0.14
5	0.19	-0.29	0.00	0.38	-0.22	0.36	-0.14
6	-0.60	-0.42	-0.33	0.38	-0.22	0.36	-0.14
7	0.08	-0.10	0.33	0.38	-0.22	-0.22	0.43
8	0.27	0.58	0.67	0.38	-0.22	-0.22	0.43
Mean	22.45	34.13	3.00	0.63	0.38	0.38	0.25
Range	17.30	41.10	3.00	1.00	1.73	1.73	1.73
Cntr, %	12.42	11.66	14.95	31.54	10.51	10.51	8.41

Example 2.7. Relative feature weighting under standard deviations and ranges may differ

For the Market town data, with $N = 45$ and $n = 12$, the summary feature characteristics are shown in Table 2.18.

As proven above, the standard deviations in Table 2.18 are at least twice as small as the ranges, which is true for all data tables. However, the ratio of the range over the standard deviation may differ for different features reaching as much as 3/0.6=5 for DIY. Therefore, using standard deviations and ranges for scaling in (2.22) may lead to differences in relative scales between the variables and, thus, to different clustering results as well as in Masterpiece data. □

Are there any regularities in the effects of data standardization (and rescaling) on the data scatter and feature contributions to it? Not many. But there are items that should be mentioned:

Table 2.18: Summary characteristics of the Market town data; Mean is grand mean, Std the standard deviation, Range the range and Cntr the relative contribution of a variable.

	P	PS	Do	Ho	Ba	Su	Pe	DIY	SP	PO	CAB	FM
Mean	7351.4	3.0	1.4	0.4	4.3	1.9	2.0	0.2	0.5	2.6	0.6	0.2
Std	6193.2	2.7	1.3	0.6	4.4	1.7	1.6	0.6	0.6	2.1	0.6	0.4
Range	21761.0	12.0	4.0	2.0	19.0	7.0	7.0	3.0	2.0	8.0	2.0	1.0
Cntr, %	8.5	5.4	11.1	8.8	5.6	6.3	5.7	4.2	10.3	7.3	9.7	17.1

Effect of shifting to the average. With shifting to the averages, the feature contributions and variances become proportional to each other, $T_v = Ns_v^2$, for all $v \in V$.

Effects of scaling of categories. Let us take a look at the effect of scaling and rescaling coefficients on categories. The contribution of a binary attribute v, standardized according to (2.22), becomes $T_v = Np_v(1 - p_v)/(b_v)^2$ where p_v is the relative frequency of category v. This can be either $p_v(1-p_v)$ or 1 or $1-p_v$ depending on whether $b_v = 1$ (range) or $b_v = \sqrt{p_v(1-p_v)}$ (Bernoulli standard deviation) or $b_v = \sqrt{p_v}$ (Poisson standard deviation), respectively. These can give some guidance in rescaling of binary categories: the first option should be taken when both zeros and ones are equally important, the second when the distribution does not matter and the third when it is only unities that matter.

Identity of binary and two-category features. An important issue faced by the user is how to treat a categorical feature with two categories such as gender (Male/Female) or voting (Democrat/Republican) or belongingness to a group (Yes/No). The three-step procedure of standardization makes the issue irrelevant: there is no difference between a two-category feature and either of its binary representations.

Indeed, let x be a two-category feature assigning each entity $i \in I$ with a category 'eks1' or 'eks2' whose relative frequencies are p_1 and p_2 such that $p_1 + p_2 = 1$. Denote by $y1$ a binary feature corresponding to category 'eks1' so that $y1_i = 1$ if $x_i =$ eks1 and $y1_i = 0$ if $x_i =$ eks2. Analogously define a binary feature $y2$ corresponding to category 'eks2'. Obviously, $y1$ and $y2$ complement each other so that their sum makes an all unity vector.

In the first stage of the standardization procedure, the user decides whether x is to be converted to a binary feature or left as a categorical one. In the former case, x is converted to a dummy feature, say, column $y1$; and in the latter case, x is converted into a two-column submatrix consisting of columns $y1$ and $y2$. Since the averages of $y1$ and $y2$ are p_1 and p_2, respectively, after shifting column $y1$'s entries become $1-p_1$, for 1, and $-p_1$, for 0. Respective entries of $y2$ are shifted to $-p2$ and $1-p2$, which can be expressed through $p_1 = 1-p_2$ as $p_1 - 1$ and p_1. That means that $y_2 = -y_1$ after the shifting: the two columns become identical, up to the sign. That implies that all their square contribution characteristics become the same, including the total contributions to the data scatter so that the total contribution of the two-column submatrix is twice greater than the contribution of a single column $y1$, whichever scaling option is accepted. However, further rescaling the two-column submatrix by the recommended $\sqrt{2}$ restores the balance of contributions: the two-column submatrix contributes as much as a single column.

Total contributions of categorical features. The total contribution of nominal variable l is

$$T_l = N \sum_{v \in V_l} p_v(1-p_v)/(b_v)^2. \tag{2.23}$$

Depending on the choice of scaling coefficients b_v, this can be

1. $T_l = N(1 - \sum_{v \in V_l} p_v^2)$ if $b_v = 1$ (range normalization);
2. $T_l = N|V_l|$ if $b_v = \sqrt{p_v(1-p_v)}$ (Bernoulli normalization);
3. $T_l = N(|V_l| - 1)$ if $b_v = \sqrt{p_v}$ (Poisson normalization).

where $|V_l|$ is the number of categories of l. The quantity on the top is the Gini coefficient of the distribution of v (2.3).

The square roots of these should be used for further rescaling qualitative categories stemming from the same nominal variable l to adjust their total impact on the data scatter.

2.5 Other table data types

2.5.1 Dissimilarity and similarity data

In many cases the entity-to-entity dissimilarity or similarity data is the preferred format of data as derived from more complex data such as Primates in Table 1.2 or as directly resulting from observations as Confusion data in Table 1.9.

Similarity scoring is especially important for treating the so-called "wide" data tables in which the number of features is much greater than the number of entities. Such is the case of unstructured textual documents for which the presence or absence of a keyword is a feature. The number of meaningful keywords may go into hundreds of thousands even when the entire text collection is in dozens or hundreds. In this case, bringing in a text-to-text similarity index may convert the problem from a virtually untreatable one into a modest size clustering exercise.

An entity-to-entity similarity index may appear as the only data for clustering. For example, similarity scores may come from experiments on subjective judgements such as scoring individual's evaluation of similarity between stimuli or products. More frequently, though, similarities are used when entities to be clustered are too complex to be put in the entity-to-feature table format. When considering two biomolecular amino acid sequences (proteins) a similarity score between them can be based on the probability of transformation of one of them into the other with evolutionary meaningful operations of substitution, deletion and insertion of amino acids [70].

The terminology reflects the differences between the two types of proximity scoring: the smaller the dissimilarity coefficient the closer the entities are, whereas the opposite holds for similarities. Also, the dissimilarity is conventionally considered as a kind of extended distance, thus satisfying some distance properties. In particular, given a matrix $D = (d_{ij})$, $i, j \in I$, where I is the entity set, the entries d_{ij} form a dissimilarity measure between entities $i, j \in I$ if D satisfies the properties:

(a) Symmetry: $d_{ij} = d_{ji}$.

(b) Non-negativity: $d_{ij} \geq 0$.

(c) Semi-definiteness: $d_{ij} = 0$ if entities i and j coincide.

A dissimilarity measure is referred to as a distance if it additionally satisfies:

(d) Definiteness: $d_{ij} = 0$ if and only if entities i and j coincide.

(e) Triangle inequality: $d_{ij} \leq d_{il} + d_{lj}$ for any $i, j, l \in I$.

A distance is referred to as an ultra-metric if it satisfies a stronger triangle inequality:

(f) Ultra-triangle inequality: $d_{ij} \leq \max(d_{il}, d_{lj})$ for any $i, j, l \in I$.

In fact, the ultra-triangle inequality states that among any three distances d_{ij}, d_{il} and d_{jl}, two are equal to each other and the third cannot be greater than that. Ultra-metrics emerge as distances between leaves of trees; in fact, they are equivalent to some tree structures such as the heighted upper cluster hierarchies considered in section 5.3.1.

No such properties are assumed for similarity data except, sometimes, for symmetry (a).

Standardization of similarity data

Given a similarity measure, all entity-to-entity similarities are measured in the same scale so that its change will not change clustering results. This is why there is no need to change the scale of similarity data. As to the shift in the origin of the similarity measure, this can be of advantage by making within- and between-cluster similarities more contrasted. Figure 2.6 demonstrates the effect

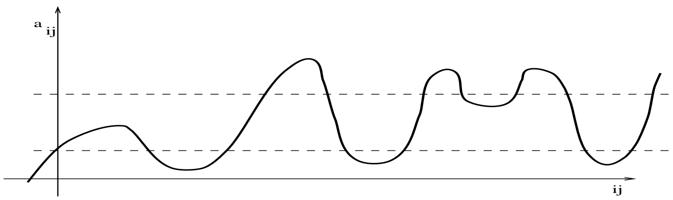


Figure 2.6: A pattern of similarity $a_{ij} = s_{ij} - a$ values depending on a subtracted threshold a.

of changing a positive similarity measure s_{ij} to $a_{ij} = s_{ij} - a$ by subtracting a threshold $a > 0$: small similarities $s_{ij} < a$ can be transformed into negative similarities a_{ij}. This can be irrelevant as for example in such clustering methods as single linkage or similarity-based K-Means. But there are methods such as ADDI-S in section 5.5.5 which can be quite sensitive to the threshold a.

Shift of the origin can be a useful option in standardizing similarity data.

Standardization of dissimilarity data

Given a dissimilarity measure d_{ij}, $i, j \in I$, it is frequently standardized by transforming it into both a row- and column-wise centered similarity measure according to the formula:

$$s_{ij} = -(d_{ij} - d_{i\cdot} - d_{\cdot j} - d_{\cdot\cdot})/2 \tag{2.24}$$

where the dot denotes the operation of averaging so that $d_{i\cdot} = \sum_{j \in I} d_{ij}/N$, $d_{\cdot j} = \sum_{i \in I} d_{ij}/N$, and $d_{\cdot\cdot} = \sum_{i,j \in I} d_{ij}/(N \times N)$.

This formula can be applied to any dissimilarity measure, but it is especially suitable in the situation in which d_{ij} is Euclidean distance squared, that is, when $d_{ij} = (x_i - x_j, x_i - x_j)$ for some multidimensional x_i, x_j, $i, j \in I$. It can be proven then that s_{ij} in (2.24) is the inner product $s_{ij} = (x_i, x_j)$ for all $i, j \in I$ if all x_i are centered.

2.5.2 Contingency and flow data

Table $F = (f_{ij})$, $i \in I$, $j \in J$, is referred to as a flow table if every entry expresses a quantity of the same matter in such a way that all of the entries can be meaningfully summed up to a number expressing the total amount of the matter in the data. Examples of flow data tables are: (a) contingency tables counting numbers of co-occurred instances; (b) mobility tables counting numbers of individual members of a group having changed their categories; (c) trade tables showing the money transferred from i to j during a specified period.

This type of data is of particular interest in processing massive information sources. The data itself can be untreatable within time-memory constraints, but by counting co-occurrences of categories of interest in a sample it can be pre-processed into the flow data format and analyzed as such.

The nature of the data associates weights of row categories $i \in I$, $f_{i+} = \sum_{j \in J} f_{ij}$, and column categories $j \in J$, $f_{+j} = \sum_{i \in I} f_{ij}$, the total flow from i and that to j. The total flow volume is $f_{++} = \sum_{i \in I} \sum_{j \in J} f_{ij}$, which is the summary weight, $f_{++} = \sum_{j \in J} f_{+j} = \sum_{i \in I} f_{i+}$. Extending concepts introduced in section 2.2.3 for contingency tables to the general flow data, we can extend the definition of the relative Quetelet index as

$$q_{ij} = \frac{f_{ij} f_{++}}{f_{i+} f_{+j}} - 1 \tag{2.25}$$

This index, in fact, compares the share of j in i's transaction, $p(j/i) = f_{ij}/f_{i+}$, with the share of j in the overall flow, $p(j) = f_{+j}/f_{++}$.

Indeed, it is easy to see that q_{ij} is the relative difference of the two, $q_{ij} = (p(j/i) - p(j))/p(j)$. Obviously, $q_{ij} = 0$ when there is no difference.

Transformation (2.25) is a standardization of flow data which takes into account the data's nature. Standardization (2.25) is very close to the so-called normalization of Rao and Sabovala widely used in marketing research and, also, the (marginal) cross-product ratio utilized in the analysis of contingency data. Both can be expressed as p_{ij} transformed into $q_{ij} + 1$.

