RR 86-10

GROUPALS: A Method to Cluster Objects with Mixed Measurement Levels

Stef van Buuren

Department of Data Theory

Leiden University

- ABSTRACT -

This paper proposes a method to cluster objects that are measured on a set of categorical variables with mixed measurement levels. The new feature of the approach is that the scaling of variables, and thereby the construction of distances between objects, and the clustering of objects are performed simultaneously. The method works by minimizing the value of a loss function that finds Principal Components of the input variables and that has been constrainted by a restriction on the objects scores. A computer program, called GROUPALS, using an Alternating Least Squares algorithm, was developed to apply the method to actual data.

- TABLE OF CONTENTS -

- PREFACE -

This report has been written as a thesis for the 'doctorandus degree' in the specialization of Methods & Techniques of Psychology at the Univerity of Leiden.

First of all, I like to thank the members of the Criminological Institute for giving me the opportunity to spend a lot of time working on this paper and for the use of various computing and copying resources.
Second, I thank Prof. P. Rousseeuw for sending me the FORTRAN code of the SILHOUETTES method. This method accounts for a valuable enhancement of the interpretability of the program output.
Third, I thank Eveline Kroezen for her thorough checking on syntax, clarity and structure of the paper.
Last but not least, I thank Willem Heiser of the Department of Datatheory who did not only serve as an excellent supervisor but who also provided the larger part of the mathematical foundation of the method as described in sections 3.2 and 3.3. This report would have been far less sophisticated without his suggestions.

3/6/86

CHAPTER ONE

INTRODUCTION

1.1 Purpose of the study

This section introduces the problem of this study.

In social sciences it is common practice to investigate social phenomena by comparing groups of people.
In some cases the groups of interest can be identified by a few variables, for instance as in experimental research. Frequently however, we want groups to be composed of people that differ on a considerable number of discriminating variables.
This study is concerned with the latter kind of analysis:

> The problem of this study is, given datamatrix H with m categorical variables and n objects, how to find that partition of n objects into k mutually exclusive groups ($k \leq n$) that is optimal with respect to the internal cohesiveness and the external isolation of these groups.

Throughout this study, the above stated problem will be referred to as 'the central problem'.

The terms internal cohesiveness and external isolation are due to Cormack (1971) and they need some clarification here.
The concept of internal cohesiveness refers to the pooled within-groups variance. The internal cohesiveness is maximized by minimizing the pooled-within groups variance. We use the criterion of internal cohesiveness to ensure that groups (= clusters) are as tight as possible.
External isolation can be expressed in terms of between-groups variance. By maximizing the total between-groups variance, we are looking for clusters that are as far apart as possible.
In general, internal cohesiveness and external isolation are desirable properties of cluster solutions. Figure 1.1 is copied from Gordon (1981) and illustrates the two concepts.

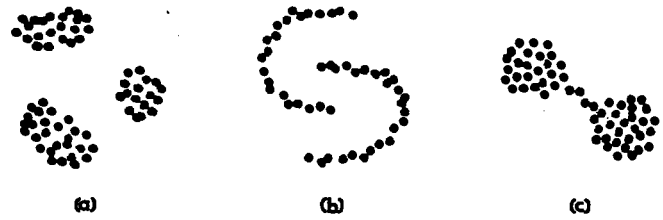

Illustration of the concepts of the cohesion and isolation of clusters: (a) clusters are cohesive and isolated; (b) clusters are isolated but not cohesive; (c) two cohesive clusters which are linked by several intermediate points.

FIGURE 1.1

Another point of the central problem is the use of categorical variables, because many data of social science research can be seen as categorical measures.

The purpose of this study is to develop a method to solve the central problem.
The study yields two products. Firstly this paper, that accounts for the theoretical considerations of the proposed method. Secondly a computerprogram, named GROUPALS, to apply the method to actual data.

1.2 Summary of contents

This section provides a short description of the contents of the report.

Chapter 2 discusses a number of commonly used approaches to related problems, mainly from the field of cluster analysis.
In chapter 3, we translate the concepts of homogenity and discrimination into a suitable loss-function. Furthermore, a procedure to minimize the value of this function is proposed.
Chapter 4 deals with the implementation and testing of the program.
In chapter 5, some illustrative examples are given.
Finally, chapter 6 concludes the report and provides some topics for further research.

An appendix, containing the necessary information to operate the GROUPALS program, is inserted at the end of the paper.

CHAPTER TWO

CLUSTER ANALYSIS

The first thing to notice regarding the central problem of this study is the fact that we want to find a (fixed) number of groups. We therefore shift our attention to the field of cluster analysis.

Cluster analysis can be looked upon as a two-stage procedure (Gordon, 1981).
The first stage consists of the preparation of the raw data. Section 2.2 deals with this phase.
The second stage is the actual application of a particular clustering technique. Section 2.1 describes various forms of cluster analysis and their use.
In section 2.3 we discuss how the central problem would have to be solved by using current methods.

2.1 Clustering techniques

> This section describes various forms of cluster analysis and their major characteristics.

Cluster analysis attempts to solve the following problem (Everitt, 1980):

> Given a number of objects or individuals, each of which is described by a set of measurements, device a classification scheme for grouping the objects into a number of classes such that objects within classes are similar in some respect and unlike those of other classes. The composition of any groups is not known at the start of the investigation.

Cluster analysis passes under various names: Q-analyis, typology, grouping, clumping, classification, numerical taxonomy, unsupervised pattern recognition and unsupervised learning.

Gordon (1981) distinguishes four principal forms of cluster analysis, viz. hierarchical, partition, clumping and geometrical methods. All four methods shall be discussed briefly.

In hierarchical techniques classes themselves are classified into groups. Repeating the classification process at different levels yields a hierarchically nested series of groups, called a tree. There are various hierarchical clustering schemes. Differences between these schemes arise because of different ways of defining the distance between an individual and a group containing several individuals, or between groups of individuals.
In general, the splitting and merging strategies are designed optimally at each level. However the partition at a given level need not be the best possible (Gower, 1967). A common mistake in using hierachical techniques is to view the partition at some level as the best available.
Hierarchical techniques are often useful when attention is focussed on hierarchical inter-dependencies among groups and individuals, that is, when the interest lies largely in the overall tree structure.
Most applications come from the field of biological taxonomy.

Partitioning methods (also called optimization techniques, named after the way in which the partition problem is solved) try to find a partitioning for a given number of groups, which is optimal with respect to a criterion. The classes are mutually exclusive, thus forming a partition of the set of individuals.
The criterion is usually specified to minimize some sort of intra-cluster distance function. The most popular clustering criterion is the minimization of the trace of the pooled-within group matrix of sums of squares and cross products (Blashfield, 1976).
An advantage of partitioning methods is that they permit relocation of objects, that is, objects can be allocated to several clusters several times, before the final solution is reached. On the other hand problems arise with respect to local minima and starting configurations.

Gordon's third form of cluster analysis is the collection of the so called clumping methods. As in partitioning methods most clumping methods are designed to minimize an objective function. However, in clumping methods the restriction that groups should be mutually exclusive has been dropped. Thus, the groups are allowed to overlap. Such overlapping groups are called clumps.
There are a number of cases in which classification must permit overlap between the classes if it is to be of any value. For example, in language studies, words may have several meanings. If

they are being classified through their meaning, they may belong to several groups.

Geometrical techniques (or ordination methods) make up the last kind of cluster methods. Geometrical methods are not real cluster methods in that they do not provide us with direct information concerning group membership. They merely seek a low-dimensional representation of objects, in which the objects that are similar to one another are represented by points that are close together. The representation is assessed by eye in an attempt to establish whether or not the points fall into distinct, well separated clusters.
Techniques that are employed are Principal Components Analysis, Multidimensional Scaling, Multidimensional Unfolding and Correspondence Analysis.
For geometrical methods, it is essential that the bulk of relevant information is retained in the representation. With some sets of data these methods may not give an adequate representation in two or three dimensions, and so visual examination may not be possible.

