HIERARCHICAL GROUPALS: A SIMULTANEOUS OPTIMIZATION METHOD FOR HIERARCHICAL CLUSTERING USING A MINIMUM VARIANCE CRITERION UNDER OPTIMAL SCALING OF THE VARIABLES

Anita J. van der Kooij

Department of Data Theory
Leiden University

RR-96-07

Preface

This report is a slightly revised version of the author's "master" thesis in Psychometrics and Research Methodology at Leiden University. The revision mainly concerns the order of some topics in chapter 3.

Abstract

GROUPALS uses a partitioning method to cluster objects under simultaneous optimal scaling of variables. In this study the clustering component of GROUPALS is extended to accomodate for hierarchical clustering using a simultaneous optimization method. The method starts from a tree which is built using a successive hierarchical method. Then a within-groups sum of squares criterion is used in optimizing the tree by performing local operations, which results in a locally optimal tree.

Contents

1 Introduction

Cluster analysis is a technique to classify objects which are described by a number of variables into groups without a priori knowledge of group membership, such that objects in one group are similar to each other and different from objects in other groups. Similarity of objects in a group is referred to as internal cohesiveness and difference between groups is referred to as external isolation. A major issue in cluster analysis is the choice of a similarity measure. In social sciences the variables often are of mixed measurement level. Then one problem is the computation of the similarity measure because the similarities on the separate variables can not easily be compared. A second problem regarding similarity measures concerns standardization of the variables. Standardization is often applied to obtain compatible units of measurement. By standardizing the variables, however, relevant scatter information can be lost.

The data that are analyzed by cluster techniques are in an object by object proximity matrix (similarities or dissimilarities) or in an object by variable matrix. Geometrically, in the latter case the scores of the objects on the variables are coordinates of the objects in a space. An advantage of working with a space is that a low-dimensional Euclidean space can be derived while simultaneously dealing with the problems of measurement level and standardization by using optimal scaling (Gifi, 1990). In optimal scaling the level of qualitative variables (binary, nominal or ordinal) is raised to numerical level by applying restricted transformations to the variables, such that the value of a loss function is minimized. The restrictions on the transformations correspond to the measurement level of the variables. Thus, an object space can be constructed that is optimal with respect to measurement levels and scales of the variables. Another advantage of working with a space compared to working with proximities is that a space can be restricted to have a special form, which is a way to handle the clustering problem.

The program GROUPALS (Van Buuren, 1986) was developed to cluster objects, measured on variables with mixed measurement levels, by deriving a low-dimensional Euclidean space using non-linear Principal Component Analysis (PCA) and by putting a restriction on this space to deal with the clustering of the objects. The choice for PCA to derive the object space follows from its property of maximizing variance, which suits the requirement of external isolation. For the non-linear PCA the PRINCALS program (Gifi, 1990) is used, because this program can handle mixed measurement levels. Allocation of the objects to groups can be derived from the PRINCALS solution. However, the scaling of the variables

in PRINCALS is optimal only with respect to the loss function for the first p principal components, not with respect to the derived allocation. In this situation it is possible that a variable with much potential power to discriminate between groups is scaled in such a way that most of this power is lost. To ensure that the scaling of the variables is optimal with respect to both the first p principal components and the clustering, a restriction is put on the object space: the objects are restricted to be located at one of K cluster points. By minimizing a loss function that incorporates both the loss due to maximizing variance of the first p principal components and the loss due to the clustering of the objects, the optimal scaling problem and the clustering problem are treated simultaneously (Van Buuren, 1986; Van Buuren & Heiser, 1989). So, a GROUPALS solution gives two things: the object space and the cluster allocations of the objects.

The cluster technique used in GROUPALS is K-means clustering. The properties of internal cohesiveness and external isolation mentioned above can be expressed as within-cluster variance and between-cluster variance. Minimizing the within-cluster variance is equivalent to maximizing the between-cluster variance (Späth, 1985). K-means clustering is a partitioning method that iteratively minimizes the within-cluster variance.