Example 2.8. Quetelet coefficients for Confusion data Coefficients (2.25) are presented in Table 2.19. One can see from the table that the numerals overwhelmingly respond to themselves. However, there are also a few positive entries in the table outside of the diagonal. For instance, 7 is perceived as 1 with the frequency 87.9% greater than the average, and 3 is perceived as 9, and vice versa, with also higher frequencies than the average. □

Table 2.19: Relative Quetelet coefficients for Confusion data, per cent.

Sti-	Response									
mulus	1	2	3	4	5	6	7	8	9	0
1	512.7	-92.6	-92.7	-75.2	-95.8	-83.0	-31.8	-100.0	-95.9	- 96.5
2	-90.2	728.9	-50.8	-95.5	-61.7	-46.7	-84.0	-69.6	-92.9	- 84.1
3	-79.7	-69.3	612.1	-92.1	-80.9	-100.0	-54.5	-69.7	54.6	- 86.8
4	4.1	-76.7	-95.8	725.9	-95.8	-87.6	-65.9	-92.7	-58.3	- 100.0
5	-90.2	-72.5	-55.0	-84.2	610.7	-10.6	-92.0	-92.7	28.3	- 87.6
6	-82.5	-85.2	-92.7	-87.6	2.9	615.8	-95.5	61.9	-88.8	- 62.1
7	87.9	-95.8	-78.0	-76.3	-92.6	-100.0	658.6	-100.0	-95.9	- 93.8
8	-92.3	-70.4	-70.7	-79.7	-80.9	-20.8	-87.5	502.7	-31.9	51.6
9	-82.5	-69.3	16.1	-48.1	-13.0	-87.6	-76.1	-14.3	459.3	- 62.1
0	-87.4	-95.8	-92.7	-87.6	-92.6	-79.6	-71.6	-25.8	-78.6	621.1

Taking into account that the flow data entries can be meaningfully summed up to the total flow volume, the distance between row entities in a contingency table is defined by weighting the columns by their "masses" p_{+l} as follows,

$$\chi(k,k') = \sum_{j\in J} p_{+j}(q_{kj} - q_{k'j})^2. \tag{2.26}$$

This is equal to the so-called chi-square distance defined between row conditional profiles in the Correspondence factor analysis (see section 5.1.4). The formula (2.26) is similar to formula (2.17) for Euclidean distance squared except for the weights. A similar chi-square distance can be defined for columns.

The concept of data scatter for contingency data is also introduced with weighting:

$$X^2(P) = \sum_{I\in I}\sum_{j\in J} p_{i+}p_{+j}q_{ij}^2 \tag{2.27}$$

The notation reflects the fact that this value is closely connected with the Pearson chi-square contingency coefficient, defined in (2.13) above (thus to Q^2 (2.12) as well). Elementary algebraic manipulations show that $X^2(P) = X^2$. Note that in this context the chi-squared coefficient has nothing to do with the statistical independence: it is just a weighted data scatter measure compatible with the specific properties of flow and contingency data. In particular, it is not difficult to prove an analogue to a property of the conventional data scatter: $X^2(P)$ is the sum of chi-square distances $\chi(k,0)$ over all k.

Thus, Pearson's chi-squared coefficient here emerges as the data scatter after the data have been standardized into Quetelet coefficients. For Table 1.9 transformed here into Table 2.19 the coefficient is equal to 4.21 which is 47% of 9, the maximum value of the coefficient for 10 × 10 tables.

Chapter 3

K-Means Clustering

After reading this chapter the reader will know about:

1. Straight and incremental K-Means.
2. The instability of K-Means with regard to initial centroids.
3. An anomalous cluster version of K-Means for incomplete clustering.
4. Three approaches to the initial setting in K-Means: random, maxmin and anomalous pattern.
5. An intelligent version of K-Means mitigating issues of the initial setting and interpretation.
6. Cross-validation of clustering results.
7. Conventional and contribution based interpretation aids for K-Means.

Base words

Anomalous pattern A method for separating a cluster which is most distant from the so-called reference point, which may coincide with the grand mean of the entity set. The method works as K-Means at K=2 except for the location of the reference point which is never changed.

Centroid A multidimensional vector minimizing the summary distance to cluster's elements. If the distance is Euclidean squared, the centroid is equal to the center of gravity of the cluster.

Cluster representative An entity that is considered to represent its cluster well. Conventionally such an entity is drawn as that nearest to the cluster centroid in the Euclidean space. The theory used here suggests that the nearest entity must be drawn according to the inner product rather than distance, which extends the cluster tendencies over the grand mean.

Contributions to the data scatter Additive items representing parts of the data scatter that are explained by certain elements of a cluster structure such as feature-cluster pairs. The greater the contribution, the more important the element. Summary contributions coincide with statistical measures of correlation and association, which is a theoretical support to the recommended data standardization rules.

Cross validation A procedure for testing consistency of a clustering algorithm or its results by the comparison of cluster results found on subsamples formed by a random partitioning of the entity set into a number of groups of equal sizes.

iK-Means An intelligent version of K-Means, in which an initial set of centroids (seeds) is found with an iterated version of the Anomalous pattern algorithm.

Incremental K-Means A version of K-Means in which entities are dealt with one-by-one.

Interpretation aids Computational tools for helping the user to interpret clusters in terms of features, external or used in the process of clustering. Conventional interpretation aids include cluster centroids and bivariate distributions of cluster partitions and features. Contribution based interpretation aids such as ScaD and QScaD tables are derived from the decomposition of the data scatter into parts explained and unexplained by the clustering.

K-Means A major clustering method producing a partition of the entity set into non-overlapping clusters along with within-cluster centroids. It proceeds in iterations consisting of two steps each; one step updates clusters according to the Minimum distance rule, the other step updates centroids as the centers of gravity of clusters. The method implements the so-called alternating minimization algorithm for the square error criterion. To initialize the computations, either a partition or a set of all K tentative centroids must be specified.

Minimum distance rule The rule which assigns each of the entities to its nearest centroid.

Reference point A vector in the variable space serving as the space origin. The Anomalous pattern is sought starting from an entity which is the farthest from the reference point, which thus models the norm from which the Anomalous pattern deviates most.

ScaD and QScaD tables Interpretation aids helping to capture cluster-specific features that are relevant to K-Means clustering results. ScaD is a cluster-to-feature table whose entries are cluster-to-feature contributions to the data scatter. QScaD is a table of the relative Quitelet coefficients of the ScaD entries to express how much they differ from the average.

Square error criterion The sum of summary distances from cluster elements to the cluster centroids, which is minimized by K-Means. The distance used is the Euclidean distance squared, which is compatible with the least-squares data recovery criterion.

3.1 Conventional K-Means

3.1.1 Straight K-Means

K-Means is a major clustering technique that is present, in various forms, in major statistical packages such as SPSS [42] and SAS [17, 119] and data mining packages such as Clementine [14], iDA tool [114] and DBMiner [44].

The algorithm is appealing in many aspects. Conceptually it may be considered a model for the human process of making a typology. Also, it has nice mathematical properties. This method is computationally easy, fast and memory-efficient. However, there are some problems too, especially with respect to the initial setting and stability of results, which will be dealt with in section 3.2.

The cluster structure in K-Means is a partition of the entity set in K non-overlapping clusters represented by lists of entities and within cluster means of the variables. The means are aggregate representations of clusters and as such they are sometimes referred to as standard points or centroids or prototypes. These terms are considered synonymous in the remainder of the text. More formally, the cluster structure is represented by subsets $S_k \subset I$ and M-dimensional centroids $c_k = (c_{kv})$, $k = 1, ..., K$. Subsets S_k form partition $S = \{S_1, ..., S_K\}$ with a set of centroids $c = \{c_1, ..., c_K\}$.

Example 3.9. Centroids of author clusters in Masterpieces data

Let us consider the author-based clusters in the Masterpieces data. The cluster structure is presented in Table 3.1 in such a way that the centroids are calculated twice, once for the raw data in Table 2.13 and the second time, with the standardized data in Table 3.2, which is a copy of Table 2.17 of the previous chapter.

□

Given K M-dimensional vectors c_k as cluster centroids, the algorithm updates cluster lists S_k according to the so-called *Minimum distance rule.*

Minimum distance rule assigns entities to their nearest centroids. Specif-

Table 3.1: Means of the variables in Table 3.2 within K=3 author-based clusters, real (upper row) and standardized (lower row).

		Mean						
Cl.	List	LS (f1)	LD (f2)	NC (f3)	SC (f4)	P (f5)	O (f6)	D (f7)
1	1, 2, 3	24.1	39.2	2.67	0	0.67	0.33	0
		0.095	0.124	-0.111	-0.625	0.168	-0.024	-0.144
2	4, 5, 6	18.7	22.4	2.33	1	0.33	0.67	0
		-0.215	-0.286	-0.222	0.375	-0.024	0.168	-0.144
3	7, 8	25.6	44.1	4.50	1	0.00	0.00	1
		0.179	0.243	0.500	0.375	-0.216	-0.216	0.433

Table 3.2: Range standardized Masterpieces matrix with the additionally rescaled nominal feature attributes copied from Table 2.17.

	LS	LD	NC	SC	Ob	Pe	Di
1	-0.20	0.23	-0.33	-0.63	0.36	-0.22	-0.14
2	0.40	0.05	0.00	-0.63	0.36	-0.22	-0.14
3	0.08	0.09	0.00	-0.63	-0.22	0.36	-0.14
4	-0.23	-0.15	-0.33	0.38	0.36	-0.22	-0.14
5	0.19	-0.29	0.00	0.38	-0.22	0.36	-0.14
6	-0.60	-0.42	-0.33	0.38	-0.22	0.36	-0.14
7	0.08	-0.10	0.33	0.38	-0.22	-0.22	0.43
8	0.27	0.58	0.67	0.38	-0.22	-0.22	0.43

ically, for each entity $i \in I$, its distances to all centroids are calculated, and the entity is assigned to the nearest centroid. When there are several nearest centroids, the assignment is taken among them arbitrarily. In other words, S_k is made of all such $i \in I$ that $d(i, c_k)$ is minimum over all centroids from $c = \{c_1, ..., c_K\}$. The Minimum distance rule is popular in data analysis and can be found under different names such as Voronoi diagrams and vector learning quatization.

In general, some centroids may be assigned no entity at all with this rule.

Having cluster lists updated with the Minimum distance rule, the algorithm updates centroids as gravity centers of the cluster lists S_k; the gravity center coordinates are defined as within cluster averages, that is, updated centroids are defined as $c_k = c(S_k)$, $k = 1, ..., K$, where $c(S)$ is a vector whose components are averages of features over S.

Then the process is reiterated until clusters do not change.

Recall that the distance referred to is Euclidean squared distance defined, for any M-dimensional $x = (x_v)$ and $y = (y_v)$ as $d(x, y) = (x_1 - y_1)^2 + ... + (x_M - y_M)^2$.

Example 3.10. Minimum distance rule at author cluster centroids in Masterpieces data

Let us apply the Minimum distance rule to entities in Table 3.2, given the standardized centroids in Table 3.1. The matrix of distances between the standardized eight row points in Table 3.2 and three centroids in Table 3.1 is in Table 3.3. The table shows that points 1,2,3 are nearest to centroid c_1, 4,5,6 to c_2, and 7, 8 to c_3, which is boldfaced. This means that the rule does not change clusters. These clusters will have the same centroids. Thus, no further calculations can change the clusters: the author-based partition is to be accepted as the result. □

Let us now explicitly formulate the algorithm, which will be referred to as straight K-Means. Sometimes the same procedure is referred to as batch K-Means or parallel K-Means.

Table 3.3: Distances between the eight standardized Masterpiece entities and centroids; within column minima are highlighted.

Centroid	Entity, row point from Table 3.2							
	1	2	3	4	5	6	7	8
	1	2	3	4	5	6	7	8
c_1	**0.22**	**0.19**	**0.31**	1.31	1.49	2.12	1.76	2.36
c_2	1.58	1.84	1.36	**0.33**	**0.29**	**0.25**	0.95	2.30
c_3	2.50	2.01	1.95	1.69	1.20	2.40	**0.15**	**0.15**

Straight K-Means

0. *Data pre-processing.* Transform data into a quantitative matrix Y. This can be done according to the three step procedure described in section 2.4.

1. *Initial setting.* Choose the number of clusters, K, and tentative centroids $c_1, c_2, ..., c_K$, frequently referred to as seeds. Assume initial cluster lists S_k empty.

2. *Clusters update.* Given K centroids, determine clusters S'_k ($k = 1, ..., K$) with the Minimum distance rule.

3. *Stop-condition.* Check whether $S' = S$. If yes, end with clustering $S = S_k$, $c = (c_k)$. Otherwise, change S for S'.

4. *Centroids update.* Given clusters S_k, calculate within cluster means c_k ($k = 1, ..., K$) and go to Step 2.

This algorithm usually converges fast, depending on the initial setting. Location of the initial seeds may affect not only the speed of convergence but, more importantly, the final results as well. Let us give examples of how the initial setting may affect results.