If we crossclassify these four methods by the way they work to find the optimal group partition, we obtain table 2.1.

algorithm \method	hier	part	clum	ordi
optimization	A	+	+	B
heuristics	+	-	-	+

TABLE 2.1

The '+' indicates that the method in the corresponding cell is used frequently. Thus, partioning and clumping methods are normally solved by optimization algorithms; the partition of objects found by hierarchical and ordination methods usually rely on heuristics. The cells marked 'A' and 'B' refer to methods that are relatively new and that are in a process of growth. The paper by De Soete et al. (1984) is a major contribution in the development of hierarchical optimization methods.
This study is mainly concerned with the cell labeled 'B'. The method to be proposed here can be seen as an ordination method that is bounded by a cluster structure on objects.

It will be clear that the theory of partitioning methods can provide useful insights in studying the central problem. We will discuss partitioning methods and their associated problems in

chapter four.
The next section is devoted to the first phase of clustering.

2.2 Similarities and spaces

> This section deals with two basic forms of cluster analysis input: similarities and spaces.
> It is concluded that for our purposes spaces are more useful than similarities.

Many classification methods require the data to be presented as a set of proximities. Others implicily use a particular distance measure (e.g. squared euclidian distances), and these require the objects to be represented in some sort of (orthogonal) coordinate system. We can thus distinguish two kinds of cluster input: similarities and spaces. As said in the introduction of this chapter, the conversion of raw data into a form that is suitable for a particular technique is the first phase of clustering.
In this section we deal with matters concerning this first phase.

The majority of clustering techniques begin with the calculation of a matrix of (dis)similarities or distances between objects. The number of proposed methods regarding this calculation is large. For example, CLUSTAN (Wishart, 1978), the most widely used cluster package, offers a choice of 38 similarity measures.
And so, the main problem associated with proximities becomes evident, namely the question which distance measure should be used. Obviously the output of the clustering technique will only be as meaningful as the input similarities are.
Because different similarity measures may have different values for the same set of data, the problem of choosing an adequate similarity coefficient is a serious one. Everitt (1980) shows for the binary case, that some coefficients aren't even monotonically related.
Leading texts on clusters analysis devote many pages to problems regarding (particular) measures. Two of these difficulties will be glanced at hereafter.

The first of these problems is related to the measurement level of the input variables. Different coefficients for different measurement levels have been proposed. The problem is, besides the arbitrary choice of a particular measure, that coefficients for

different measurement levels can not be easily compared. Thus, when the columns of the original datamatrix consist of variables with mixed measurement levels, similarities cannot be computed adequately.
Gower (1971) developed a general similarity coefficient for the mixed levels case. However, this measure makes no provisions for ordinal variables.

A second problem regarding the use of similarities is the issue of standardizing the input variables. Standardization is usually undertaken to obtain compatible units of measurement. Typically, one transforms each variable to Z-scores.
Cronbach & Gleser (1953) however pointed out that averaging and standardizing variables eliminates the 'elevation' and 'scatter' differences between object profiles. In general, 'elevation' and 'scatter' information is relevant. Therefore, they advice against the use of standardization for most practical applications.
Thus, because most similarity coefficients involve some kind of 'averaging the variables', we run the risk of discarding valuable information.

Some other problems with similarities are missing values, conditionally present variables, and weighting of variables. For these issues, the reader is reffered to Gordon (1981).
Furthermore, when the number of objects is large, the computation of a matrix of (dis)similarities becomes impracticable.

In the introduction of this section we stated that there are some clustering methods that can operate without the need to calculate a proximity-matrix. These methods usually rely upon the assumption that objects can be adequatly represented in some sort of metric space. Euclidian spaces with low dimensionality are preferred.
Of course, the problem now is to determine these spaces. At least, we must deal with the above mentioned problems of measurement level and standardization. We therefore resort to the optimal scaling theory (Nishisato, 1980; Tenenhaus & Young, 1985).
The theory of optimal scaling enables us to raise the measurement level of binary, nominal and ordinal variables to numerical level. The process works by carrying out restricted transformations of qualitative variables, such that the value of a loss function is minimized.
The restrictions put on the transformations correspond to the measurement levels of the variables. By selecting the appropriate loss fuction we can generate a low dimensional euclidian space of objects. In chapter three we describe the optimal scaling approach in more detail.

In comparing proximities and spaces we note that using spaces has some advantages. For example, we can construct spaces of objects that are optimal with respect to measurement level and scales of variables. For proximities there is no way of achieving this. Furthermore, spaces allow us to put restrictions on them. We can do well out of this fact, because it enables us to treat the measurement level and standardizing problems in a more formal way. This will be done in chapter three.

2.3 Approaching the problem

> This section discusses some approaches to solve the central problem.

We can approach the central problem in a number of ways by using some common methods. But before doing this, we discuss some crucial elements of the problem.

In the central problem, there are three keypoints that ask for our special attention. First, we want to find a fixed number of groups. In the second place, these groups have to be internal cohesive, external isolated and non-overlapping. Third, the input variables are of nominal, ordinal or numeric type, or of any mix of these types.
We consider these three keypoints as desiderata or as criteria by which we can judge a clustering technique by its aptitude to solve the central problem. Below, these desiderata and their implications will be described in more detail.

The first desideratum is that the clustering technique must provide us with a fixed, that is, specified in advance, number of classes. Thus, the cluster method must be able to form this number of classes. Most cluster methods manage to do this.
In many practical situations, the investigator may not have any idea of how many clusters are present in the data. Many methods to determine the number of cluster have been proposed. Milligan & Cooper (1985) revised 30 methods and concluded that no completely satisfactory solution is available.
The 'number of clusters' problem is beyond the scope of this study and instead we assume that the number of clusters is known in advance.

The second desideratum is related to three properties of the found classes.
First, the classes should be mutually exclusive. Only hierarchical and partitioning techniques yield non-overlapping clusters, so our scope becomes limited to these two techniques.
The second property, internal cohesiveness of clusters, is much more restrictive. One way to express internal cohesiveness in terms of variances is to minimize the criterion *trace*(W). The matrix W is the pooled-within group scatter matrix. This clustering criterion was suggested by Edwards & Cavalli-Sforza (1965) and Singleton & Kautz (1965).
Methods that attempt to minimize *trace*(W) are proposed by Forgey (1965), Jancey (1966), Macqueen (1967) and Ball & Hall (1965). These methods are all partitioning methods and they rely on iterative optimization of the *trace*(W) criterion. The K-means algorithm (Macqueen, 1967; Hartigan, 1975) is one of the most popular methods. Bayne et al. (1980) examined a number of selected cluster procedure and found the K-means algorithm to be among the best.
The third cluster-related property is external isolation. External isolation means that we want the clusters to be as distinct as possible from each other. Obviously, cluster separation is very much affected by the spread of the input variables. Therefore, it could be useful to construct new variables by determining linear combinations of variables, that have maximum variance, and to use these new variables as cluster analysis input. A method to construct such new variables is Principal Components Analysis (PCA).

The third desideratum of the clustering technique is its aptitude to various and mixed measurement levels. As we pointed out in section 2.2 we advocate the use of spaces instead of similarities. Spaces enable us to build in non-metric generalizations into the first phase of cluster analysis.
A common way of deriving spaces is to apply PCA to the data matrix. Several generalizations of PCA have been made (Kruskal & Shepard, 1974; Tenenhaus, 1977; Young, Takane & De Leeuw, 1978; Gifi, 1981), and the associate programs can handle mixed measurement levels. In principle, any non-metric technique that creates low dimensional euclidian spaces can be used. The advantage of using non-metric PCA (NMPCA) is that it seems to serve two purposes. First, it is a vehicle to get around with different measurement levels; second, its property of maximizing variance suits the criterion of external isolation.

We are now able to review a few current methods to solve the central problem. Table 2.2 summarizes some possible ways to handle the problem.