Partitioning methods are one of the important cluster techniques (Everitt, 1980). These methods are used to find a specified number of clusters. Another important cluster technique is hierarchical clustering. A hierarchical method finds a set of partitionings of a collection of objects into clusters that are ordered, so that all clusters are grouped into clusters of higher order. Hierarchical methods are used when one is interested in the entire structure of the data or in structures on several levels (of course, there must be theoretical grounds to assume a hierarchical structure in the data). In biological taxonomy for example, an object is commonly regarded as belonging successively to a species, a genus, a family and an order. Hierarchical methods also are applied in several areas of psychology and cognitive science. For example, hierarchical cluster analysis of data concerning similarity of body parts (see section 5.1), gives an idea of the cognitive representation of the body: smaller parts are grouped together into classes of bigger parts; toe, knee and thigh for example are grouped together into a class "leg".

Hierarchical methods are also useful when a partitioning method is preferred. When using a partitioning method, the number of clusters must be specified by the researcher. The number of clusters is an important issue, because it is usually unknown. A hierarchical method can serve as a guideline for choosing the number of clusters. Another problematical

issue when using a partitioning method is the choice of a starting allocation. A random allocation can be used, but also an allocation derived from a hierarchical analysis can serve as the starting point.

Because hierarchical methods form an important cluster technique, extending GROUPALS with a hierarchical method would be a worthwhile addition to the program. Implementing a hierarchical method in GROUPALS and providing the theoretical background is the purpose of this study. The central problem of the study can be divided into two components: (1) estimation of the object space and the restriction of this space to a hierarchical structure and (2) estimation of the hierarchical structure; these problems are the topics of chapter 3 and 4 respectively. In chapter 2 some definitions of a hierarchical structure and an overview of the basic hierarchical cluster methods are given. In chapter 5 two illustrative examples are provided. The conclusion of the study and topics for further research are given in chapter 6.

2 Hierarchical Clustering

Section 2.1 gives definitions of two concepts that are relevant in hierarchical classification: an N-tree and a binary tree. In section 2.2 the basic hierarchical clustering techniques, agglomerative and divisive, are described.

2.1 Trees

An N-tree on a set of objects $I \equiv \{1, 2, ..., N\}$ is a set S_W of subsets S_w, called clusters, of I, satisfying the conditions:

(1) $I \in S_W$

(2) $\{i\} \in S_W$, for all $i \in I$

(3) $\emptyset \notin S_W$

(4) if S_a,$S_b \in S_W$, then $S_a \cap S_b \in \{\emptyset, S_a, S_b\}$

Conditions (1) to (3) state that the cluster consisting of all objects, and all singletons (clusters consisting of one object, also called external nodes; non-singleton clusters are also called internal nodes) belong to the set of clusters and that there are no empty clusters. Condition (4) ensures that the clusters are hierarchically nested. An N-tree is called a binary tree if each internal node has two offspring subsets, that is, if for every non-singleton cluster S_w there exist two clusters $S_{w_1}, S_{w_2} \in S_W$ which are its proper subsets, such that $S_{w_1} \cup S_{w_2} = S_w$.

The clusters S_{w_1} and S_{w_2} are called the children of S_w, while S_w is called the parent (a parent will also be denoted by S_{pw}; children will also be denoted by S_{w_k}, where for "S_{w_k}" one should read "S_{w_k}, with k=1 or k=2"). The internal nodes are labeled with w=1,...,N-1 such that the root (which corresponds to I) is S_1. External nodes are denoted by S_{wi}, w=0 and $i \in I$ (S_{0i} denotes object i). For an illustration see Figure 1, in which the external nodes represent the objects and the internal nodes represent non-singleton subsets of objects. In addition to the complete set $I \equiv$\{1,...,6\} and the singleton subsets $\{i\}$ (i=1, ..., 6), the tree diagram comprises the subsets \{1,2\}, \{3,4\}, \{1,2,3,4\} and \{5,6\}. All internal nodes split into two proper subsets.