Example 3.11. Successful application of K-Means

Let us apply K-Means to the same Masterpiece data in Table 3.2, this time starting with entities 2, 5 and 7 as tentative centroids (Step 1). To perform Step 2, the matrix of entity-to-centroid distances is computed (see Table 3.4 in which within column minima are boldfaced). The Minimum distance rule produces three cluster lists, $S_1 = \{1,2,3\}, S_2 = \{4,5,6\}$ and $S_3 = \{7,8\}$. These coincide with the author-based clusters and produce within-cluster means (Step 4) already calculated in Table 3.1. Since these differ from the original tentative centroids (entities 2, 5, and 7), the algorithm returns to Step 2 of assigning clusters around the updated centroids. We do not do this here since the operation has been done already with distances in Table 3.3, which produced the same author-based lists according to the Minimum distance rule. The process thus stops. □

Example 3.12. Unsuccessful run of K-Means with different initial seeds

Let us take entities 1, 2 and 3 as the initial centroids (assuming the same data in Table 3.2). The Minimum distance rule, according to entity-to-centroid distances in

Table 3.4: Distances between entities 2, 5, 7 as seeds and the standardized Masterpiece entities.

	Row-point							
Centroid	1	2	3	4	5	6	7	8
2	**0.51**	**0.00**	**0.77**	1.55	1.82	2.99	1.90	2.41
5	2.20	1.82	1.16	**0.97**	**0.00**	**0.75**	0.83	1.87
7	2.30	1.90	1.81	1.22	0.83	1.68	**0.00**	**0.61**

Table 3.5: Distances between the standardized Masterpiece entities and entities 1, 2, 3 as seeds.

	Row-point							
Centroid	1	2	3	4	5	6	7	8
1	**0.00**	0.51	0.88	**1.15**	2.20	2.25	2.30	3.01
2	0.51	**0.00**	0.77	1.55	1.82	2.99	1.90	2.41
3	0.88	0.77	**0.00**	1.94	**1.16**	**1.84**	**1.81**	**2.38**

Table 3.5, leads to cluster lists $S_1 = \{1,4\}, S_2 = \{2\}$ and $S_3 = \{3,5,6,7,8\}$. With the centroids updated at Step 4 as means of these clusters, a new application of Step 3 leads to slightly changed cluster lists $S_1 = \{1,4,6\}, S_2 = \{2\}$ and $S_3 = \{3,5,7,8\}$. Their means calculated, it is not difficult to see that the Minimum distance rule does not change clusters anymore. Thus the lists represent the final outcome, which differs from the author-based solution.

□

The intuitive inappropriateness of the results in this example may be explained by the stupid choice of the initial centroids, all by the same author. However, K-Means can lead to inconvenient results even if the initial setting is selected according to clustering by authors.

Table 3.6: Distances between the standardized Masterpiece entities and entities 1, 4, 7 as tentative centroids.

	Row-point							
Centroid	1	2	3	4	5	6	7	8
1	**0.00**	**0.51**	**0.88**	1.15	2.20	2.25	2.30	3.01
4	1.15	1.55	1.94	**0.00**	0.97	**0.87**	1.22	2.46
7	2.30	1.90	1.81	1.22	**0.83**	1.68	**0.00**	**0.61**

Example 3.13. Unsuccessful K-Means with author-based initial seeds

With the initial centroids at rows 1, 4, and 7, the entity-to-centroid matrix in Table 3.6 leads to cluster lists $S_1 = \{1,2,3\}, S_2 = \{4,6\}$ and $S_3 = \{5,7,8\}$ that do not change in the follow-up operations. These results put a piece by Mark Twain among those by Leo Tolstoy. Not a good outcome. □

3.1.2 Square error criterion

The instability of clustering results with respect to the initial settings leads to a natural question whether there is anything objective in the method at all. Yes, there is.

It appears, there is a scoring function, an index, that is minimized by K-Means. To formulate the function, let us define the within cluster error. For a cluster S_k with centroid $c_k = (c_{kv})$, $v \in V$, its square error is defined as the summary distance from its elements to c_k:

$$W(S_k, c_k) = \sum_{i \in S_k} d(y_i, c_k) = \sum_{i \in S_k} \sum_{v \in V} (y_{iv} - c_{kv})^2. \tag{3.1}$$

The square error criterion is the sum of these values over all clusters:

$$W(S, c) = \sum_{k=1}^{K} W(S_k, c_k) \tag{3.2}$$

Criterion $W(S, c)$ (3.2) depends on two groups of arguments: cluster lists S_k and centroids c_k. Criteria of this type are frequently optimized with the so-called alternating minimization algorithm. This algorithm consists of a series of iterations. At each of the iterations, $W(S, c)$ is, first, minimized over S, given c, and, second, minimized over c, given the resulting S. This way, at each iteration a set c is transformed into a set c'. The calculations stop when c is stabilized, that is, $c' = c$.

Statement 3.3. *Straight K-Means is the alternating minimization algorithm for the summary square-error criterion (3.2) starting from seeds $c = \{c_1, ..., c_K\}$ specified in step 1.*

Proof: Equation

$$W(S, c) = \sum_{k=1}^{K} \sum_{i \in S_k} d(i, c_k),$$

following from (3.1), implies that, given $c = \{c_1, ..., c_K\}$, the Minimum distance rule minimizes $W(S, c)$ over S. Let us now turn to the problem of minimizing $W(S, c)$ over c, given S. It is obvious, that minimizing $W(S, c)$ over c can be done by minimizing $W(S_k, c_k)$ (3.1) over c_k independently for every $k = 1, ..., K$. Criterion $W(S_k, c_k)$ is a quadratic function of c_k and, thus, can be optimized with just first-order optimality conditions that the derivatives of $W(S_k, c_k)$ over c_{kv} must be equal to zero for all $v \in V$. These derivatives are equal to $F(c_{kv}) = -2\sum_{i \in S_k}(y_{iv} - c_{kv})$, $k = 1, ..., K; v \in V$. The condition $F(c_{kv}) = 0$ obviously leads to $c_{kv} = \sum_{i \in S_k} y_{iv}/|S_k|$, which proves that the optimal centroids must be within cluster gravity centers. This proves the statement.

Square-error criterion (3.2) is the sum of distances from entities to their cluster centroids. This can be reformulated as the sum of within cluster variances $\sigma^2_{kv} = \sum_{i \in S_k}(y_{iv} - c_{kv})^2/N_k$ weighted by the cluster cardinalities:

$$W(S,c) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v \in V} (y_{iv} - c_{kv})^2 = \sum_{v \in V} \sum_{k=1}^{K} N_k \sigma^2_{kv} \tag{3.3}$$

Statement 3.3. implies, among other things, that K-Means converges in a finite number of steps because the set of all partitions S over a finite I is finite and $W(S,c)$ is decreased at each change of c or S. Moreover, as experiments show, K-Means typically does not move far away from the initial setting of c. Considered from the perspective of minimization of criterion (3.2), this leads to the conventional strategy of repeatedly applying the algorithm starting from various randomly generated sets of prototypes to reach as deep a minimum of (3.2) as possible. This strategy may fail especially if the feature set is large because in this case random settings cannot cover the space of solutions in a reasonable time.

Yet, there is a different perspective, of typology making, in which the criterion is considered not as something that must be minimized at any cost but rather a beacon for direction. In this perspective, the algorithm is a model for developing a typology represented by the prototypes. The prototypes should come from an external source such as the advice of experts, leaving to data analysis only their adjustment to real data. In such a situation, the property that the final prototypes are not far away from the original ones, is more of an advantage than not. What is important, though, is defining an appropriate, rather than random, initial setting.

The data recovery framework is consistent with this perspective since the model underlying K-Means is based on a somewhat simplistic claim that entities can be represented by their cluster's centroids, up to residuals. This model, according to section 5.2.1, leads to an equation involving K-Means criterion $W(S,c)$ (3.2) and the data scatter $T(Y)$:

$$T(Y) = B(S,c) + W(S,c) \tag{3.4}$$

where

$$B(S,c) = \sum_{k=1}^{K} N_k c^2_{kv} \tag{3.5}$$

In this way, data scatter $T(Y)$ is decomposed into two parts: that one explained by the cluster structure (S,c), that is, $B(S,c)$, and the other unexplained, that is, $W(S,c)$. The larger the explained part the better the match between clustering (S,c) and data.

Criterion $B(S, c)$ measures the part of the data scatter taken into account by the cluster structure.

Example 3.14. Explained part of the data scatter

The explained part of the data scatter, $B(S, c)$, is equal to 43.7% of the data scatter $T(Y)$ for partition $\{\{1, 4, 6\}, \{2\}, \{3, 5, 7, 8\}\}$, found with entities 1,2,3 as initial centroids. The score is 58.9% for partition $\{\{1, 2, 3\}, \{4, 6\}, \{5, 7, 8\}\}$, found with entities 1,4,7 as initial centroids. The score is 64.0% for the author based partition $\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8\}\}$, which is thus superior. □

Advice for selecting the number of clusters and tentative centroids at Step 1 will be given in sections 3.2 and 7.5.

3.1.3 Incremental versions of K-Means

Incremental versions of K-Means are those at which Step 4, with its Minimum distance rule, is executed not for all of the entities but for one of them only. There can be two principal reasons for doing so:

R1 The user is not able to operate with the entire data set and takes entities in one by one, because of either the nature of the data generation process or the largeness of the data set sizes. The former cause is typical when clustering is done in real time as, for instance, in an on-line application. Under traditional assumptions of probabilistic sampling of the entities, convergence of the algorithm was explored in paper [83], from which K-Means became known publicly.

R2 The user operates with the entire data set, but wants to smooth the action of the algorithm so that no drastic changes in the cluster contents may occur. To do this, the user may specify an order of the entities and run entities one-by-one in this order for a number of times. (Each of the runs through the data set is referred to as an "epoch" in the neural network discipline.) The result of this may differ from that of Straight K-Means because of different computations. This computation can be especially effective if the order of entities is not constant but depends on their contributions to the criterion optimized by the algorithm. In particular, each entity $i \in I$ can be assigned value d_i, the minimum of distances from i to centroids c_1,..., c_K, so that i minimizing d_i is considered first.

When an entity y_i joins cluster S_t whose cardinality is N_t, the centroid c_t changes to c'_t to follow the within cluster average values:

$$c'_t = \frac{N_t}{N_t + 1} c_t + \frac{1}{N_t + 1} y_i.$$

When y_i moves out of cluster S_t, the formula remains valid if all pluses are changed for minuses. By introducing the variable z_i which is equal to $+1$ when y_i joins the cluster and -1 when it moves out of it, the formula becomes

$$c'_t = \frac{N_t}{N_t + z_i} c_t + \frac{z_i}{N_t + z_i} y_i \tag{3.6}$$

Accordingly, the distances from other entities change to $d(y_j, c'_t)$.

Because of the incremental setting, the stopping rule of the straight version (reaching a stationary state) may be not necessarily applicable here. In case R1, the natural stopping rule is to end when there are no new entities observed. In case R2, the process of running through the entities one-by-one stops when all entities remain in their clusters. Also, the process may stop when a pre-specified number of runs (epochs) is reached.

This gives rise to the following version of K-Means.

Incremental K-Means: one entity at a time.
1. *Initial setting.* Choose the number of clusters, K, and tentative centroids, $c_1, c_2, ..., c_K$.
2. *Getting an entity.* Observe an entity $i \in I$ coming either randomly (setting R1) or according to a prespecified or dynamically changing order (setting R2).
3. *Cluster update.* Apply Minimum distance rule to determine to what cluster list S_t $(t = 1, ..., K)$ entity i should be assigned.
4. *Centroid update.* Update within cluster centroid c_t with formula (3.6). For the case in which y_i leaves cluster t' (in R2 option), $c_{t'}$ is also updated with (3.6). Nothing is changed if y_i remains in its cluster. Then the stopping condition is checked as described above, and the process moves to observing the next entity (Step 2) or ends (Step 5).
5. *Output.* Output lists S_t and centroids c_t with accompanying interpretation aids (as advised in section 3.4).

Example 3.15. Smoothing action of incremental K-Means

Let us apply version R2 to the Masterpieces data with the entity order dynamically updated and $K = 3$ starting with entities 1, 4 and 7 as centroids. Minimum distances d_i to the centroids for the five remaining entities are presented in the first column of Table 3.7 along with the corresponding centroid (iteration 0). Since $d_2 = 0.51$ is minimum among them, entity 2 is put in cluster I whose centroid is changed accordingly. The next column, iteration 1, presents minimum distances to the updated centroids.

This time the minimum is at $d_8 = 0.61$, so entity 8 is put in its nearest cluster III and its center is recomputed. In iteration 2, the distances are in column 2. Among remaining entities, 3, 5, and 6, the minimum distance is $d_3 = 0.70$, so 3 is added to its closest cluster I. Thus updated the centroid of cluster I leads to the change in minimum distances recorded at iteration 3. This time $d_6 = 0.087$ becomes minimum for the remaining entities 5 and 6 so that 6 joins cluster II and, in the next iteration,

Table 3.7: Minimum distances between standardized Masterpiece entities and dynamically changed centroids I, II and III.