	METHOD	intern cohesi	extern isolat	measur levels	fixed number
1	RD - prox - hierachical	no	no	no	yes
2	RD - prox - Kmeans	yes	no	no	yes
3	RD - PCA - (prox) - Kmeans	yes	yes	no	yes
4	RD - NMPCA - visual insp.	no	yes	yes	no
5	RD - NMPCA - Kmeans	yes	yes	yes	yes

TABLE 2.2

All methods in table 2.2 find non-overlapping groups. Hierachical methods do not provide an optimal partition of objects at a given level, so they are unsuitable for solving the problem of this study. Methods 2 and 3 both suffer from measurement level problems. In method 4 the problem is that visual inspection may not be able to recover the preset number of groups; no structure is imposed on the dataset.
Finally, method 5 seems an adequate way of handling the problem. However, the scaling of variables in the NMPCA phase is only optimal with respect to the loss function for the first p principal components, and not with respect to the derived group allocations. Thus, it is possible that a variable with much potential discriminatory power could be scaled in such a way that most of this power is lost. Clearly, losing discriminatory information is not desirable.

We can overcome this problem if we minimize the value of a loss function that both incorporates loss caused by maximizing variance of the first p principal components, and loss caused by the group allocations of objects. By minimizing such a loss function, we simultaneously scale variables and cluster objects; and so the two-stage procedure of cluster analysis collapses to one single phase.

The remainder of this study is concerned with the specification and minimization of the above mentioned type of loss function. Chapter three deals with the description of the loss function.

CHAPTER THREE

THE THEORY OF GROUPALS

GROUPALS is built from two elements of the Gifi system, namely the optimal scaling approach and the PRINCALS loss function. These will be discussed in section 3.1.
Section 3.2 introduces the GROUPALS minimization problem. The GROUPALS loss function is the same as the one used in PRINCALS, but minimization is subject to different constraints.
In section 3.3 we deal with matters concerning the normalization of the solution.

3.1 Optimal scaling and PRINCALS

> This section discusses the Gifi approach to mixed measurement levels and introduces the PRINCALS loss function.

Most classical multivariate techniques are based upon the assumption that the input variables are at least measured on a interval scale. So, if we want to use these techniques (and we do) for the nominal and ordinal measurement levels, we have to transform the scale of the variables to interval level. Because there is an infinite number of ways to ascribe numbers to categories, we need some criterion to choose among all possibilities.
In the remainder, we will adopt the criterion of optimal scaling.

Gifi (1980,1981) describes a system of multivariate techniques for categorical data. The central issue of the Gifi-system is the optimal scaling of variables, such that a loss function is minimized.

Optimal scaling of a variable can be represented as a transformation problem:

$$t_j[h_j] = q_j \ . \qquad (3.1)$$

Here, h_j is the j'th raw data vector, and q_j is the vector of

optimally scaled values (quantifications) for h_j. The problem is to find the transformation function t_j.
It will be clear that t_j is subject to some measurement constraints.

First of all, we want the objects that fall in the same category to have the same scaled value. Thus t_j has to satisfy

$$t_j: (h_{ij} \tilde{} h_{kj}) \rightarrow (q_{ij}=q_{kj}), \tag{3.2}$$

where $\tilde{}$ indicates empirical equivalence (i.e., membership of the same category), where h_{ij} is the score and where q_{ij} is the quantification for object i.

In addition to (3.2) ordinal variables t_j have to satisfy

$$t_j: (h_{ij}<h_{kj}) \rightarrow (q_{ij} \leq q_{kj}) \ . \tag{3.3}$$

Constraint (3.3) preserves the order of the categories, but untied objects may become tied.

Numerical (interval or ratio) variables require that the real numbers assigned to the observations must be functionally related to that observations. For example, quantifications and raw observations could be related by some polynomial rule:

$$t_j: q_{ij} = \sum_{p=0}^{r} b_p h_{ij}^p \tag{3.4}$$

If r=2 we have a quadratic relationship between the optimally scaled and the raw observations.

In De Leeuw, Young & Takane (1976) a system of measurement and process levels is discussed, which can be used to define many different types of <u>cones</u> K_j in a k_j-dimensional space, where k_j is the number of categories for variable j. Usually, k_j is much smaller than the number of observations.
The cone K_j sets up the feasible region in which all scaled values q_j must lie in order to satisfy the measurement constraints of variable j. Let us define

$$q_j = G_j y_j \ , \tag{3.5}$$

where y_j is a k_j vector of category quantification for variable j and where G_j is the indicator matrix of the raw data vector j (see Gifi, 1980). So

$$y_j \in K_j \qquad (3.6)$$

can be used as a general formulation for all measurement constraints concerning variable j.

The values of the elements of y_j can be found by means of optimal scaling. The goal of optimal scaling is to find optimal transformations t_j for each variable j by minimizing a loss function, while satisfying (3.6) for all j's.

For our purposes the PRINCALS loss function is of interest. The PRINCALS computerprogram is designed to find a specified number of principal components from categorical data by means of an alternating least squares algorithm. PRINCALS minimizes the following loss function:

$$\sigma(X;Y_1,\ldots,Y_j,\ldots,Y_m) = 1/m \sum_{j=1}^{m} \mathrm{tr}(X-G_jY_j)'(X-G_jY_j) \qquad (3.7)$$

Minimization of $\sigma()$ takes place over the $k_j \times p$ matrices Y_j (the multiple category quantifications) and the $n \times p$ matrix X (the object scores), where k_j is the number of categories of variable j, n is the number of observations, and p is the dimensionality of the problem. The $n \times k_j$ matrices G_j are indicator matrices, which represent the input data.
Note that we use Y_j for multiple category quantifications (i.e. quantifications in more than one dimension), whereas we reserve lowercase y_j for single quantifications.

The PRINCALS program minimizes (3.7) with normalizations $X'X=I$ and $u'X=0$, and with condition $y_j \in K_j$. The algorithm estimates $Y_j (j=1,m)$ while holding X fixed, and vica versa, until some preset convergence criterion is reached.
The condition $y_j \in K_j$ is satisfied by performing simple regression (numerical), weighted monotone regression (ordinal) or centroid scaling (nominal) for each variable seperately during the Y_j estimation phase.

The PRINCALS loss function and algorithm will be the starting point of the next section. By defining a suitable extra restriction on this loss function, we can create a procedure that finds clusters of objects.

3.2 GROUPALS

> This section is concerned with the specification of the GROUPALS loss function. The loss function can be split up into two independent parts.

In (3.7) X represents a n×p matrix of objects scores. If p=2 we can plot the object scores in a two-dimensional object space. Figure 3.1 is such a plot of a hypothetical sample of objects.

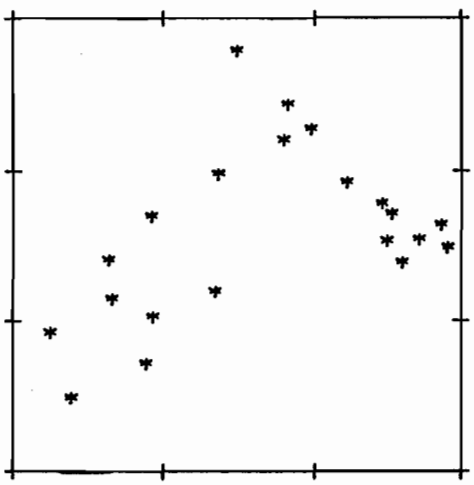

FIGURE 3.1

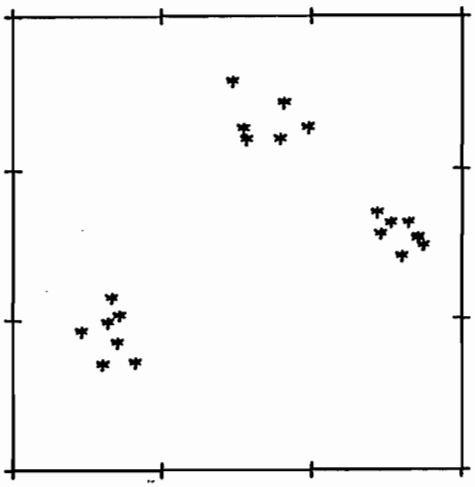

FIGURE 3.2

The objects are scattered in a horse-shoe form. Suppose we are interested to discriminate among three groups of objects, then fig. 3.2 would be more useful for our purposes; we can easily distinguish three groups of objects.
Unfortunately, many multivariate techniques do not provide with such easy-to-interprete pictures like 3.2. Rather, we would get something like 3.1, if we are lucky, or worse.
However, if we have an idea of the number of groups that are present in the data, we can force the objects to cluster into that number of groups. We achieve this by specifying an extra restriction on the object space, namely the restriction that all objects in the same group must lie on the same place (the cluster mean) in the object space.