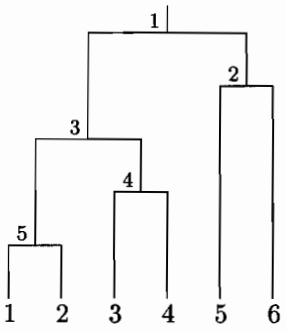


Figure 1: A binary tree

2.2 Overview of Hierarchical Clustering Methods

The process of hierarchical clustering is a step-by-step process of forming clusters from existing clusters. When the clustering process is top-down, the cluster method is called divisive, when the clustering process is bottom-up, the cluster method is called agglomerative. In the divisive method, at each stage a cluster is split into two clusters, starting with one cluster consisting of all the N objects, and repeating the steps until N clusters, all consisting of one member, are formed. In the agglomerative method, at each stage two clusters are fused, starting with N clusters all with one member, and repeating the steps until one cluster of N members is formed. So, for both methods, at each stage a new level is created, the total number of clusters on each level being one more (divisive method) or one less (agglomerative method) than the total number of clusters on the previous level. Both the agglomerative and the divisive methods result in a binary tree.

The levels of the tree are labeled such that the level number corresponds to the label of the cluster formed on that level. Each level K gives a partitioning of the N objects into K clusters, where K equals the number of clusters existing on a level that are not joined with another cluster on that level, that is, clusters that are fused with or split off from another cluster on a level higher in the hierarchy than K (such clusters will be called top nodes at level K). For example, in Figure 1 at level 3 the objects are partitioned into the three clusters S_3, S_{05} and S_{06}; S_3 is joined with another cluster at level 1, S_{05} and S_{06} are joined with another cluster at level 2.

At each stage, a criterion must be evaluated to decide upon the next step. Besides the difference in the process of building the hierarchy, hierarchical methods differ in the criterion they use. The most commonly used criteria are a distance criterion or a minimum variance criterion. Methods can be further distinguished by the way the distance or minimum variance is defined.

Table 1: Classification of hierarchical methods

	Successive optimization	Simultaneous optimization
Distance criterion	Single and complete linkage	Hartigan, Chandon et al. De Soete et al.
Minimum-variance criterion	Ward, Mirkin	Hierarchical GROUPALS

An important feature of classical hierarchical methods, such as single linkage and complete linkage (Gordon, 1996) and Ward's method (Ward, 1963), is the use of successive optimization, that is, each level is optimized given the existing levels, which does not necessarily optimize the total structure (see section 4.1). Successive optimization is also used in the method developed by Mirkin (1995). Methods for simultaneous optimization of a hierarchical structure have been developed in the least-squares framework; see, for example, Hartigan (1967), Chandon, Lemaire, and Pouget (1980), and De Soete, DeSarbo, and Carroll (1985).

This study concerns a method for simultaneous optimization of a hierarchical structure, using a minimum-variance criterion. An algorithm is developed that iteratively optimizes a hierarchical structure by performing local operations on a tree. The criterion used to evaluate the operations is the within-groups sum of squares per level summed over all levels except the root, called overall SS.

A classification of methods for hierarchical clustering by the optimization approach and the criterion they use is presented in Table 1.

3 Theory of Hierarchical GROUPALS

In this chapter the theory of GROUPALS in the context of hierarchical clustering is discussed. Section 3.1 describes the loss function. The form of the loss function in hierarchical GROUPALS is the same as in K-means GROUPALS, but for the restriction of the object scores different matrices are needed; these matrices are discussed in section 3.2. As in K-means GROUPALS, the object scores are restricted to cluster points. The cluster points in the case of hierarchical GROUPALS are treated in section 3.3. In section 3.4 the restriction of the node quantifications is discussed. Section 3.5 explains the hierarchical decomposition of the object scores and several cluster indices resulting from this decomposition. The final section 3.6 comments on the number of dimensions of the solution.