Entity	Iteration					
	0	1	2	3	4	5
2	0.51/I	0.13/I	0.13/I	0.19/I	0.19/I	0.19/I
3	0.87/I	0.70/I	0.70/I	0.31 /I	0.31/I	0.31/I
5	0.83/III	0.83/III	0.97/II	0.97/II	0.97/II	0.28/II
6	0.87/II	0.87/II	0.87/II	0.87/II	0.22/II	0.25/II
8	0.61/III	0.61/III	0.15/III	0.15/III	0.15/III	0.15/III

5 follows it. Then the partition stabilizes: each entity is closer to its cluster centroid than to any other. The final partition of the set of masterpieces is the author based one. We can see that this procedure smoothes the process indeed: starting from the same centroids in Example 3.13, straight K-Means leads to a different and worse partition. □

3.2 Initialization of K-Means

To initialize K-Means, one needs to specify:

(1) the number of clusters, K, and

(2) initial centroids, $c_1, c_2, ..., c_K$.

Each of these can be of issue in practical computations. Both depend on the user's expectations related to the level of resolution and typological attitudes, which remain beyond the scope of the theory of K-Means. This is why some claim these considerations are beyond the clustering discipline. There have been however a number of approaches suggested for specifying the number and location of initial centroids, which will be briefly described in section 7.5.1. Here we present, first, the most popular existing approaches and, second, two approaches based on preliminary analysis of the data set structure.

3.2.1 Traditional approaches to initial setting

Conventionally, either of two extremes is adhered to in initial setting. One view assumes no knowledge of the data and domain and takes initial centroids randomly; the other, on the contrary, relies on the user being an expert and defining initial centroids as hypothetical prototypes.

The first approach randomly selects K of the entities (or generates K n-dimensional points within the feature ranges) as the initial seeds (centroids), and apply K-Means (either straight or incremental). After repeating this a pre-specified number of times (for instance, 100 or 1000), the best solution according to the square-error criterion (3.2) is taken. This approach can be handled by any package containing K-Means. For instance, SPSS allows the taking of the first K entities in a data set as the initial seeds. This can be repeated as many times as needed, each time reformatting the data matrix by putting a random K entity sample as its first K rows.

Selection of K can be done empirically by following this strategy for different values of K, say, in a range from 2 to 15. However, the optimal value of the square-error criterion decreases when K grows and thus cannot be utilized, as is, for the purpose. In the literature, a number of coefficients and tricks have been suggested based on the use of the square error (see later in section 7.5.1). Unfortunately, they all may fail even in the relatively simple situations of controlled computation experiments.

Comparing clusterings found for different K may lead to insights on the cluster structure. In many real world computations, the following phenomenon has been observed by the author and other researchers. When repeatedly proceeding from a larger K to $K-1$, the found $K-1$ clustering, typically, is rather similar to that found by merging some of clusters in the K clustering, in spite of the fact that the K- and $(K-1)$-clusterings are found independently. However, in the process of decreasing K this way, a critical value of K is reached such that $(K-1)$-clustering doesn't resemble K-clustering at all. If this is the case, the value of K can be taken as that corresponding to the cluster structure.

This can be a viable strategy. There are two critical points though.

1. The K-Means algorithm, as is, doesn't seek a global minimum of the square-error criterion and, moreover, the local minima achieved with K-Means are not very deep. Thus, with the number of entities in order of hundreds or thousands and K within a dozen, or more, the number of tries needed to reach a representative set of the initial centroids may become too large and make it a computationally challenging problem.

   To overcome this, some effective computational strategies have been suggested as that of random jumps from a subset of centroids. Such random track changing, typically, produces much deeper minima than the standard K-Means [45].

2. Even if one succeeds in getting a deep or a global minimum of the square-error criterion, it should not be taken for granted that the clusters found reflect the cluster structure. There are some intrinsic flaws in the criterion that would not allow us to accept it as the only means for deciding upon whether the clusters minimizing it are those we are looking for. The square-error criterion needs to be supplemented with other tools for getting better insights into the data structure. Setting of the initial centroids can be utilized as such a tool.

The other approach relies on the opinion of an expert in the subject domain.

Example 3.16. K-Means at Iris data

Table 3.8 presents results of the straight K-Means applied to the Iris data on page 11 with K=3 and specimens numbered 1, 51, and 101 taken as the initial centroids and

Table 3.8: Cross-classification of 150 Iris specimens according to K-Means clustering and the genera; entries show Count/Proportion.

		Iris genus		
Cluster	Setosa	Versicolor	Virginica	Total
S_1	50/0.333	0/0	0/0	50/0.333
S_2	0/0	47/0.313	14/0.093	61/0.407
S_3	0/0	3/0.020	36/0.240	39/0.260
Total	50/0.333	50/0.333	50/0.333	150/1.000

cross-classified with the prior three class partition. The clustering does separate genus *Setosa* but misplaces 14+3=17 specimens between two other genera. This corresponds to the visual pattern in Figure 1.10, page 25. □

Similarly, an expert may propose to distinguish numeral digits by the presence of a closed drawing in them, so that this feature is present in 6 and absent from 1, and suggest these entities as the initial seeds. The expert may even go further and suggest one more feature, presence of a semi-closed drawing instantiated by 3, to be taken into account.

This is a viable approach, too. It allows seeing how the conceptual types relate to the data and to what extent the hypothetical seed combinations match real data.

However, in a common situation in which the user cannot make much sense of his data because they reflect superficial measurable features rather than those of essence, which cannot be measured, the expert vision may fail to suggest a reasonable degree of resolution, and the user should take a more data-driven approach to tackle the problem. Two data-driven approaches are described in the next two sections.

3.2.2 MaxMin for producing deviate centroids

This approach is based on the following intuition. If there are cohesive clusters in the data, then entities within any cluster must be close to each other and rather far away from entities in other clusters. The following method, based on this intuition, has proved to work well in real and simulated experiments.

MaxMin

1. Take entities $y_{i'}$ and $y_{i''}$ maximizing the distance $d(y_i, y_j)$ over all $i, j \in I$ as c_1 and c_2.

2. For each of the entities y_i, that have not been selected to the set c of initial seeds so far, calculate $d_c(y_i)$, the minimum of its distances to $c_t \in c$.

3. Find i^* maximizing $d_c(y_i)$ and check Stop-condition (see below). If it doesn't hold, add y_{i^*} to c and go to Step 2. Otherwise, end and output c as the set of initial seeds.

As the Stop-condition in MaxMin either or all of the following pre-specified constraints can be utilized:

1. The number of seeds has reached a pre-specified threshold.

2. Distance $d_c(y_{i^*})$ is larger than a pre-specified threshold such as $d = d(c_1, c_2)/3$.

3. There is a significant drop, such as 35%, in the value of $d_c(y_{i^*})$ in comparison to that at the previous iteration.

Example 3.17. MaxMin for selecting intial seeds

The table of entity-to-entity distances for Masterpieces is displayed in Table 3.9. The maximum distance here is 3.43, between AK and YA, which makes the two of them initial centroids according to MaxMin. The distances from other entities to these two are in Table 3.10; those minimal at the two are boldfaced. The maximum among them, the next MaxMin distance, is 2.41 between DS and AK. The decrease here is less than 30% suggesting that this can represent a different cluster. Thus, we add DS to the list of candidate centroids and then need to look at distances from other entities to these three (see Table 3.11). This time the MaxMin distance is 0.87 between TS and YA. We might wish to stop the process at this stage since we expect only three meaningful clusters in Masterpieces data and, also, there is a significant drop, 64% of the previous MaxMin distance. It is useful to remember that such a clear-cut situation may not necessarily occur in other examples. The three seeds selected have been shown in previous examples to produce the author based clusters with K-Means. □

Table 3.9: Distances between Masterpieces from Table 3.2.

	OT	DS	GE	TS	HF	YA	WP	AK
OT	0.00	0.51	0.88	1.15	2.20	2.25	2.30	3.01
DS	0.51	0.00	0.77	1.55	1.82	2.99	1.90	2.41
GE	0.88	0.77	0.00	1.94	1.16	1.84	1.81	2.38
TS	1.15	1.55	1.94	0.00	0.97	0.87	1.22	2.46
HF	2.20	1.82	1.16	0.97	0.00	0.75	0.83	1.87
YA	2.25	2.99	1.84	0.87	0.75	0.00	1.68	3.43
WP	2.30	1.90	1.81	1.22	0.83	1.68	0.00	0.61
AK	3.01	2.41	2.38	2.46	1.87	3.43	0.61	0.00

Table 3.10: Distances from Masterpieces entities to YA and AK

	OT	DS	GE	TS	HF	WP
YA	**2.25**	3.00	**1.84**	**0.87**	**0.75**	1.68
AK	3.01	**2.41**	2.38	2.46	1.87	**0.61**

Table 3.11: Distances between DS, YA, and AK and other Masterpiece entities.

	OT	GE	TS	HF	WP
DS	**0.51**	**0.77**	1.55	1.82	1.90
YA	2.25	1.84	**0.87**	**0.75**	1.68
AK	3.01	2.38	2.46	1.87	**0.61**

The issues related to this approach are typical in data mining. First, it involves ad hoc thresholds which are not substantiated in terms of data. Second, it can be computationally intensive when the number of entities N is large since finding the maximum distance at Step 1 involves computation of $O(N^2)$ distances.

3.2.3 Deviate centroids with Anomalous pattern

The method described in this section provides an alternative to MaxMin for the initial setting, which is less intensive computationally and, also, reduces the number of ad hoc parameters.

Reference point based clustering

To avoid the computationally intensive problems of analyzing pair-wise distances, one may employ the concept of a reference point which is chosen to exemplify an average or norm of the features which define the entities. For example, the user might choose, as representing a "normal student," a point which indicates good marks in tests and serious work in projects, and then see what patterns of observed behavior deviate from this. Or, a bank manager may set as his reference point, a customer having specific assets and backgrounds, not necessarily averaged, to see what types of customers deviate from this. In engineering, a moving robotic device should be able to classify the elements of the environment according to the robot's location, with things that are closer having more resolution, and things that are farther having less resolution: the location is the reference point in this case. In many cases the gravity center of the entire entity set can be the reference point of choice.

Availability of a reference point allows the comparison of entities with it, not with each other, which drastically reduces computations. To find a cluster which is most distant from a reference point, a version of K-Means described in [92] can be utilized. According to this procedure, the only ad hoc choice is the cluster's seed. There are two seeds here: the reference point which is unvaried in the process and the cluster's seed, which is taken to be the entity which is farthest from the reference point. Only the anomalous cluster is built here, defined as the set of points that are closer to the cluster seed than to the reference point. Then the cluster seed is substituted by the cluster's gravity center, and the procedure is reiterated until it converges. An exact formulation of this follows.

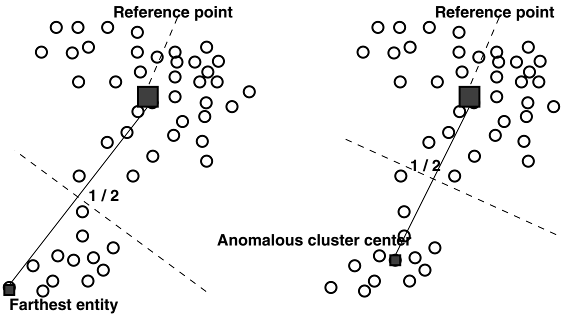


Figure 3.1: Extracting an 'Anomalous pattern' cluster with the reference point in the gravity center: the initial iteration is on the left and the final one on the right.

Anomalous pattern (AP)

1. *Pre-processing.* Specify a reference point $a = (a_1, ..., a_n)$ (this can be the data grand mean) and standardize the original data table with formula (2.22) at which shift parameters a_k are the reference point coordinates. (This way, the space origin is shifted into a.)

2. *Initial setting.* Put a tentative centroid, c, as an entity which is the most distant from the origin, 0.

3. *Cluster update.* Determine cluster list S around c against the only other "centroid" 0 with the Minimum distance rule so that y_i is assigned to S if $d(y_i, c) < d(y_i, 0)$.

4. *Centroid update.* Calculate the within S mean c' and check whether it differs from the previous centroid c. If c' and c do differ, update the centroid by assigning $c \leftarrow c'$ and return to Step 3. Otherwise, go to 5.

5. *Output.* Output list S and centroid c with accompanying interpretation aids (as advised in section 3.4) as the most anomalous pattern.

The process is illustrated in Figure 3.1. Obviously, the Anomalous pattern method is a version of K-Means in which:

(i) the number of clusters K is 2;

(ii) centroid of one of the clusters is 0, which is forcibly kept there through all the iterations;

(iii) the initial centroid of the anomalous cluster is taken as an entity point which is the most distant from 0.