In other words, we require X to be a quantified categorical variable (Heiser, 1986):

$$X = G_c Y_c \ , \tag{3.8}$$

where G_c is a $n \times k_c$ binary matrix of allocations (that is, an indicator matrix of group memberships), where Y_c is a $k_c \times p$ matrix of locations (the cluster means) and where k_c is the preset number of groups. The $_c$ mnemonic stands for categorical. Both G_c and Y_c are initially unknown.

The data-analytical problem now is to minimize loss function (3.7) under the constraints (3.6) and (3.8). This problem will be referred to as the GROUPALS minimization problem.

We will now pay attention to the mathematical elaboration of this minimization problem.

If we define D_c by

$$D_c = G_c' G_c \tag{3.9}$$

then D_c is a diagonal matrix, since we require the groups to be mutually exclusive. The diagonal cells of D_c are filled with the total number of objects per group.
Furthermore, let $\underline{X}$ be defined as

$$\underline{X} = 1/m \sum_{j=1}^{*} G_j Y_j \tag{3.10}$$

then (3.7) can be reexpressed as

$$\sigma(X; Y_1, \ldots, Y_j, \ldots, Y_*) = \sigma(\underline{X}; Y_1, \ldots, Y_j, \ldots, Y_*) + \mathrm{tr}(\underline{X}-X)'(\underline{X}-X) \ . \tag{3.11}$$

The matrices Y_j (j=1,...,m) contain the category quantifications, $\underline{X}$ is the matrix of centroids of Y_j (j=1,...,m) and X is the expanded array of cluster means.

Notice that formula (3.11) allows us to minimize its left hand side by alternatingly minimizing the two components of its right hand side.
The first component is nothing more than the PRINCALS loss function (3.7). Its goal is to find optimal $Y_1, \ldots, Y_j, \ldots, Y_*$ for a given X. The procedure to find these quantifications is described in Gifi (1981) and will not be further discussed here.
By minimizing the second component we force each object to be as

close as possible to its cluster mean. If we insert (3.8) the remaining problem is to minimize

$$\sigma_c(G_c;Y_c) = \mathrm{tr}(\underline{X}-G_cY_c)'(\underline{X}-G_cY_c) \ . \qquad (3.12)$$

The major question now is to find the group allocation matrix G_c. For any fixed G_c the cluster means can be found by

$$Y_c = D_c^{-1}G_c\underline{X} \ , \qquad (3.13)$$

i.e., the centroids (k_c-dimensional means) of objects belonging to the same group. It is convenient to reexpress (3.12) in the form

$$\min_{g_i \in I \ (i=1,n)} \sum_{i=1}^{n} (\underline{X}_i - Y_c'g_i)'(\underline{X}_i - Y_c'g_i) \ , \qquad (3.14)$$

where $\underline{X}_i$ is the i'th row of $\underline{X}$, where g_i is the i'th row of G_c, and where the notation $g_i \in I$ indicates that g_i must be chosen from the columns of the identity matrix. The solution of (3.14) can be obtained by solving n distinct subproblems, in each of which we have to allocate point i to the cluster with the closest centroid.

Surprisingly, alternating between (3.13) and (3.14) is the K-means method applied to $\underline{X}$. We have seen the K-means method before in chapter two.
Thus we can use K-means to minimize the second part of the GROUPALS loss function. In chapter four we discuss two implementations of the K-means method.

The remainder of this chapter will be concerned with a rather technical topic. Section 3.3 will deal with the normalization of the solution. Normalization is undertaken to match minimization procedures and to avoid trivial solutions.

3.3 Normalization

> This section discusses some problems regarding the normalization of the GROUPALS loss function. A two-step procedure is proposed as a solution.

The value of the GROUPALS loss function is minimized over the object scores X and over the quantifications $Y_j (j=1,\ldots,m)$. In

order to prevent the algorithm from making all X and Y_j zero, X or Y must be normalized.

The PRINCALS normalization convention is to require $X'X=I$. By inserting constraint (3.8) this convention transforms into

$$Y_c'D_cY_c = I \ , \tag{3.15}$$

which complicates the efficient solution of (3.7). A problem arises when we want to allocate an object to another group. We cannot update the allocation matrix G_c (and D_c) without violating (3.15).

The alternative is to normalize the Y_j (j=1,...,m) matrices. This normalization requires the solution to satisfy

$$\sum_{j=1}^{\bullet} Y_j'D_jY_j = I \ . \tag{3.16}$$

By using normalization (3.16) we can freely update the G_c matrix. However, if we wish to update the Y_j matrices from X, we encounter similar difficulties as by using (3.15).
The principle is that only the unnormalized part of the solution can be updated.

Clearly, neither (3.15) nor (3.16) are suitable normalizations for the GROUPALS problem.
A way out of this difficulty is to apply a <u>temporary</u> rescaling, as is done in the canonical correlation method proposed by Van der Burg and De Leeuw (1983).
The idea is to switch between the normalizations (3.15) and (3.16), while <u>preserving</u> the loss between the X- and Y_j- configurations.
The advantage of such a procedure is that we are able to estimate both Y_c and Y_j (under different normalizations) within the same procedure.

The problem now is to find transformation matrices such that the loss is preserved.

Suppose $(X;Y_1,\ldots,Y_j,\ldots,Y_m)$ is any candidate solution with $X'X = I$ and the $Y_1,\ldots,Y_j,\ldots,Y_m$ is unnormalized, then there exist matrices A and B such that

$$\sigma(X;Y_1,\ldots,Y_j,\ldots,Y_m) = \sigma(XA;Y_1B,\ldots,Y_jB,\ldots,Y_mB) \tag{3.17}$$

and where $\underline{Y}_j = Y_jB$ (j=1,...,m) satisfies

$$1/m \sum_{j=1}^{*} \underline{Y}_j' D_j \underline{Y}_j = I \qquad (3.18)$$

The expression XA represents the unnormalized X, the $\underline{Y}_j$ stands for the corresponding normalized category quantifications.
The normalization procedure now involves the shifting between the left and right hand of (3.17). We normalize the temporary solution according to the left hand side (X normalized) if we want to estimate the Y_j matrices; we transfer the normalization to the category quantifications if we want to update the object scores (i.e. by assigning the cluster means to the object scores).

The tranformation matrices can be obtained by means of the following procedure:

(a) for any set of unnormalized $Y_1, \ldots, Y_j, \ldots, Y_*$, find the eigenvector/eigenvalue decomposition

$$K\Lambda^2 K' = 1/m \sum_{j=1}^{*} Y_j' D_j Y_j ,$$

(then we can identify $A = K\Lambda$ and $B = K\Lambda^{-1}$).

(b) Use

$$\underline{X} = 1/m \sum_{j=1}^{*} G_j \underline{Y}_j = 1/m \sum_{j=1}^{*} G_j Y_j K \Lambda^{-1}$$

for the minimization (3.12) over X unnormalized.

(c) Suppose the clustering result is $\underline{X}^* = G_c^* Y_c^*$; next compute the decomposition

$$L\Phi^2 L = (\underline{X}^*)' \underline{X}^*$$

(d) Use

$$X = \underline{X}^* L \Phi^{-1}$$

for the minimization over $Y_1, \ldots, Y_j, \ldots, Y_*$ unnormalized (but satisfying the usual PRINCALS measurement restrictions).