3.1 The Loss Function

GROUPALS uses the PRINCALS loss function with the scores of the objects on the first p components, called object scores, restricted to cluster points in order to cluster them. A cluster point is defined as the point that represents a cluster. Let $\mathbf{X}$ be the $N\times p$ matrix of object scores, where p denotes the dimensionality of the solution. Each observed variable is coded into a $N\times k_j$ indicator matrix $\mathbf{G}_j$, where k_j denotes the number of categories of variable j. $\mathbf{Y}_j$ is the $k_j\times p$ matrix of category quantifications for variable j. PRINCALS minimizes the loss function

$$\sigma(\mathbf{X};\mathbf{Y}_1,\ldots,\mathbf{Y}_m) = \frac{1}{m}\sum_j \|\mathbf{X}-\mathbf{G}_j\mathbf{Y}_j\|^2 \tag{1}$$

under the normalization conditions $\mathbf{u}'\mathbf{X}=\mathbf{0}$ (where $\mathbf{u}$ is a vector of ones) and $\mathbf{X}'\mathbf{X}=\mathbf{I}$. The matrices $\mathbf{Y}_j$ satisfy $\mathbf{u}'\mathbf{G}_j\mathbf{Y}_j = 0$. Mixed measurement levels are dealt with by restricting the class of data transformations, that is, by restricting $\mathbf{Y}_j$ (Gifi, 1990).

In GROUPALS, the object scores are restricted to cluster points by enforcing the cluster restriction $\mathbf{X}=\mathbf{EC}$. In the case of K-means GROUPALS each object is restricted to lie at only one of k cluster points. Here $\mathbf{E}$ is a $N\times k$ matrix, with k the number of clusters, for the allocation of the objects. Then, with $\mathbf{C}$ a $k\times p$ matrix of cluster points, $\mathbf{X}=\mathbf{EC}$ restricts the score of each object to the cluster point of the cluster the object belongs to. In hierarchical GROUPALS we have nested allocations of the objects on each of the N-1 levels of the hierarchy. Here $\mathbf{E}$ is a $N\times(N\text{-}1)$ so-called nested incidence matrix and $\mathbf{C}$ a $(N\text{-}1)\times(N\text{-}1)$ matrix of (internal) node quantifications. The matrices $\mathbf{E}$ and $\mathbf{C}$ are discussed in more detail in the next section.

Inserting the cluster restriction in (1) gives the GROUPALS loss function:

$$\sigma(\mathbf{E};\mathbf{C};\mathbf{Y}_1,\ldots,\mathbf{Y}_m) = \frac{1}{m}\sum_j \|\mathbf{EC}-\mathbf{G}_j\mathbf{Y}_j\|^2. \tag{2}$$

Loss function (2) is minimized by using an alternating least squares algorithm consisting of three major steps. In the first step, keeping $\mathbf{E}$ and $\mathbf{C}$ fixed, updates for the category quantification matrices $\mathbf{Y}_j$ are found using the procedures described in Gifi (1990). In the second step, keeping the $\mathbf{Y}_j$'s fixed and using a normalization procedure that is the same as in K-means GROUPALS (the transfer of normalization procedure, described in Van Buuren (1986) and Van Buuren and Heiser (1989)), we update the unrestricted object scores $\mathbf{Z}$ as $\mathbf{Z} = \frac{1}{m}\Sigma_j \mathbf{G}_j\mathbf{Y}_j$ with $\mathbf{u}'\mathbf{Z} = \mathbf{0}$ and update $\mathbf{E}$ by estimating a locally optimal tree for $\mathbf{Z}$. In the third step, keeping the $\mathbf{Y}_j$'s and $\mathbf{E}$ fixed, the matrix of node quantifications $\mathbf{C}$ is updated and the restricted object scores are estimated as $\mathbf{EC}$. A flow chart of the algorithm is given in appendix A.