Property (iii) mitigates the issue of determining appropriate initial seeds, which allows using Anomalous pattern algorithm for finding an initial setting for K-Means.

Like K-Means itself, the Anomalous pattern alternately minimizes a criterion,

$$W(S,c) = \sum_{i \in S} d(y_i, c) + \sum_{i \notin S} d(y_i, 0) \tag{3.7}$$

which is a specific version of K-Means general criterion $W(S,c)$ in (3.2): S is a partition in the general criterion and a subset in AP. More technical detail of the method can be found in section 5.5.

Example 3.18. Anomalous pattern in Market towns

The Anomalous pattern method can be applied to the Market towns data in Table 1.1 assuming the grand mean as the reference point and scaling by range. The point farthest from 0, the tentative centroid at step 2, appears to be entity 35 (St Austell) whose distance from zero is 4.33, the maximum. Step 3 adds three more entities, 26, 29 and 44 (Newton Abbot, Penzance and Truro), to the cluster. They are among the largest towns in the data, though there are some large towns like Falmouth that are out of the list, thus being closer to 0 rather than to St Austell in the range standardized feature space. After one more iteration, the anomalous cluster stabilizes.

Table 3.12: Iterations in finding an anomalous pattern in Market towns data.

Iteration	List	#	Distance	Cntr	Cntr, %
1	26, 29, 35, 44	4	2.98	11.92	28.3
2	4, 9, 25, 26, 29, 35, 41, 44	8	1.85	14.77	35.1

The iterations are presented in Table 3.12. It should be noted that the scatter's cluster part (contribution) increases along the iterations as follows from the theory in section 5.5.3: the decrease of the distance between centroid and zero is well compensated by the influx of entities. The final cluster consists of 8 entities and takes into account 35.13 % of the data scatter. Its centroid is displayed in Table 3.13. As frequently happens, the anomalous cluster here consists of better off entities – towns with all the standardized centroid values larger than the grand mean by 30 to 50 per cent of the feature ranges. This probably relates to the fact that they comprise eight out of eleven towns which have a resident population greater than 10,000. The other three largest towns have not made it into the cluster because of their deficiencies in services such as Hospitals and Farmers' markets. The fact that the scale of measurement of population is by far the largest in the original table doesn't much affect the computation here as it runs with the range standardized scales at which the total contribution of this feature is mediocre, about 8.5% only (see Table 2.18). It is rather a concerted action of all features associated with greater population which makes up the cluster. As follows from the last line in Table 3.13, the most important for the cluster separation

Table 3.13: Centroid of the extracted pattern of Market towns.

Centroid	P	PS	Do	Ho	Ba	SM	Pe	DIY	SP	PO	CAB	FM
Real	18484	7.6	3.6	1.1	11.6	4.6	4.1	1.0	1.4	6.4	1.2	.4
Stded	.51	.38	.56	.36	.38	.38	.30	.26	.44	.47	.30	.18
Over GM, %	151	152	163	181	170	139	102	350	181	143	94	88
Related Cntr, %	**167**	147	**154**	81	144	126	84	87	104	**163**	51	10

are the following features: Population, Post offices, and Doctors, highlighted with the boldface. This analysis suggests a simple decision rule separating the cluster entities from the rest with these variables: "P is greater than 10,000 and Do is 3 or greater." □

3.3 Intelligent K-Means

3.3.1 Iterated Anomalous pattern for iK-Means

When clusters in the feature space are well separated from each other or the cluster structure can be thought of as a set of differently contributing clusters, the clusters can be found with iterative application of Anomalous pattern that mitigates the need for pre-setting the number of clusters and their initial centroids. Moreover, this can be used as a procedure to meaningfully determine the number of clusters and initial seeds for K-Means. In this way we come to an algorithm that can be referred to as an intelligent K-Means, because it relieves from the user the task of specifying the initial setting.

Some other potentially useful features of the method relate to its flexibility with regard to dealing with outliers and the "swamp" of inexpressive, ordinary, entities around the grand mean.

iK-Means

0. *Setting.* Put $t = 1$ and I_t the original entity set. Specify a threshold of resolution to discard all AP clusters whose cardinalities are less than the threshold.

1. *Anomalous pattern.* Apply AP to I_t to find S_t and c_t. There can be either option taken: do Step 1 of AP (standardization of the data) at each t or only at $t = 1$. The latter is the recommended option as it is compatible with the theory in section 5.5.

2. *Control.* If Stop-condition (see below) does not hold, put $I_t \leftarrow I_t - S_t$ and $t \leftarrow t + 1$ and go to Step 1.

3. *Removal of small clusters.* Remove all of the found clusters that are smaller than a pre-specified *cluster discarding threshold* for the cluster size. (Entities comprising singleton clusters should be checked for the errors in their data entries.) Denote the number of remaining clusters by K and their centroids by c_1,..., c_K.

4. *K-Means.* Do Straight (or Incremental) K-Means with c_1,..., c_K as initial seeds.

The Stop-condition in this method can be any or all of the following:

1. **All clustered.** $S_t = I_t$ so that there are no unclustered entities left.

2. **Large cumulative contribution.** The total contribution of the first t

clusters to the data scatter has reached a pre-specified threshold such as 50 %.

3. **Small cluster contribution.** Contribution of t-th cluster is too small, say, compared to the order of average contribution of a single entity, $1/N$.

4. **Number of clusters reached.** Number of clusters, t, has reached a pre-specified value K.

The first condition is natural if there are "natural" clusters that indeed differ in their contributions to the data scatter. The second and third conditions can be considered as imposing further degrees of resolution with which the user looks at the data.

At step 4, K-Means can be applied to either the entire dataset or to the set from which the smaller clusters have been removed. This may depend on the situation: in some problems, such as structuring of a set of settlements for better planning or monitoring, no entity should be left out of the consideration, whereas in other problems, such as developing synoptic descriptions for text corpora, some deviant texts should be left out of the coverage.

Example 3.19. Iterated Anomalous patterns in Market towns

Applied to the Market towns data with Stop-condition 1, the iterated AP algorithm has produced 12 clusters of which 5 are singletons. Each of the singletons has a strange pattern of town facilities with no similarity to any other town in the list. For instance, entity 19 (Liskeard, 7044 residents) has an unusually large number of Hospitals (6) and CABs(2), which makes it a singleton cluster.

The seven non-singleton clusters are in Table 3.14, in the order of their extraction in the iterated AP. Centroids of the seven clusters are presented in Table 3.20 in the next section.

Table 3.14: Iterated AP Market towns clusters.

Cluster	Size	Elements	Cntr,%
1	8	4, 9, 25, 26, 29, 35, 41, 44	35.1
3	6	5, 8 , 12, 16, 21, 43	10.0
4	18	2, 6, 7, 10, 13, 14, 17, 22, 23, 24, 27, 30, 31, 33, 34, 37, 38, 40	18.6
5	2	3 , 32	2.4
6	2	1,11	1.6
8	2	39 , 42	1.7
11	2	20 45	1.2

The cluster structure doesn't much change when, according to the iK-Means algorithm, Straight K-Means is applied to the seven centroids (with the five singletons put

Table 3.15: Clusters found by the iterated AP algorithm in Bribery data.

Cluster	Size	Elements	Contribution, %
1	7	5,16,23,27,28,41,42	9.8
2	1	25	2.2
3	2	17,22	3.3
4	1	49	2.2
5	1	2	2.1
6	1	35	2.1
7	13	12,13,20,33,34,38,39,43 45,47,48,50,51	10.7
8	9	4,6,9,10,21,26,30,31,40	10.2
9	5	1,3,15,29,32	6.3
10	2	7,52	3.3
11	3	8,14,36	3.4
12	8	11,24,37,44,46,53,54,55	7.8
13	2	18,19	2.6

back into the data). Moreover, similar results have been observed with clustering of the original list of about thirteen hundred Market towns described by an expanded list of eighteen characteristics of their development: the number of non-singleton clusters was the same, with their descriptions (see page 101) very similar.

□

Example 3.20. Intelligent K-Means on Bribery data

Let us apply iK-Means to the Bribery data in Table 1.12 on page 20. According to the prescriptions above, the data processing includes the following steps:

1. Data standardization. This is done by subtracting the feature averages (grand means) from all entries and then dividing them by the feature ranges. For a binary feature corresponding to a qualitative category, this reduces to subtraction of the category proportion, p, from all the entries which in this way become either $1-p$, for "yes," and $-p$, for "no."

2. Repeatedly performing AP clustering. Applying AP to the pre-processed data matrix with the reference point taken as the space origin 0 and never altered, 13 clusters have been produced as shown in Table 3.15. They explain 64% of the data variance.

3. Initial setting for K-Means. There are only 5 clusters that have more than three elements according to Table 3.15. This defines the number of clusters as well as the initial setting: the first elements of the five larger clusters, indexed as 5, 12, 4, 1, and 11, are taken as the initial centroids.

4. Performing K-Means. K-Means, with the five centroids from the previous step, produces five clusters presented in Table 3.16. They explain 45% of the data scatter. The reduction of the proportion of the explained data scatter is obviously caused by the reduced number of clusters.

Conceptual description of the clusters is left to the next section (see page 106) which is devoted to interpretation aids. □

Table 3.16: Clusters found by K-Means in the entire Bribery data set from the largest clusters in Table 3.15.

Cluster	#	Elements	Contribution, %
1	8	5,16,23,25,27,28,41,42	10.0
2	19	7,8,12,13,14,20,33,34,35,3638,39,43,45,47,48,50,51,52	9.8
3	10	4,6,9,10,21,22,26,30,31,40	10.0
4	7	1,3,15,17,29,32,49	7.0
5	11	2,11,18,19,24,37,44,46,53,54,55	8.1

3.3.2 Cross validation of iK-Means results

As described in section 1.2.6, the issue of validation of clusters may be subject to different perspectives. According to the classification paradigm, validation of clusters is provided by their interpretation, that is, by the convenience of the clusters and their fitting into and enhancing the existing knowledge. In the statistics paradigm, a cluster structure is validated by its correspondence to the underlying model. In the machine learning perspective, it is learning algorithms that are to be validated. In data mining, one validates the cluster structure found. In machine learning and data mining, validation is treated as the testing of how stable the algorithm results are with respect to random changes in the data. We refer the reader to section 7.5 for a general discussion of validation criteria in clustering.

Here we concentrate on the most popular validation method, m-fold cross-validation. According to this method, the entity set is randomly partitioned into m equal parts and m pairs of training and testing sets are formed by taking each one of the m parts as the testing set, with the rest considered the training set.

This scheme is easy to use regarding the problems of learning of decision rules: a decision rule is formed using a training set and then tested on the corresponding testing set. Then testing results are averaged over all m train-test experiments. How can this line of thought be applied to clustering?

In the literature, several methods for extending of the cross-validation techniques to clustering have been described (see references in section 7.5.2). Some of them fall in the machine learning perspective and some in the data mining perspective. The common idea is that the set of m training sets supplied by the cross validation approach constitute a convenient set of random samples from the entity set. In the remainder of this section, we describe somewhat simplified experiments in each of the two frameworks.

In the machine learning framework, one tests the consistency of a clustering algorithm. To do this, results of the algorithm run over each of the m training sets are compared. But how can two clusterings be compared if they partition different sets? One way to do this is by extending each clustering from the training set to the full entity set by assigning appropriate cluster labels to the test set elements. Another way would be to compare partitions pairwise over the overlap of their training sets. The overlap is not necessarily small. If, for instance, $m = 10$, then each of the training sets covers 90% of entities and the pairwise overlap is 80%.

In data mining, it is the clustering results that are tested. In this framework, the selected clustering method is applied to the entire data set before the set is split into m equal-size parts. Then m training sets are formed as usual, by removing one of the parts and combining the other parts. These training sets are used to verify the clustering results found on the entire data set. To do this, the clustering algorithm is applied to each of the m training sets and the found clustering is compared with that obtained on the entire data set.

Let us consider, with examples, how these strategies can be implemented.

Example 3.21. Cross-validation of iK-Means clusters of the Market towns data

Let us address the issue of consistency of clustering results, a data mining approach. We already have found a set of clusters in the Market towns data, see example 3.19 on page 94. This will be referred to as base clustering. To explore how stable base clusters are, let us do 10-fold cross-validation. First, randomly partition the set of 45 towns in 10 classes of approximately the same size, five classes with four towns and five classes with five towns in each. Taking out each of the classes, we get ten 90% subsamples of the original data as the training sets and run iK-Means on each of them. To see how much these clusterings differ from the base clustering found using the entire set, we use three scoring functions, as follows.