(e) Compute stress and go back to (a) if its rate of change is still larger than some predetermined stopvalue.

In the testing phase of this procedure, the following difficulty was encountered. In case of single variables (single nominal or

single ordinal) the minimization over $Y_1,\ldots,Y_j,\ldots,Y_m$ in step (d) sometimes increases the loss function value and this will corrupt proper convergence.
To solve this difficulty, we have to consider that category quantifications for single variables are subject to a rank-one restriction. This restriction

$$Y_j = y_j b_j' \ , \tag{3.19}$$

can be satisfied by minimizing

$$\sigma(y_j;b_j) = \mathrm{tr}\ (y_j b_j' - Y_j^*)' D_j (y_j b_j' - Y_j^*) \tag{3.20}$$

over $y_j \in K_j$ and over b_j. Y_j^* is defined as

$$Y_j^* = D_j^{-1} G_j' X \ . \tag{3.21}$$

For the PRINCALS algorithm the value of (3.20) is always lowered by one inner iteration, during which y_j and b_j are estimated. This is not the case for the GROUPALS algorithm, because of a possible rotation of the complete solution in step (c).
One way to solve this is to allow more inner iterations to estimate y_j and b_j. A more efficient method however is to adapt the b_j-vectors (j=1,...,m) to the current normalization.
Therefore, we extend the normalization procedure with two substeps. To step (b) we add

$$b_j^* = b_j K \Lambda^{-1} \qquad (\text{for } j=1,\ldots,m)$$

and to (d) we add

$$b_j^{+} = b_j^* L \Phi \qquad (\text{for } j=1,\ldots,m)$$

We use b_j^{+} as the input component loadings for the minimization over $Y_1,\ldots,Y_j,\ldots,Y_m$.

Now, the sequence of loss function values never increases, so convergence is assured in the usual way.

CHAPTER FOUR

IMPLEMENTATION AND TESTING

The theoretical discussion of the preceeding chapter provides the basis of the GROUPALS computerprogram. In this chapter we account for the implementation and testing of the program.
Section 4.1 summarizes the steps taken by the main algorithm; in 4.2 we focus on one of these steps, namely the clustering step. Section 4.3 reports some results of testruns that were made in order to study local minima.

4.1 The GROUPALS main algorithm

> This section discusses the flow of the GROUPALS main iteration loop.

A computerprogram to minimize the value of function (3.7) with the appropriate measurement- and cluster- restrictions was developed in standard ANSI FORTRAN IV. The appendix contains necessary information on the operation of this program that is called GROUPALS. This section describes the structure of the main iteration loop.

The following arrays have to be defined on entry of the main iteration loop:

G_c - initial cluster allocations
X - initial restricted object scores (normalized)
G_j - indicator matrices (j=1,...,m)
D_j - marginal frequencies of G_j (j=1,...,m)

If the measurement level over all variables is either multiple nominal or numerical, then X is computed from a matrix of random values between -1 and +1, using G_c and a modified Gram-Schmidt orthogonalization. In all other cases, G_c and X result from an initial cluster-restricted numerical SVD of the datamatrix.

The main algorithm can be subdivided into four parts, each of which consists of several steps. These four parts include two estimation

phases and two normalization phases.
Below the algorithm will be described. Each step is accompanied by a clarification of its function. The subscript «s-1» indicates that values of the preceeding iteration are used.

The following steps are executed until the absolute difference of the stress of two consecutive iterations is smaller than some preset criterium value:

--- Estimation: Quantifications over variables ---

(1) $Y_j = D_j^{-1} G_j' X_{(s-1)}$ — Computation of the multiple category quantifications from the restricted, normalized object scores.

Step (2) through (5) are executed for single variables only.

(2) $y_j = P_j(Y_j b_{j(s-1)})$ — The y_j-vector is the projection of $Y_j b_j$ on the cone K_j. This projection is realized through monotone or linear regression for resp. the ordinal and the numerical case.

(3) $z_j = y_j (y_j' D_j y_j)^{-1/2}$ — Standardization of the single datavector such that $z'Dz = I$.

(4) $b_j = z_j Y_j$ — Computation of the updated component loadings b_j.

(5) $Y_j = z_j b_j'$ — Computation of the category quantifications restricted by the single vector approach.

--- Normalization: transfer to category quantifications ---

(6) $T = \sum_{j=1}^{\bullet} Y_j' D_j Y_j$ — Preparation step.

(7) $K = \text{EIGVEC}(T)$
$\Lambda^2 = \text{EIGVAL}(T)$ — Eigenvalue decomposition of T provides for K and Λ.

(8) $\underline{X} = 1/m \sum_{j=1}^{\bullet} G_j Y_j K \Lambda^{-1}$ — Computation of the expanded, unrestricted object scores scaled such that $\sum Y_j' D_j Y_j = I$.

(9) $a_j = b_j K \Lambda^{-1}$ — Rescale the component loadings.

--- Estimation: Cluster allocations ---

(10) G_c = KMEANS($\underline{X}$, $G_{c(t-1)}$) Find the best partition given the new object scores and the old partition.

(11) $Y_c = (G_c'G_c)^{-1}G_c\underline{X}$ Extra computation of the updated cluster means.

(12) $\underline{X}^* = G_cY_c$ Restrict the object scores to the cluster means.

--- Normalization: Transfer to object scores ---

(13) $T = (\underline{X}^*)'\underline{X}^*$ Preparation step.

(14) L = EIGVEC(T) Eigenvalue decomposition of T provides
Φ^2 = EIGVAL(T) for L and Φ.

(15) $X = \underline{X}^*L\Phi^{-1}$ Rescale the restricted object scores such that $X'X=I$.

(16) $b_j = a_jL\Phi$ Rescale the component loadings.

The above algorithm will serve as a reference point for our further discussion of the program.

As we can see in step 10, the algorithm uses K-means inner iterations to find the best partition of objects from the expanded, unnormalized object scores. The next section is concerned with some special characteristics of the K-means subproblem.

4.2 K-means

> This section introduces two versions of the K-means and discusses some characteristics of these algorithms.

In section 3.2 we pointed out that the K-means clustering method can be used as a tool to minimize σ_c. This section discusses the method in more detail and brings up some difficulties.

The K-means algorithm (Macqueen, 1967; Hartigan, 1975) is a nonhierarchical clustering method that belongs to the family of optimization techniques. The algorithm is designed to find that partition of n objects into k groups, that minimizes the value of an objective function. The K-means objective function is the sum of squared distances between objects and the corresponding cluster means (cf. formula (3.12) and (3.14)).

The K-means algorithm is based on the principle of iterative relocation. In its simplest form (which we call 'next-first'), the algorithm consist of the following six steps:

K-MEANS (next first)

(a) Assume initial clusters 1,2,...,k.
Compute the cluster means.

(b) For each object, repeat steps (c) to (e)

(c) Compute the squared euclidian distances between the object and all cluster means.
(d) Allocate the object to the nearest cluster.
(e) Recompute the cluster means.

(f) If no movement of an object from one cluster to another cluster occurs for any object, stop.
Otherwise, go to step (b).

Step (d) is the central (re)allocation step. Notice that updated cluster means are used as the input for the next iteration. The above algorithm thus converges towards a local minimum.

Hartigan (1975) suggests some variations on the above algorithm. A particularly interesting modification is to determine that object

that decreases the objective function most, prior to performing the reallocation step. This 'best-first' algorithm consists of the following steps:

K-MEANS (best first)

(a) Assume initial clusters 1,2,...,k.
Compute the cluster means.

(b) For each object, repeat steps (c) to (e)
(c) Compute the squared euclidian distances between the object and all cluster means.
(d) Determine the reduction in total distance if the object is reallocated to the nearest cluster.
(e) If this reduction is the largest sofar over all objects, store the object identification.

(f) Reallocate the identified object to the nearest cluster.

(e) Recompute the cluster means.

(g) If no movement of an object from one cluster to another cluster occurs for any object, stop.
Otherwise, go to step (b).