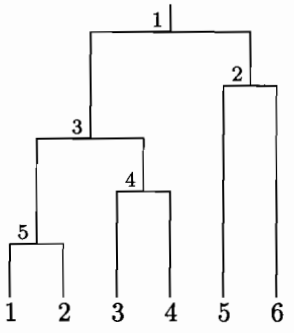
3.2 Restriction of the Object Scores

Mirkin (1995) shows that when working with an object by variable matrix (the variables in our case being the components), a hierarchical decomposition of this matrix can be applied because a binary hierarchy can be coded into a nested incidence matrix $\mathbf{E}$, that forms an orthonormal (N-1)-dimensional basis of the variable space (the component space in our case). In section 3.5 this will be discussed in more detail. Following Mirkin, we define $\mathbf{E}$ as:

$$\mathbf{E} = \begin{cases} \sqrt{\frac{n_{w_2}}{n_{w_1} n_w}} & \text{if } i \in S_{w_1} \\ \sqrt{\frac{n_{w_1}}{n_{w_2} n_w}} & \text{if } i \in S_{w_2} \\ 0 & \text{if } i \notin S_w \end{cases} \tag{3}$$

with n_{w_1} and n_{w_2} the number of objects in S_{w_1} and S_{w_2}, and $n_w = n_{w_1} + n_{w_2}$. It can be easily verified that $\mathbf{E}$ has the properties $\mathbf{u}'\mathbf{E} = \mathbf{0}$ and $\mathbf{E}'\mathbf{E} = \mathbf{I}$. Moreover, $\mathbf{E}$ has the property $\mathbf{EE}' = \mathbf{J}$, with $\mathbf{J}$ the centering operator $\mathbf{I} - \frac{\mathbf{uu}'}{\mathbf{u}'\mathbf{u}}$. Thus, in each column $\mathbf{e}_w$ of $\mathbf{E}$, the objects of S_w are splitted into the two children of S_w. The values of the elements of $\mathbf{E}$ can be determined when the tree is known. See Figure 2 for an example of coding a tree into $\mathbf{E}$.

To minimize loss function (2) over the node quantifications $\mathbf{C}$ for fixed $\mathbf{Y}_j$'s and $\mathbf{E}$ we split (2) into two additive components as follows: by inserting the identity $\mathbf{EC} = \mathbf{Z}-(\mathbf{Z}-\mathbf{EC})$



$$\mathbf{E} = \begin{bmatrix} \sqrt{\frac{2}{24}} & 0 & \sqrt{\frac{2}{8}} & 0 & \sqrt{\frac{1}{2}} \\ \sqrt{\frac{2}{24}} & 0 & \sqrt{\frac{2}{8}} & 0 & -\sqrt{\frac{1}{2}} \\ \sqrt{\frac{2}{24}} & 0 & -\sqrt{\frac{2}{8}} & \sqrt{\frac{1}{2}} & 0 \\ \sqrt{\frac{2}{24}} & 0 & -\sqrt{\frac{2}{8}} & -\sqrt{\frac{1}{2}} & 0 \\ -\sqrt{\frac{4}{12}} & \sqrt{\frac{1}{2}} & 0 & 0 & 0 \\ -\sqrt{\frac{4}{12}} & -\sqrt{\frac{1}{2}} & 0 & 0 & 0 \end{bmatrix}$$