1. **Average distance between centroids** *adc***.** Let c_k $(k = 1, ..., 7)$ be base centroids and c'_l $(l = 1, ..., L)$ centroids of the clustering found on a 90% sample. For each c_k find the nearest c'_l over $l = 1, ..., L$, calculate $d(c_k, c'_l)$ and average the distance over all $k = 1, ..., 7$. (The correspondence between c_k and c'_l can also be established with the so-called best matching techniques [3].) This average distance scores the difference between base clusters and sample clusters. The smaller it is the more consistent is the base clustering.
2. **Relative distance between partitions of samples** M**.** Given a 90% training sample, let us compare two partitions of it: (a) the partition found on it with the clustering algorithm and (b) the base partition constrained to the sample. Cross classifying these two partitions, we get a contingency table $P = (p_{tu})$ of frequencies p_{tu} of sample entities belonging to the t-th class of one partition and the u-th class of the other. The distance, or mismatch coefficient, is

$$M = \sum_t p_{t+}^2 + \sum_u p_{+u}^2 - 2\sum_{t,u} p_{tu}^2$$

   where p_{t+} and p_{+u} are summary frequencies over rows and columns of P, as introduced later in formula (7.12).
3. **Relative chi-square contingency coefficient** T**.** This is computed in the same way as distance M; the only difference is that now chi-squared coefficient (2.12), (2.13)

$$X^2 = \sum_{t,u} p_{tu}^2/(p_{t+}p_{+u}) - 1$$

Table 3.17: Averaged results of fifteen cross-validations of Market towns clusters with real and random data.

Method	Real data	Random data
adc	0.064 (0.038)	0.180 (0.061)
M_s	0.018 (0.018)	0.091 (0.036)
T	0.865 (0.084)	0.658 (0.096)

and its normalized version $T = X^2/\sqrt{(K-1)(L-1)}$, the Tchouprov coefficient, are used. Tchouprov coefficient cannot be greater than 1.

Averaged results of fifteen independent 10-fold cross validation tests are presented in the left column of Table 3.17; the standard deviations of the values are in parentheses.

We can see that distances *adc* and M_s are low and contingency coefficient T is high. But how low and how high are they? Can any cornerstones or benchmarks be found?

One may wish to compare *adc* with the average distance between uniformly random vectors. This is not difficult, because the average squared difference $(x-y)^2$ between numbers x and y that are uniformly random in a unity interval is 1/6. This implies that the average distance in 12-dimensional space is 2 which is by far greater than the observed 0.064.

This difference however, shouldn't impress anybody, because the distance 2 refers to an unclustered set. Let us generate thus a uniformly random 45×12 data table and simulate the same computations as with the real data. Results of these computations are in the column on the right of Table 3.17. We can see that distances *adc* and M_s over random data are small too; however, they are 3-5 times greater than those on the real data. If one believes that the average distances at random and real data may be considered as sampling averages of normal or chi-square distributions, one may consider a statistical test of difference such as that by Fisher [63, 50] to be appropriate and lead to a statistically sound conclusion that the hypothesis that the clustering of real data differs from that of random data can be accepted with a great confidence level. □

Example 3.22.

Cross-validation of iK-Means algorithm on the Market towns data

In this example, the cross-validation techniques are applied within the machine learning context, that is to say, we are going to address the issue of the consistency of the clustering algorithm rather than its results.

Thus, the partitions found on the training samples will be compared not with the base clustering but with each other. A 10-fold cross-validation is applied here as in the previous example. Ten 90% cross-validation subsamples of the original data are produced and iK-Means is applied to each of them. Two types of comparison between the ten subsample partitions are used, as follows.

1. **Comparing partitions on common parts.** Two 90% training samples' overlap comprises 80% of the original entities, which allows the building of their contingency table over those common entities. Then both the distance M and chi-squared T coefficients can be used.

2. **Comparing partitions by extending them to the entire entity set.** Given a 90% training sample, let us first extend it to the entire entity set. To

Table 3.18: Averaged comparison scores between iK-Means results at 80% real Market towns and random data.

Method	Real data	Random data
M_s	0.027 (0.025)	0.111 (0.052)
T	0.848 (0.098)	0.604 (0.172)

Table 3.19: Averaged comparison scores between iK-Means results extended to all real Market towns and random data.

Method	Real data	Random data
M_s	0.032 (0.028)	0.128 (0.053)
T	0.832 (0.098)	0.544 (0.179)

do so, each entity from the 10% testing set is assigned to the cluster whose centroid is the nearest to the entity. Having all ten 90% partitions extended this way to the entire data set, their pair-wise contingency tables are built and scoring functions, the distance M and chi-squared T coefficients, are calculated.

Tables 3.18 and 3.19 present results of the pair-wise comparison between partitions found by iK-Means applied to the Market towns data in both ways, on 80% overlaps and on the entire data set after extension, averaged over fifteen ten-fold cross-validation experiments. The cluster discarding threshold has been set to 1 as in the previous examples. We can see that these are similar to figures observed in the previous example though the overall consistency of clustering results decreases here, especially when comparisons are conducted over extended partitions.

It should be noted that the issue of consistency of the algorithm is treated somewhat simplistically in this example, with respect to the Market towns data only, not to a pool of data structures. Also, the concept of algorithm's consistency can be defined differently, for instance, with regards to the criterion optimized by the algorithm.

□

Example 3.23. Higher dimensionality effects

It is interesting to mention that applying the same procedure to the original set of 18 features (not presented), the following phenomenon has been observed. When a matrix 45 × 18 is filled in by a set of uniformly random numbers, iK-Means with the cluster discarding threshold 2, produces two clusters only. However, at the 90% training subsamples iK-Means fails most of the times to produce more than one non-trivial cluster. This is an effect of the higher dimensionality of the feature space relative to the number of entities in this example. Random points are situated too far away from each other in this case and can not be conflated by iK-Means into clusters. One may safely claim that iK-Means differs from other clustering algorithms in that respect that, in contrast to the others, it may fail to partition a data set if it is random. This happens not always but only in the cases in which the number of features is comparable to or greater than half of the number of entities. □

3.4 Interpretation aids

As it was already pointed out, interpretation is an important part of clustering, especially from the classification perspective in which it is a validation tool as well. Unfortunately, this subject is generally not treated within the same framework as 'proper' clustering. The data recovery view of clustering allows us to fill in some gaps here as described in this section.

3.4.1 Conventional interpretation aids

Two conventional tools for interpreting K-Means clustering results (S, c) are:

(1) analysis of cluster centroids c_t and

(2) analysis of bivariate distributions between cluster partition $S = \{S_t\}$ and various features.

In fact, under the zero-one coding system for categories, cross-classification frequencies are nothing but cluster centroids, which allows us to safely suggest that analysis of cluster centroids at various feature spaces is the only conventional interpretation aid.

Example 3.24. Conventional interpretation aids applied to Market towns clusters.

Let us consider Table 3.20 displaying centroids of the seven clusters of Market towns data both in real and range standardized scales. These show some tendencies rather clearly. For instance, the first cluster appears to be a set of larger towns that score 30 to 50 % higher than average on almost all 12 features in the feature space. Similarly, cluster 3 obviously relates to smaller than average towns. However, in other cases, it is not always clear what features caused the separation of some clusters. For instance, both clusters 6 and 7 seem too close to the average to have any real differences at all. □

Table 3.20: Patterns of Market towns in the cluster structure found with iK-Means; the first column displays cluster numbering (top) and cardinalities (bottom).

k/#	Centr	P	PS	Do	Ho	Ba	Su	Pe	DIY	SP	PO	CAB	FM
1	Real	18484	7.63	3.63	1.13	11.63	4.63	4.13	1.00	1.38	6.38	1.25	0.38
8	Stand	0.51	0.38	0.56	0.36	0.38	0.38	0.30	0.26	0.44	0.47	0.30	0.17
2	Real	5268	2.17	0.83	0.50	4.67	1.83	1.67	0.00	0.50	1.67	0.67	1.00
6	Stand	-0.10	-0.07	-0.14	0.05	0.02	-0.01	-0.05	-0.07	0.01	-0.12	0.01	0.80
3	Real	2597	1.17	0.50	0.00	1.22	0.61	0.89	0.00	0.06	1.44	0.11	0.00
18	Stand	-0.22	-0.15	-0.22	-0.20	-0.16	-0.19	-0.17	-0.07	-0.22	-0.15	-0.27	-0.20
4	Real	11245	3.67	2.00	1.33	5.33	2.33	3.67	0.67	1.00	2.33	1.33	0.00
3	Stand	0.18	0.05	0.16	0.47	0.05	0.06	0.23	0.15	0.26	-0.04	0.34	-0.20
5	Real	5347	2.50	0.00	1.00	2.00	1.50	2.00	0.00	0.50	1.50	1.00	0.00
2	Stand	-0.09	-0.04	-0.34	0.30	-0.12	-0.06	-0.01	-0.07	0.01	-0.14	0.18	-0.20
6	Real	8675	3.80	2.00	0.00	3.20	2.00	2.40	0.00	0.00	2.80	0.80	0.00
5	Stand	0.06	0.06	0.16	-0.20	-0.06	0.01	0.05	-0.07	-0.24	0.02	0.08	-0.20
7	Real	5593	2.00	1.00	0.00	5.00	2.67	2.00	0.00	1.00	2.33	1.00	0.00
3	Stand	-0.08	-0.09	-0.09	-0.20	0.04	0.10	-0.01	-0.07	0.26	-0.04	0.18	-0.20

3.4.2 Contribution and relative contribution tables

Here two more interpretation aids are proposed:

1. Decomposition of the data scatter over clusters and features (table ScaD);
2. Quetelet coefficients for the decomposition (table QScaD).

According to (3.4) and (3.5), clustering decomposes the data scatter $T(Y)$ in the explained and unexplained parts, $B(S,c)$ and $W(S,c)$, respectively. The explained part can be further presented as the sum of additive items $B_{kv} = N_k c_{kv}^2$, which account for the contribution of every pair S_k $(k = 1, ..., K)$ and $v \in V$, a cluster and a feature. The unexplained part can be further additively decomposed in contributions $W_v = \sum_{k=1}^{K} \sum_{i \in S_k} (y_{iv} - c_{kv})^2$, which can be differently expressed as $W_v = T_v - B_{+v}$ where T_v and B_{+v} are parts of $T(Y)$ and $B(S,c)$ related to feature $v \in V$, $T_v = \sum_{i \in I} y_{iv}^2$ and $B_{+v} = \sum_{k=1}^{K} B_{kv}$.

This can be displayed as a decomposition of $T(Y)$ in a table ScaD whose rows correspond to clusters, columns to variables and entries to the contributions (see Table 3.21).

Table 3.21: ScaD: Decomposition of the data scatter over a K-Means cluster structure.

Feature Cluster	$f1$	$f2$	fM	Total
S_1	B_{11}	B_{12}	B_{1M}	B_{1+}
S_2	B_{21}	B_{22}	B_{2M}	B_{2+}
S_K	B_{K1}	B_{K2}	B_{KM}	B_{K+}
Expl	B_{+1}	B_{+2}	B_{+M}	$B(S,c)$
Unex	W_1	W_2	W_M	$W(S,c)$
Total	T_1	T_2	T_M	$T(Y)$

Summary rows, Expl and Total, and column, Total, are added to the table; they can be expressed as percentages of the data scatter $T(Y)$. The notation follows the notation of flow data. The row Unex accounts for the "unexplained" differences $W_v = T_v - B_{+v}$. The contributions highlight relative roles of features both at individual clusters and in total.

These can be applied within clusters as well (see Table 3.26 further on as an example).

Example 3.25. Contribution table ScaD for Market towns clusters
Table 3.22 presents the Market towns data scatter decomposed, as in Table 3.21, over both clusters and features.

The table shows that, among the variables, the maximum contribution to the data scatter is reached at FM. This can be attributed to the fact that FM is a binary

Table 3.22: Table ScaD at Market towns: Decomposition of the data scatter over clusters and features.

Cl-r	P	PS	Do	Ho	Ba	Su	Pe	DIY	SP	PO	CAB	FM	Total	Tot.,%
1	2.09	1.18	2.53	1.05	1.19	1.18	0.71	0.54	1.57	1.76	0.73	0.24	14.77	35.13
2	0.06	0.03	0.11	0.01	0.00	0.00	0.02	0.03	0.00	0.09	0.00	3.84	4.19	9.97
3	0.86	0.43	0.87	0.72	0.48	0.64	0.49	0.10	0.85	0.39	1.28	0.72	7.82	18.60
4	0.10	0.01	0.07	0.65	0.01	0.01	0.16	0.07	0.20	0.00	0.36	0.12	1.75	4.17
5	0.02	0.00	0.24	0.18	0.03	0.01	0.00	0.01	0.00	0.04	0.06	0.08	0.67	1.59
6	0.02	0.02	0.12	0.20	0.02	0.00	0.01	0.03	0.30	0.00	0.03	0.20	0.95	2.26
7	0.02	0.02	0.03	0.12	0.00	0.03	0.00	0.02	0.20	0.00	0.09	0.12	0.66	1.56
Expl	3.16	1.69	3.96	2.94	1.72	1.88	1.39	0.79	3.11	2.29	2.56	5.33	30.81	73.28
Unex	0.40	0.59	0.70	0.76	0.62	0.79	1.02	0.96	1.20	0.79	1.52	1.88	11.23	26.72
Total	3.56	2.28	4.66	3.70	2.34	2.67	2.41	1.75	4.31	3.07	4.08	7.20	42.04	100.00

variable: as shown in section 2.1.2, contributions of binary variables are maximal when they cover about half of the sample. The least contributing is DIY. The value of the ratio of the explained part of DIY to the total contribution, 0.79/1.75=0.451, amounts to the correlation ratio between the partition and DIY, as explained in sections 3.4.4 and 5.2.3.