The best-first K-means algorithm requires more computational effort than the next-first version, because of the inclusion of the object selection steps (c) to (e). On the other hand, the best-first algorithm may be more effective in avoiding local minima. To test this supposition, we have to compare both versions on a considerable number of solutions, each of which has a different starting allocation. We will do this in the section 4.3.

Below, we set out three properties of both algorithms and the relation they have with GROUPALS.

First of all, the K-means algorithm will find a local optimal group allocation, and not necessarily a global optimum. In principle, it is possible to examine all possible partitions for the global minimum value of the objective function. However, even for small problems this would require enormous computational effort. Gower (1967) estimated that a global optimal partitioning of 41 subjects into 2 groups would require 540 years of time on the fastest computer available at that time. In mathematical programming, the allocation subproblem of GROUPALS is classified as a NP-hard assignment problem, which does not possess a solution other than

explicit enumeration.
Which local optimum will be found largely depends upon the chosen starting allocation (Milligan, 1979). Different initial clusters may lead to different solutions. Much controversy exists on how to select the initial allocation. We will study the sensitivity of GROUPALS to local minima in section 4.3.

A second characteristic of K-means is that it needs a specification of k, the number of clusters. It will be clear that different k's can cause quite different solutions.
If one has no clearcut idea of how many clusters are present in the data (e.g. data-exploration), the problem is to choose a reasonable value of k. Some partitioning methods (Ball & Hall, 1965; Macqueen, 1967) allow k to vary using splitting and merging strategies. However, the problem then shifts to the specification of minimum and maximum cluster sizes.
For the central problem, we treat k as fixed. To assess the validity of the found allocations, we adopt the SILHOUETTES graphic method developed by Rousseeuw (1984). This method establishes a visual representation of a given partitioning and can, amongst others, be used to diagnose bad clusters.
The appendix contains a description of the operation and use of SILHOUETTES.

Third, K-means implies assumptions about the shape of clusters. The method always finds spherical clusters, even if natural clusters in the data are of another (e.g. elliptical) form. The shape preference of K-means is a major issue in determining the form of the input data.
In GROUPALS, the K-means clustering is applied to the unnormalized, expanded object scores $\underline{X}$, which are computed in step 8 of the main algorithm. In this step, the coordinates of the object scores are multiplied by the square root of the eigenvalues Λ. Postmultiplying Λ has the effect of magnifying each dimension proportional to the square root of its eigenvalue.
Because K-means finds spherically shaped clusters, the contribution of each dimension to the overall clustering solution is directly related to its eigenvalue: the higher the eigenvalue, the more the effect is on clustering.
Thus, for the central problem, the K-means shape preference can be built in nicely into GROUPALS by relating eigenvalues and clustering effect.

The next section is concerned with the difficulty of local minima found by K-means.

4.3 Starting allocations and local minima

> In this section the next-first and best-first algorithms are compared. It is concluded that no method is superior to the other.
> Furthermore, local minima are of frequent occurrence. Additional study is necessary.

In section 4.2 we pointed out that the K-means algorithm does not guarantee the found partition of objects to be globally optimal. In this section we study the convergence properties of the two K-means versions by computing a number of solutions with different starting allocations. Furthermore, we estimate the deviation of the local optimal partition from the global allocation.

As said in section 4.2, the starting cluster allocation determines which local optimal partition will be found.
The initial allocation can be set up in a number of ways (see Thorndike, 1953; Friedman & Rubin, 1967; Macqueen, 1967; Hartigan, 1975). Some recent evidence (Scheibler & Schneider, 1985) indicates that the K-means method performs significantly better if the initial allocation was being constructed by a robust hierarchical technique (e.g. Ward's method).
For our problem however, most of these initialization methods are unsuitable because they assume that (dis)similarities between objects are known in advance. In GROUPALS, the object scores and the optimal partition are alternatingly estimated, and thus we do not have similarities to our disposal at the initialization phase.
For GROUPALS we use one of the simplest approaches, due to Späth (1980). Späth advices to compute the initial partition by

$$G_c(i) = \mathrm{MOD}(i,k) \quad (i=1,\ldots,n) \ ,$$

where $G_c(i)$ indicates the initial cluster allocation for object i, where n is the number of objects and where k is the number of clusters. To aviod local optimal solutions, Späth recommends to repeat the analysis a number of times with a different ordering of objects and to select the best partition.
Späth's allocation strategy provides for clusters which are initially nearly equal in size. By randomizing the row-order of the datamatrix we obtain distinct initial cluster estimates.

Both next-first and best-first versions of K-means were being implemented in GROUPALS. In order to study the appearance and nature of local minima the program was applied to a testdataset consisting of 118 objects and 7 variables, each of which contained five categories.
Three conditions were varied: the number of clusters (3, 6 and 15), the dimensionality (2 and 5) and the measurement level parameter (multiple nominal and single ordinal).
Table 4.1 summarizes the results of 100 testruns for some combinations of these conditions. Each testrun was given a pseudo-random permutation of the datamatrix rows.

All fit-values that differ in the fourth digit after the decimal point refer to distinct solutions (i.e. different partitions). If fits differ less than 0.0001, the partitions found by the program are, in general, identical.

6 x 100 testruns -- NEXT FIRST method

di	cl	lev	time /sol.	mean iter	mean fit	st dev	min	max	max freq
2	3	NOM	0.09	6.23	1.054	.158	0.697	1.366	12
2	15	NOM	0.32	11.43	1.502	.037	1.149	1.530	1
5	6	NOM	0.26	6.00	2.022	.094	1.452	2.109	1
2	3	ORD	0.13	5.62	0.764	.004	0.741	0.767	38
2	15	ORD	0.48	16.75	0.846	.002	0.842	0.850	1
5	6	ORD	0.71	14.20	0.893	.019	0.845	0.931	1

TABLE 4.1a

6 x 100 testruns -- BEST FIRST method

di	cl	lev	time /sol.	mean iter	mean fit	st dev	min	max	max freq
2	3	NOM	0.21	6.59	1.137	.200	0.574	1.366	19
2	15	NOM	1.51	11.43	1.511	.013	1.495	1.528	1
5	6	NOM	0.76	5.85	2.019	.113	1.419	2.126	1
2	3	ORD	0.24	5.16	0.764	.004	0.756	0.767	36
2	15	ORD	1.50	18.20	0.848	.002	0.840	0.854	1
5	6	ORD	1.42	17.00	0.897	.016	0.856	0.931	1

TABLE 4.1b

These tables show that the next-first and best-first methods are roughly comparable for the mean, standard deviation, minimum and maximum of the found fits. There seems to be a slight trend in the mean values in favor of the best-first algorithm; however, this trend is not strong enough to make up for the difference in execution time.

The last column of the tables lists the number of times the maximum fit (over 100 runs) was found.
It is striking that only for a low number of clusters (in our case 3), the maximum, perhaps globally optimal, value was found more than once. This is can be partly explained by the fact that the number of possible partitions dramaticly increases with the number of clusters, which makes it harder to single out the optimal partition.
Furthermore, as the standard deviation indicates, more distinct local minima are found in the case of nominal variables. One of the analysis (Next first, 5 dimensions, 6 clusters, nominal) showed a number of 96 (out of 100) different local minima.

From a mathematical point of view one could say that, due to the NP-hardness of the allocation problem, both clustering methods are inadequate to minimize the value of the GROUPALS loss function, and, that the solutions are seriously troubled by the presence of local minima, especially if the number of clusters is large and the variables are nominal.

In practice however, one could argue that the majority of sub-optimal solutions would be quite acceptable if the found partitions are nearly similar to the optimal partition (i.e. only differ in a small number of allocations).
Table 4.2 lists the fitvalues that were found in the analysis that yielded the smallest amount of local minima (118 objects, 2 dimensions, 3 clusters and ordinal level). If we assume that the highest fit value (.7672) represents the global optimum, we are able to compute the number of 'misclassifications' for each local minimum. Table 4.2 also provides the frequency each minimum was found by both clustering methods.