Figure 2: coding of a tree into a nested incidence matrix $\mathbf{E}$

into (2), we obtain:

$$\begin{aligned} \sigma(\mathbf{E};\mathbf{C};\mathbf{Y}_1,\ldots,\mathbf{Y}_m) &= \frac{1}{m}\textstyle\sum_j \|(\mathbf{Z}-\mathbf{G}_j\mathbf{Y}_j)-(\mathbf{Z}-\mathbf{EC})\|^2 \\ &= \frac{1}{m}\textstyle\sum_j \|\mathbf{Z}-\mathbf{G}_j\mathbf{Y}_j\|^2 + \frac{1}{m}\sum_j \|\mathbf{Z}-\mathbf{EC}\|^2 - \frac{2}{m}\sum_j \text{tr}\,[(\mathbf{Z}-\mathbf{G}_j\mathbf{Y}_j)'(\mathbf{Z}-\mathbf{EC})] \\ &= \frac{1}{m}\textstyle\sum_j \|\mathbf{Z}-\mathbf{G}_j\mathbf{Y}_j\|^2 + \|\mathbf{Z}-\mathbf{EC}\|^2 - 2\text{tr}\,[(\mathbf{Z}-\frac{1}{m}\sum_j \mathbf{G}_j\mathbf{Y}_j)'(\mathbf{Z}-\mathbf{EC})] \\ &= \frac{1}{m}\textstyle\sum_j \|\mathbf{Z}-\mathbf{G}_j\mathbf{Y}_j\|^2 + \|\mathbf{Z}-\mathbf{EC}\|^2, \end{aligned} \tag{4}$$

where the cross product vanishes due to the definition of $\mathbf{Z}$. In minimizing (4) for fixed $\mathbf{Y}_j$'s, the first term is constant, so only the last term of (4) has to be minimized over $\mathbf{E}$ and $\mathbf{C}$. It is well-known that $\widehat{\mathbf{C}} = (\mathbf{E}'\mathbf{E})^{-1}\mathbf{E}'\mathbf{Z}$ minimizes this term. Because $\mathbf{E}'\mathbf{E} = \mathbf{I}$, the estimation of $\mathbf{C}$ simplifies to $\widehat{\mathbf{C}} = \mathbf{E}'\mathbf{Z}$. For level w and dimension t the values of $\widehat{\mathbf{C}}$ can be expressed as

$$\begin{aligned} \hat{c}_{wt} = \mathbf{e}_w'\mathbf{z}_t &= \sqrt{\frac{n_{w_2}}{n_{w_1}n_w}}\sum_{i\in S_{w_1}} z_{it} - \sqrt{\frac{n_{w_1}}{n_{w_2}n_w}}\sum_{i\in S_{w_2}} z_{it} \\ &= \sqrt{\frac{n_{w_1}^2 n_{w_2}}{n_{w_1}n_w}}M_{w_1t} - \sqrt{\frac{n_{w_2}^2 n_{w_1}}{n_{w_2}n_w}}M_{w_2t} \\ &= \sqrt{\frac{n_{w_1}n_{w_2}}{n_w}}(M_{w_1t} - M_{w_2t}), \end{aligned} \tag{5}$$

where M_{w_1t} and M_{w_2t} are the means of the objects scores on dimension t for the objects in S_{w_1} and S_{w_2}. Thus, the node quantifications are estimated as the weighted difference between the means of the children of a node.

Since $\mathbf{Z}$ is centered on the origin, the cluster restriction $\mathbf{E}\widehat{\mathbf{C}}$ results in $\mathbf{E}\mathbf{E}'\mathbf{Z} = \mathbf{J}\mathbf{Z} = \mathbf{Z}$, that is, $\mathbf{E}\widehat{\mathbf{C}}$ gives the unrestricted object scores. So, to find object scores restricted to cluster points we have to impose a restriction on the node quantifications $\widehat{\mathbf{C}}$. This restriction is

discussed in section 3.4. First we have to know how to find the cluster points, which is the topic of the next section.