The entries in the table actually combine together cardinalities of clusters with squared differences between the grand mean vector and within-cluster centroids. Some show an exceptional value such as contribution 3.84 of FM to cluster 2, which covers more than 50 % of the total contribution of FM and more than 90% of the total contribution of the cluster. Still, overall they do not give much guidance in judging whose variables' contributions are most important in a cluster because of differences between relative contributions of individual rows and columns. □

To measure the relative influence of contributions B_{kv}, let us utilize the property that they sum up to the total data scatter and, thus, can be considered an instance of the flow data. The table of contributions can be analyzed in the same way as a contingency table (see section 2.2.3). Let us define, in particular, the relative contribution of feature v to cluster S_k, $B(k/v) = B_{kv}/T_v$, to show what part of the variable contribution goes to the cluster. The total explained part of T_v, $B_v = B_{+v}/T_v = \sum_{k=1}^{K} B(k/v)$, is equal to the correlation ratio $\eta^2(S,v)$ introduced in section 2.2.3.

More sensitive measures can be introduced to compare the relative contributions $B(k/v)$ with the contribution of cluster S_k, $B_{k+} = \sum_{v \in V} B_{kv} = N_k d(0, c_k)$, related to the total data scatter $T(Y)$. These are similar to Quetelet coefficients introduced for flow data: the difference $g(k/v) = B(k/v) - B_{k+}/T(Y)$ and the relative difference $q(k/v) = g(k/v)/(B_{k+}/T(Y)) = \frac{T(Y)B_{kv}}{T_v B_{k+}} - 1$. The former compares the contribution of v with the average contribution of variables to S_k. The latter relates this to the cluster's contribution. Index $q(k/v)$ can also be expressed as the ratio of the relative contributions of v: within S_k, B_{kv}/B_{k+}, and in the whole data, $T_v/T(Y)$. We refer to $q(k/v)$ as the Relative contribution index, $RCI(k,v)$.

For each cluster k, features v with the largest $RCI(k, v)$ should be presented to the user for interpretation.

Example 3.26. Table QScaD of the relative and Quetelet indexes
All three indexes of association, $B(k/v)$, $g(k/v)$ and $RCI\, q(k/v)$, applied to the Market towns data in Table 3.22 are presented in Table 3.23 below cluster centroids.

Table 3.23: Tendencies of the cluster structure of Market towns. At each cluster, the first and second lines show the cluster's centroid in raw and standardized scales; the other lines display the relative contribution $B(k/v)$ (Rcnt), difference $g(k/v)$ (Dcnt), and RCI $q(k, v)$, respectively, expressed as percentages. The last three lines show these three indexes applied to the explained parts of feature contributions.

k	C-d	P	PS	Do	Ho	Ba	Su	Pe	DIY	SP	PO	CAB	FM
1	Real	18484.00	7.63	3.63	1.13	11.63	4.63	4.13	1.00	1.38	6.38	1.25	0.38
	Stand	0.51	0.38	0.56	0.36	0.38	0.38	0.30	0.26	0.44	0.47	0.30	0.17
	Rcnt	58.75	51.52	54.17	28.41	50.61	44.31	29.37	30.67	36.43	57.31	17.99	3.40
	Dcnt	23.62	16.39	19.04	-6.72	15.48	9.18	-5.76	-4.46	1.30	22.18	-17.14	-31.73
	RCI	67.23	46.65	54.21	-19.12	44.05	26.14	-16.40	-12.69	3.69	63.15	-48.80	-90.31
2	Real	5267.67	2.17	0.83	0.50	4.67	1.83	1.67	0.00	0.50	1.67	0.67	1.00
	Stand	-0.10	-0.07	-0.14	0.05	0.02	-0.01	-0.05	-0.07	0.01	-0.12	0.01	0.80
	Rcnt	1.54	1.33	2.38	0.41	0.09	0.05	0.73	1.88	0.00	2.79	0.02	53.33
	Dcnt	-8.43	-8.64	-7.59	-9.57	-9.88	-9.93	-9.24	-8.09	-9.97	-7.18	-9.95	43.36
	RCI	-84.52	-86.61	-76.08	-95.93	-99.10	-99.54	-92.72	-81.17	-99.96	-72.05	-99.82	434.89
3	Real	2597.28	1.17	0.50	0.00	1.22	0.61	0.89	0.00	0.06	1.44	0.11	0.00
	Stand	-0.22	-0.15	-0.22	-0.20	-0.16	-0.19	-0.17	-0.07	-0.22	-0.15	-0.27	-0.20
	Rcnt	24.11	18.84	18.60	19.46	20.31	24.06	20.38	5.63	19.60	12.70	31.39	10.00
	Dcnt	5.51	0.24	-0.00	0.86	1.71	5.46	1.79	-12.96	1.00	-5.90	12.79	-8.60
	RCI	29.62	1.30	-0.01	4.63	9.20	29.36	9.61	-69.71	5.39	-31.70	68.78	-46.23
4	Real	11245.33	3.67	2.00	1.33	5.33	2.33	3.67	0.67	1.00	2.33	1.33	0.00
	Stand	0.18	0.05	0.16	0.47	0.05	0.06	0.23	0.15	0.26	-0.04	0.34	-0.20
	Rcnt	2.70	0.38	1.56	17.66	0.37	0.37	6.70	3.76	4.54	0.13	8.73	1.67
	Dcnt	-1.47	-3.79	-2.61	13.49	-3.80	-3.80	2.53	-0.41	0.38	-4.04	4.56	-2.50
	RCI	-35.32	-90.91	-62.62	323.75	-91.10	-91.19	60.68	-9.87	9.06	-96.94	109.47	-60.00
5	Real	5347.00	2.50	0.00	1.00	2.00	1.50	2.00	0.00	0.50	1.50	1.00	0.00
	Stand	-0.09	-0.04	-0.34	0.30	-0.12	-0.06	-0.01	-0.07	0.01	-0.14	0.18	-0.20
	Rcnt	0.48	0.17	5.09	4.86	1.26	0.29	0.00	0.63	0.00	1.28	1.55	1.11
	Dcnt	-1.12	-1.43	3.50	3.27	-0.33	-1.30	-1.59	-0.97	-1.59	-0.31	-0.04	-0.48
	RCI	-70.08	-89.58	219.92	205.73	-20.61	-81.96	-99.79	-60.66	-99.91	-19.48	-2.58	-30.17
6	Real	8674.60	3.80	2.00	0.00	3.20	2.00	2.40	0.00	0.00	2.80	0.80	0.00
	Stand	0.06	0.06	0.16	-0.20	-0.06	0.01	0.05	-0.07	-0.24	0.02	0.08	-0.20
	Rcnt	0.52	0.92	2.60	5.41	0.73	0.02	0.54	1.56	6.93	0.08	0.74	2.78
	Dcnt	-1.74	-1.34	0.34	3.15	-1.53	-2.24	-1.72	-0.69	4.67	-2.18	-1.52	0.52
	RCI	-77.04	-59.31	14.89	139.25	-67.69	-99.25	-76.27	-30.73	206.73	-96.44	-67.17	22.95
7	Real	5593.00	2.00	1.00	0.00	5.00	2.67	2.00	0.00	1.00	2.33	1.00	0.00
	Stand	-0.08	-0.09	-0.09	-0.20	0.04	0.10	-0.01	-0.07	0.26	-0.04	0.18	-0.20
	Rcnt	0.55	0.95	0.57	3.24	0.17	1.23	0.01	0.94	4.54	0.13	2.33	1.67
	Dcnt	-1.01	-0.61	-0.99	1.68	-1.39	-0.33	-1.56	-0.62	2.98	-1.43	0.76	0.11
	RCI	-64.79	-38.95	-63.22	107.78	-89.21	-20.98	-99.68	-39.84	191.16	-91.84	48.96	6.78
Ex.	Rcnt	88.64	74.11	84.97	79.45	73.54	70.32	57.72	45.07	72.05	74.42	62.74	73.96
	Dcnt	15.36	0.83	11.69	6.17	0.26	-2.95	-15.56	-28.21	-1.22	1.14	-10.54	0.68
	RCI	20.96	1.14	15.96	8.42	0.35	-4.03	-21.23	-38.49	-1.67	1.56	-14.38	0.93

Now contributions have become visible indeed. One can see, for instance, that variable Do highly contributes to cluster 5: RCI is 219.9. Why? As the upper number in the cell, 0, shows, this is a remarkable case indeed: no Doctor surgeries in the cluster at all.

The difference between clusters 6 and 7, that was virtually impossible to spot with other interpretation aids, now can be explained by the high RCI values of SP, in excess of 100%, reached at these clusters. A closer look at the data shows that there is a swimming pool in each town in cluster 7 and none in cluster 6. If the variable SP is removed then clusters 6 and 7 will not differ anymore and join together.

Overall, the seven nontrivial clusters can be considered as reflecting the following four tiers of the settlement system: largest towns (Cluster 1), small towns (Cluster

Table 3.24: ScaD for Masterpieces data in Table 3.2.

Title	LenS	LenD	NChar	SCon	Pers	Obje	Dire	Total	Total,%
Dickens	0.03	0.05	0.04	1.17	0.00	0.09	0.06	1.43	24.08
Twain	0.14	0.25	0.15	0.42	0.09	0.00	0.06	1.10	18.56
Tolstoy	0.06	0.12	0.50	0.28	0.09	0.09	0.38	1.53	25.66
Expl	0.23	0.41	0.69	1.88	0.18	0.18	0.50	4.06	68.30
Unex	0.51	0.28	0.20	0.00	0.44	0.44	0.00	1.88	31.70
Total	0.74	0.69	0.89	1.88	0.63	0.63	0.50	5.95	100.00

3), large towns (Clusters 4 and 6), and small-to-average towns (Clusters 2,5 and 7). In particular, the largest town Cluster 1 consists of towns whose population is two to three times larger than the average, and they have respectively larger numbers of all facilities, of which even more represented are Post Offices, Doctors, Primary Schools, and Banks. The small town Cluster 3 consists of the smallest towns with 2-3 thousand residents on average. Respectively, the other facilities are also smaller and some are absent altogether (such as DIY shops and Farmers' markets). Two large town clusters, Cluster 4 and Cluster 6, are formed by towns of nine to twelve thousand residents. Although lack of such facilities as Farmers' market is common to them, Cluster 4 is by far the richer, with service facilities that are absent in Cluster 6, which probably is the cause of the separation of the latter within the tier. Three small-to-average town clusters have towns of about 5,000 residents and differ from each other by the presence of a few fancy objects that are absent from the small town cluster, as well as from the other two clusters of this tier. These objects are: a Farmers' market in Cluster 2, a Hospital in Cluster 5, and a Swimming pool in Cluster 7. □

Example 3.27. ScaD and QScaD for Masterpieces

Tables 3.24 and 3.25 present similar decompositions with respect to author-based clustering of the Masterpieces data in Table 2.13 on page 61. This time, only Quetelet indexes of variables, $RCI(k, v)$ are presented (in Table 3.25).

Table 3.25 shows feature SCon as the one most contributing to the Dickens cluster, feature LenD to the Twain cluster, and features NChar and Direct to the Tolstoy cluster. Indeed, these clusters can be distinctively described by the statements “SCon=0,” “LenD $<$ 28,” and “NChar $>$ 3” (or “Narrative is Direct”), respectively. Curiously, the decisive role of LenD for the Twain cluster cannot be recognized from the absolute contributions in Table 3.24: SCon prevails over the Twain cluster in that table. □

Table 3.25: Relative centroids: cluster centroids standardized and Relative contribution indexes of variables, in cluster first and second lines, respectively.

Title	LenS	LenD	NChar	SCon	Pers	Obje	Dire
Dickens	0.10	0.12	-0.11	-0.63	0.17	-0.02	-0.14
	-83.3	-70.5	-81.5	**158.1**	-40.1	-100.0	-50.5
Twain	-0.21	-0.29	-0.22	0.38	-0.02	0.17	-0.14
	1.2	**91.1**	-9.8	20.2	-100.0	-22.3	-35.8
Tolstoy	0.18	0.24	0.50	0.38	-0.22	-0.22	0.43
	-68.3	-33.0	**119.5**	-41.5	-43.3	-43.3	**197.0**

3.4.3 Cluster representatives

The user can be interested in a conceptual description of a cluster, but he also can be interested in looking at the cluster via its representative, a "prototype." This is especially appealing when the representative is a well known object. Such an object can give much better meaning to a cluster than a logical description in situations where entities are complex and the concepts used in description are superficial and do not penetrate deep into the phenomenon. This is the case, for instance, in mineralogy where a class of minerals can be represented by its stratotype, or in literary studies where a general concept can be represented by a literary character.