Table 4.2 indicates that there is no monotone relationship between the value of the fit and the number of misclassifications. Fortunately however, 'severe minima' (i.e. minima that have a large number of misclassifications) are found less often than 'acceptable minima'.

fit	mis	nf-fr.	bf-fr.
.7672	0	38	36
.7648	3	39	39
.7622	28	5	4
.7568	16	17	21
.7414	20	1	0

TABLE 4.2

Most of the misclassified objects lie in between clusters. Except for a slight shift of cluster centres, the overall configuration of different solutions look very similar.
The average number of misclassifications was found to be about 5% of the number of objects.

It is questionable if the same results would be obtained for other numbers of clusters and dimensions, measurement levels and data. Further study is needed to generalize these results.

For the time being, we advice to make some provisions in order to avoid the most severe minima. Therefor, we can follow one of two ways: the 'brute force' approach and the 'guided' approach.
By using the 'brute force' approach we simply generate a large number of solutions, each with a different initial allocation, and subsequently pick out the best. GROUPALS has a built in TESTMODE for rapidly generating a large number of solutions.
The 'guided' approach makes use of the notion that deviations from the optimal partition are usually caused by objects that do not tend to cluster very well. These objects can be easily isolated by inspecting the SILHOUETTES. By using the reading and writing options of GROUPALS, we can set the allocations of all objects that have a SILHOUETTE-width lower than a certain value (say .50) to an unknown cluster value (=0) and rerun the program with this partly known initial allocation.
In general, it is useful to provide as much information as possible to set up the starting partition.

We conclude that both K-means versions are sensitive to the initial partition and are likely to produce local optimal solutions. The best-first method does not prove to avoid local minima more effectively, and hence we choose the next-first method as the default clustering algorithm, primarily because it is faster.
We need further study on local minima. For the present, we have to reckon with the possibility that the found partition could depart significantly from the optimal one.

CHAPTER FIVE

EXAMPLES

5.1 Iris data

> In this section the Iris data are used to compare GROUPALS to CLUSTAN and HOMALS.

The Iris data (Fisher, 1936) have become a favorite example for illustrating clustering procedures (Duda & Hart, 1973). In this section we use the Iris data to compare GROUPALS to CLUSTAN and HOMALS.

The Iris dataset consists of measurements of 150 irises, which can be grouped into 3 species: *setosa*, *versicolor* and *virginica*. The obtained measurements are *sepal length*, *sepal width*, *petal length* and *petal width*. Table 5.1 contains the observed means for each species of flowers.

	sepal length	*sepal width*	*petal length*	*petal width*
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

TABLE 5.1

In general, the *setosa* irises can be easily distincted from the other two species; the difficulty is to discriminate between *versicolor* and *virginica*.

GROUPALS and the CLUSTAN RELOCATE procedure (Wishart, 1978) were applied to the Iris data. The goal was to partition the 150 flowers into three groups. The CLUSTAN input dissimilarities were obtained by computing euclidian distances from the first two principal components of the datamatrix. For GROUPALS, the measurement level of all input variables was conceived to be numerical.
Figure 5.1 shows the configuration of objects found by GROUPALS for

two dimensions and three groups. Each object is labeled by its cluster number.

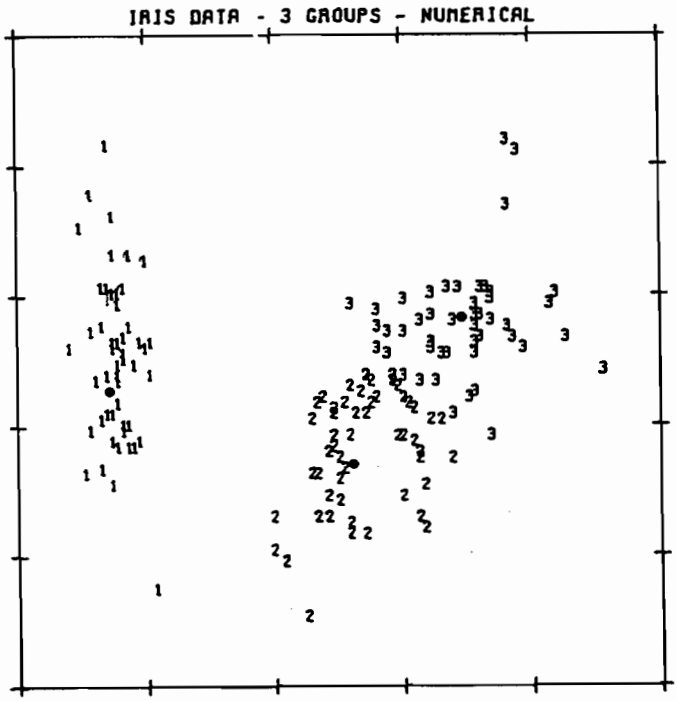

FIGURE 5.1

The fit of the solution is 0.7669. Cluster 1 appears to be separated very well from clusters 2 and 3. It contains all *setosa* irises and none of the other species. For clusters 2 and 3 discrimination is less powerful.

If we compare the partitions found by GROUPALS and CLUSTAN to the botanical typology, it becomes evident that both methods do not exactly reproduce the biological classification. Table 5.2 is a crossclassification of botanical type by both obtained partitions.

	CLUSTAN CLUSTERS				GROUPALS CLUSTERS			
TYPE	1	2	3		1	2	3	
setosa	50	-	-	50	50	-	-	50
versicolor	-	39	11	50	-	38	12	50
virginica	-	14	36	50	-	14	36	50
	50	53	47		50	52	48	

- TABLE 5.2 -

Both clustering procedures classify approximately 5 out of 6 objects corresponding to the botanical type of the object. All

setosa irises are succesfully isolated; clusters 2 and 3 contain a number of objects that are allocated to the 'wrong' cluster. This result indicates that additional discriminating information is needed to identify the *versicolor* and *virginica* flowers more completely.
Note that the partitions found by GROUPALS and CLUSTAN RELOCATE are nearly identical. Only one (!) object was classified otherwise. From this example we conclude that CLUSTAN RELOCATE and numerical GROUPALS are likely to yield partitions that bear great a resemblance.

GROUPALS provides the possibility to cluster on non-numerical variables. We will now analyse the Iris data analogous to the above analysis, except that the variables are defined as multiple nominal. We compare the result to the configuration found by HOMALS.

One could question whether it is correct for this dataset to replace the single numerical by a multiple restriction. We justify the use of multiple quantifications by the following line of thought.
By using single restrictions, the quantification for a variable is, up to a scale factor, identical for all dimensions. Multiple quantifications on the other hand, are computed for each dimension seperately. Now, if one group clearly differs from all other groups in the analysis (as in our example the *setosa* irises), single variable quantifications will highly reflect this difference. Because the same quantifications are being used for all dimensions, the whole solution becomes dominated by the presence of one unique group, which causes the differences among other groups to become overshadowed. If we use multiple quantifications, the first dimension of the solution almost surely discriminates the unique group from the other groups; however, the higher dimensions are left free to discriminate among the others.

HOMALS and GROUPALS were applied to the Iris data. Each variable was recoded into seven, approximately equally filled, categories. Local minima were avoided by selecting the GROUPALS solution with the highest fit over 50 runs, each of which that started with random allocation.
The fits of the solutions are 1.44 for HOMALS and 1.27 for GROUPALS. The difference of the fits (0.17) can be attributed to the extra restriction placed on the object space. The configurations of objects, each labeled by its botanical classification, are shown in the figures 5.2 and 5.3.

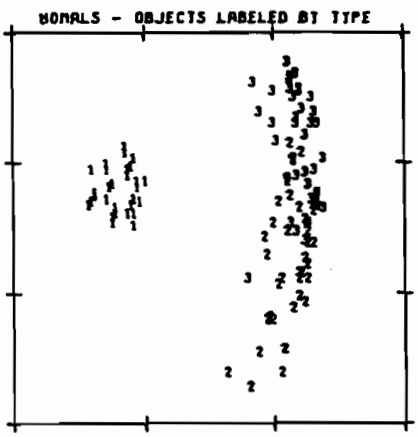

FIGURE 5.2

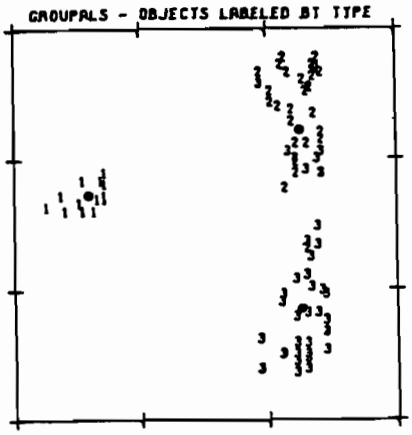

FIGURE 5.3

Both figures show the same trend: group 1 differs from the other groups in the horizontal direction and group 2 is distinct from group 3 mainly in the vertical direction. The difference however is that the groups found by GROUPALS are more pronounced: the loss of fit has been traded in for a much clearer picture.

Furthermore, the partition found by the GROUPALS with nominal variables is substantially closer to the botanical classification than the partition for numerical variables. Table 5.3 illustrates this fact.

	GROUPALS CLUSTERS			
TYPE	1	2	3	
setosa	50	-	-	50
versicolor	-	49	1	50
virginica	-	9	41	50
	50	58	42	

TABLE 5.3

The variables of the dataset are measured on an interval scale. In the above analysis we assumed, for the reasons named, the variables to be nominal. By inspecting the variable quantifications we can assess the violations of a numerical assumption and we can determine if there is any systematic trend in these violations. In figures 5.4 and 5.5 the original categories of the variable *petal width* are plotted against the quantifications.

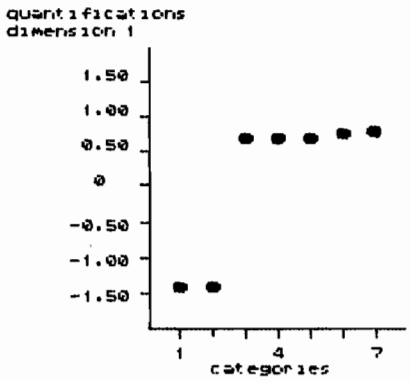

FIGURE 5.4

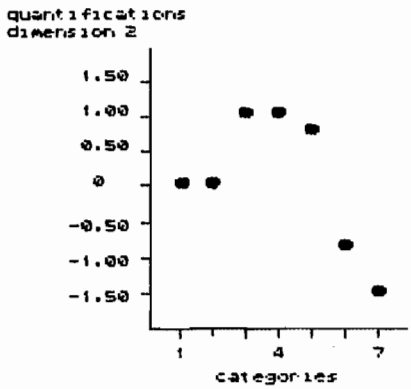

FIGURE 5.5

For the first dimension the category quantifications are a monotone increasing function of the category numbers. Note that there is a sharp distinction between the low and the high category numbers. This distinction is parallel to the difference between the *setosa* and the other species of irises. Furthermore, the quantifications for the higher categories (from 3 on) are nearly similar, so these quantifications do not yield big differences between the other species.
Another situation occurs for the second dimension. Here the main distinction falls in the region of the higher categories. These categories correspond to the *versicolor* and *virginica* species. Thus the species that are similar on the first dimension are discriminated on the second.
The variable is clearly not scaled numerically. However, if we do not take in account the *setosa* quantifications for the second dimension, a multiple ordinal assumption holds.

5.2 Whales data

> In this example the Whales data are used to demonstrate a number of GROUPALS features.

Vescia (1985) lists a dataset of 36 different types of whales. Each type of whale is described by 15 parameters regarding morphology, osteology and behaviour. Most parameters were picked from the GRASSE's zoological treatise (Grasse, 1955).
In this section, we compare the partition obtained by GROUPALS to the partition proposed by Grasse. Furthermore, a feature not seen

in other cluster programs is highlighted. This feature concerns the direct access to the discriminatory power of the input variales.

The 15 variables of the whales dataset are:

V1	*NECK*	*present or not*
V2	*FORM OF HEAD*	*cylindric, conical, flat, convex etc.*
V3	*SIZE OF HEAD*	*medium or big*
V4	*BEAK*	*no beak, large, narrow, long*
V5	*DORSAL FIN*	*no fin, triangular, falciform, backward*
V6	*FLIPPERS*	*small, long, medium, large, narrow*
V7	*TEETH*	*no teeth, on upper/lower jaw*
V8	*FEEDING*	*squish, fish, seal, plankton*
V9	*BLOW HOLE*	*left, right, middle, two holes*
V10	*COLOR*	*blackish, spotted, no pigment, light ventral*
V11	*VERTEBRAE*	*free or welded*
V12	*JUGAL BONES*	*one piece or independent*
V13	*HABITAT*	*rivers, warms seas, cold seas, coasts*
V14	*FURROWS*	*no furrows, small number, large number*
V15	*HEAD BONES*	*symmetrical, unsym., very unsymmetrical*

Grasse classifies the 36 types of whales into nine families or species. These species can be grouped hierarchically to form other partitions, as follows:

Classification given by P. Grasse

	# members
BALEEN WHALES:	
1 BALEEN WHALES: BALAENIDAE	3
2 GREY WHALE: ESCHRICHTIIDAE	1
3 FINBACK WHALES: BALAENOPTERIDAE	3
TOOTHED WHALES:	
PLATANISTOIDEA	
4 RIVER DOLPHINS: PLATANISTIDAE	4
DELPHINOIDEA	
5 DOLPHINS: DELPHINIDAE	14
6 PORPOISE: PHOCAENIDAE	2
7 WHITE WHALES: DELPHINAPTERIDAE	2
PHYSETEROIDEA	
8 SPERM WHALES: PHYSETERIDAE	2
9 BEAKED WHALES: ZIPHIIDAE	5

Two GROUPALS analysis, both based on nine clusters, were made. In the first analysis, the above classification was read as the initial cluster allocation and was held fixed during the analysis by means of the METH parameter (see the appendix). This kind of analysis is called the discriminant approach of GROUPALS: the G_c matrix of cluster allocations is treated as fixed. The result is that the variables are optimally scaled with respect to the given partition. Note that because of the difference in loss functions, the discriminant approach of GROUPALS is not identical to the discriminant option of CANALS.
In the second GROUPALS analysis we employed the K-means sums of squares clustering to obtain nine clusters, starting from a random starting allocation.

Both analyses used a problem-dimensionality of eight: the number of clusters minus one. Although it is possible to specify a lower number of dimensions, we advice to set the dimensionality to the number of clusters minus one. In this case, the number of groups fits exactly in the available space and maximum use is made of the discriminating information of the input variables.

Except for variables 14 and 15, which are single ordinal, all variables are considered to be multiple nominal.
The fits of the analyses are 1.70 (Grasse) and 1.78 (GROUPALS). It can thus be concluded that the classification of Grasse is not the best partition into nine groups.
The principal differences between the cluster allocations of the two solution can easily be detected by inspecting the SILHOUETTE plots given by GROUPALS. It appears that the allocations for the clusters 1,2,3,4,8 and 9 are exactly identical; differences are seen only for clusters 5,6 and 7 (the *DELPHINOIDEA* whales). Figures 5.6 and 5.7 are the SILHOUETTES for these clusters.

From figures 5.6 and 5.7 we can infer that a number of *dolphins* moves from cluster 5 to the *PORPOISES* and *WHITE WHALES* clusters. Apparently, the original cluster of *dolphins* is not tight enough to hold its members. On the other hand, the *PORPOISES* and *WHITE WHALES* do not seem distinct enough from the *dolphins*; and so, these species become mixed up with some of the *dolphins*.

The SILHOUETTE profiles can be used to assess the validity of clusters and to identify objects that do not cluster very well. In our example, the object no. 11 (*GRAMPUS* dolphin) is the least 'clusterable' object in both solutions.