To specify what entity should be taken as a representative of its cluster, conventionally that entity is selected which is the nearest to its cluster's centroid. This strategy can be referred to as "the nearest in distance." It can be justified in terms of the square error criterion $W(S,c) = \sum_{k=1}^{K} \sum_{h \in S_k} d(y_h, c_k)$ (3.2). Indeed, the entity $h \in S_k$ which is the nearest to c_k contributes the least to $W(S,c)$, that is, to the unexplained part of the data scatter.

The contribution based approach supplements the conventional approach. Decomposition of the data scatter (3.4) suggests a different strategy by relating to the explained rather than unexplained part of the data scatter. This strategy suggests that the cluster's representative must be the entity that maximally contributes to the explained part, $B(S,c) = \sum_{k=1}^{K} \sum_{v} c_{kv}^2 N_k$.

How can one compute the contribution of an entity to that? There seems nothing of entities in $B(S,c)$. To reveal contributions of individual entities, let us recall that $c_{kv} = \sum_{i \in S_k} y_{iv}/N_k$. Let us take c_{kv}^2 in $B(S,c)$ as the product of c_{kv} with itself, and change one of the factors for the definition. This way we obtain equation $c_{kv}^2 N_k = \sum_{i \in S_k} y_{iv} c_{kv}$. This leads to a formula for $B(S,c)$ as the summary inner product:

$$B(S,c) = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v \in V} y_{iv} c_{kv} = \sum_{k=1}^{K} \sum_{i \in S_k} (y_i, c_k), \qquad (3.8)$$

which shows that the contribution of entity $i \in S_k$ is (y_i, c_k).

The most contributing entity is "the nearest in inner product" to the cluster centroid, which may lead sometimes to different choices. Intuitively, the choice according to the inner product follows tendencies represented in c_k towards the whole of the data rather than c_k itself, which is manifested in the choice according to distance.

Example 3.28. Different concepts of cluster representatives

The entity based elements of the data scatter decomposition for the Dickens cluster from Table 3.24 are displayed in Table 3.26. Now some contributions are negative, which shows that a feature at an entity may be at odds with the cluster centroid. According to this table, the maximum contribution to the data scatter, 8.82%, is

Table 3.26: Decomposition of feature contributions to the Dickens cluster in Table 3.24 (in thousandth). The right-hand column shows distances to the cluster's centroid.

Title	LenS	LenD	NChar	SCon	Pers	Obje	Dire	Cntr	Cntr,%	Dist
OTwist	-19	29	37	391	61	5	21	524	8.82	222
DoSon	38	6	0	391	61	5	21	521	8.77	186
GExpect	8	12	0	391	-36	-9	21	386	6.49	310
Dickens	27	46	37	1172	86	2	62	1431	24.08	0

Table 3.27: Two Dickens' masterpieces along with features contributing to their differences.

Item	LenS	LenD	NChar
OTwist	19.0	43.7	2
DoSon	29.4	36.0	3
Cluster mean	24.1	39.2	2.67
Grand mean	22.4	34.1	3.00

delivered by the novel Oliver Twist. Yet the minimum distance to the cluster's centroid is reached at a different novel, Dombey and Son.

To see why this may happen, let us take a closer look at the two novels versus within cluster and grand means (Table 3.27).

Table 3.27 clearly shows that the cluster's centroid is greater than the grand mean on the first two components and smaller on the third one. These tendencies are better expressed in Dombey and Son over the first component and in Oliver Twist over the other two, which accords with the contributions in Table 3.26. Thus, Oliver Twist wins over Dombey and Son as better representing the differences between the cluster centroid and the overall gravity center, expressed in the grand mean. With the distance measure, no overall type tendency can be taken into account. □

Example 3.29. Interpreting Bribery clusters

Let us apply similar considerations to the five clusters of the Bribery data listed in Table 3.16. Since individual cases are not of interest here, no cluster representatives will be considered. However, it is highly advisable to consult the original data and their description on page 19.

In cluster 1, the most contributing features are: Other branch (777%), Change of category (339%), and Level of client (142%). Here and further in this example the values in parentheses are relative contribution indexes RCI. By looking at the cluster's centroid, one can find specifics of these features in the cluster. In particular, all its cases appear to fall in Other branch, comprising such bodies as universities or hospitals. In each of the cases the client's issue was of a personal matter, and most times (six of the eight cases) the service provided was based on re-categorization of the client into a better category. The category Other branch (of feature Branch) appears to be distinctively describing the cluster: the eight cases in this category constitute the cluster.

Cluster 2 consists of nineteen cases. Its most salient features are: Obstruction of justice (367%), Law enforcement (279%), and Occasional event (151%). By looking at the centroid values of these features, one can conclude: (1) all corruption cases in this cluster have occurred in the law enforcement system; (2) they are mostly done via obstruction of justice for occasional events. The fact (1) is not sufficient for distinctively describing the cluster since there are thirty-four cases, not just nineteen, that have occurred in the law enforcement branch. Two more conditions have been found by a cluster description algorithm, APPCOD (see in section 7.1), to be conjunctively added to (1) to make the description distinctive: (3) the cases occurred at office levels higher than Organization, and (4) no cover-up was involved.

Cluster 3 contains ten cases for which the most salient categories are: Extortion in variable III Type of service (374%), Organization (189%), and Government (175%) in X Branch. Nine of the ten cases occurred in the Government branch, overwhelmingly at the level of organization (feature I) and, also overwhelmingly, the office workers extorted money for rendering their supposedly free services (feature III). The client level here is always of an organization, though this feature is not that salient as the other three features.

Cluster 4 contains seven cases, and its salient categories are: Favors in III (813%), Government in X (291%), and Federal level of Office (238%). Indeed, all its cases occurred in the government legislative and executive branches. The service provided was mostly Favors (six of seven cases). Federal level of corrupt office was not frequent, two cases only. Still, this frequency was much higher than the average, for the two cases are just half of the total number, four, of the cases in which Federal level of office was involved.

Cluster 5 contains eleven cases and pertains to two salient features: Cover-up (707%) and Inspection (369%). All of the cases involve Cover-up as the service provided, mostly in inspection and monitoring activities (nine cases of eleven). A distinctive description of this cluster can be defined to conjunct two statements: it is always a cover-up but not at the level of Organization.

Overall, the cluster structure leads to the following overview of the situation. Most important, it is Branch which is the feature defining Russian corruption when looked at through the media glass. Different branches tend to involve different corruption services. The government corruption involves either Extortion for rendering their free services to organizations (Cluster 3) or Favors (Cluster 4). The law enforcement corruption in higher offices is for either Obstruction of justice (Cluster 2) or Cover-up (Cluster 5). Actually, Cover-up does not exclusively belong in the law enforcement branch: it relates to all offices that are to inspect and monitor business activities (Cluster 5). Corruption cases in Other branch involve re-categorization of individual cases into more suitable categories (Cluster 1). □

3.4.4 Measures of association from ScaD tables

Here we are going to see that summary contributions of clustering towards a feature in ScaD tables are compatible with traditional statistical measures of correlation considered in section 2.2.

Quantitative feature case: Correlation ratio

As proven in section 5.2.3, the total contribution $B_{+v} = \sum_k B_{vk}$ of a quantitative feature v to the cluster-explained part of the scatter, presented in the ScaD tables, is proportional to the correlation ratio between v and cluster partition S, introduced in section 2.2.2. In fact, the correlation ratios can be found by relating the row Expl to row Total in the general ScaD table 3.21.

Example 3.30. Correlation ratio from a ScaD table

The correlation ratio of the variable P (Population resident) over the clustering in Table 3.22 can be found by relating the corresponding entries in rows Expl and Total; it is 3.16/3.56=0.89. This relatively high value shows that the clustering closely – though not entirely – follows this variable. In contrast, the clustering has rather little to do with variable DIY, the correlation ratio of which is equal to 0.79/1.75=0.45. □

Categorical feature case: Chi-square and other contingency coefficients

The summary contribution of a nominal feature l having V_l as the set of its categories, to the clustering partition S has something to do with contingency coefficients introduced in section 2.2.3. It is proven in section 5.2.4 to be equal to

$$B(S,l) = \frac{N}{|V_l|} \sum_{k=1}^{K} \sum_{v \in V_l} \frac{(p_{kv} - p_{k+}p_{+v})^2}{p_{k+}b_v^2} \tag{3.9}$$

where b_v stands for the scaling coefficient at the data standardization. Divisor $|V_l|$, the number of categories, comes from the rescaling stage introduced in section 2.4.

The coefficient $B(S,l)$ in (3.9) can be further specified depending on the scaling coefficients b_v. In particular, the items summed up in (3.9) are:

1. $\frac{(p_{kv}-p_k p_v)^2}{p_k}$ if $b_v = 1$, the range;
2. $\frac{(p_{kv}-p_k p_v)^2}{p_k p_v(1-p_v)}$ if $b_v = \sqrt{p_v(1-p_v)}$, the Bernoullian standard deviation;
3. $\frac{(p_{kv}-p_k p_v)^2}{p_k p_v}$ if $b_u = \sqrt{p_u}$, the Poissonian standard deviation.

Items 1 and 3 above lead to $B(S,l)$ being equal to the summary Quetelet coefficients introduced in section 2.2.3. The Quetelet coefficients, thus, appear to be related to the data standardization. Specifically, G^2 corresponds to $b_v = 1$ and $Q^2 = X^2$ to $b_v = \sqrt{p_v}$. Yet item 2, the Bernoullian standardization, leads to an association coefficient which has not been considered in the literature.

Example 3.31. ScaD based association between a feature and clustering

Let us consider the contingency table between the author-based clustering of masterpieces and the only nominal variable in the data, Narrative (Table 3.28). In this example, the dummy variables have been range normalized and then rescaled with $b'_v = \sqrt{3}$, which is consistent with formula (3.9) with $b_v = 1$ and $|V_l| = 3$ for the calculation of the summary contribution $B(S,l)$. Table 3.29 presents the values of $\frac{(p_{kv}-p_k p_v)^2}{3p_k/N}$ in each cell of the cross classification. In fact, these are entries of the full ScaD table in Table 3.24, page 104, related to the categories of Narrative (columns) and the author-based clusters (rows), with row Total corresponding to row Expl in Table 3.24. In particular, the total contribution of the clustering and variable Narrative is equal to 0.18+0.18+0.50=0.86, or about 14.5% of the data scatter. □

Table 3.28: Cross-classification of the author-based partition and Narrative at the eight masterpieces (in thousandth).

Class	Personal	Objective	Direct	Total
Dickens	125	250	0	375
Twain	250	125	0	375
Tolstoy	0	0	250	250
Total	375	375	250	1000

Table 3.29: Elements of calculation $B(S, l)$ according to formula (3.9) (in ten-thousandth).

Class	Personal	Objective	Direct	Total
Dickens	17	851	625	1493
Twain	851	17	625	1493
Tolstoy	938	938	3750	5626
Total	1806	1806	5000	8606

3.5 Overall assessment

K-Means advantages:

1. Models typology building activity.
2. Computationally effective both in memory and time.
3. Can be utilized incrementally, "on-line."
4. Straightforwardly associates feature salience weights with feature scales.
5. Applicable to both quantitative and categorical data and mixed data provided that care has been taken of the relative feature scaling.
6. Provides a number of interpretation aids including cluster prototypes and features and entities most contributing to cluster specificity.

K-Means issues:

1. Simple convex spherical shape of clusters.
2. Choosing the number of clusters and initial seeds.
3. Instability of results with respect to initial seeds.

The issues above are not necessarily shortcomings. To cope with issue 1, the feature set should be chosen carefully according to the goals of the data analysis. To cope with issue 2, the initial seeds should be selected based on conceptual understanding of the substantive domain or preliminary data analysis with the AP clustering approach. There can be some advantages in the issues as well. Issue 3 keeps solutions close to pre-specified centroid settings, which is good when centroids have been conceptually substantiated. Issue 1 of simplicity of cluster shapes provides for a possibility of deriving simple conjunctive descriptions of the clusters, which can be used as supplementary interpretation aids (see section 6.3).

A clustering algorithm should present the user with a comfortable set of options to do clustering. In our view, the intelligent version of K-Means described above and its versions, implementing the possibility of removal of entities that have been found either (1) “deviant” (contents of small Anomalous pattern clusters), or (2) “intermediate” (entities that are far away from their centroids, or have small attraction index values), or (3) “trivial” (entities that are close to the grand mean), give the user an opportunity to select a preferred option without imposing on him technical issues.