3.3 Cluster Points

In section 3.1 a cluster point was defined as the point that represents the cluster. Since an external node consists of one object, the cluster point on dimension t for an external node S_{0i} is the score of object i on dimension t, z_{it}. Writing out the elements of $\mathbf{Z}$ we obtain

$$z_{it} = \mathbf{e}_i'\hat{\mathbf{c}}_t = \sum_{w=1}^{N-1} e_{iw}\hat{c}_{wt} \qquad i \in S_w. \tag{6}$$

Using expression (5) and the fact that $n_w = n_{w_1} + n_{w_2}$, the terms in the sum in expression (6) can be written as

$$\begin{aligned} e_{iw}\hat{c}_{wt} &= \sqrt{\frac{n_{w_2}}{n_{w_1} n_w}}\sqrt{\frac{n_{w_1} n_{w_2}}{n_w}}(M_{w_1 t} - M_{w_2 t}) \\ &= \frac{n_{w_2}}{n_w}(M_{w_1 t} - M_{w_2 t}) = -(M_{wt} - M_{w_1 t}) \qquad \text{if } i \in S_{w_1} \\ \text{and } e_{iw}\hat{c}_{wt} &= -\sqrt{\frac{n_{w_1}}{n_{w_2} n_w}}\sqrt{\frac{n_{w_1} n_{w_2}}{n_w}}(M_{w_1 t} - M_{w_2 t}) \\ &= -\frac{n_{w_1}}{n_w}(M_{w_1 t} - M_{w_2 t}) = -(M_{wt} - M_{w_2 t}) \quad \text{if } i \in S_{w_2}. \end{aligned} \tag{7}$$

For external clusters the cluster points are given by expression (6). This expression can be generalized by noting that the sum in (6) is taken over the nodes on the path from the root to the parent of the cluster because e_{iw} is zero for $i \notin S_w$. Thus, in general, using expression (7), the cluster point for S_l on dimension t, R_{lt}, is

$$R_{lt} = \sum_{w=1}^{pl} e_{iw}\hat{c}_{wt} = \sum_{w=1}^{pl}(M_{w_k t} - M_{wt}) \qquad i \in S_w, \tag{8}$$

where pl denotes the level of the parent of S_l. Because the sum in expression (8) is taken over the clusters on the path from the root to the parent of S_l, for each successive pair of terms in expression (8) the child in the first term is the parent in the next term. Using this and noting that $M_{1t} = 0$, expression (8) can be written as

$$R_{lt} = \sum_{w=1}^{pl}(M_{w_k t} - M_{wt}) = \sum_{w=2}^{l} M_{wt} - \sum_{w=1}^{pl} M_{wt} = M_{lt}, \tag{9}$$

with w on the path from S_1 to S_l. For example, see Figure 1 (page 4): cluster 3 is a child of cluster 1 and cluster 5 is a child of cluster 3: $S_{1_k} = S_3$ and $S_{3_k} = S_5$. The cluster point on

dimension t of cluster 5 is $R_{5t} = M_{1_kt} - M_{1t} + M_{3_kt} - M_{3t} = M_{3t} - M_{1t} + M_{5t} - M_{3t} = M_{5t}$. So, the cluster points are the centroids of the clusters in terms of $\mathbf{Z}$.

3.4 Restriction of the Node Quantifications

As was concluded in section 3.2, we need to impose a restriction on the matrix of node quantifications $\widehat{\mathbf{C}}$ to restrict the object scores $\mathbf{Z}$ to cluster points. To see what this restriction should be, we need to consider the restriction of the object scores to the hierarchical structure. On the first dimension the objects must be restricted to lie in one of two cluster points, on the second dimension they must be restricted to lie in one of three cluster points, etc. Thus, in general, the object scores on dimension t must be restricted to $t+1$ cluster points. As was mentioned in chapter 2, on level K of the tree the objects are partitioned into K clusters. So, on dimension t the object scores must be restricted to the cluster points on dimension t of the clusters belonging to the partition at level $t+1$ (top nodes at level $t+1$). Recall from chapter 2 that the top nodes at level $t+1$ have parents that are higher (lower labels) in the hierarchy than $t+1$, which means that in expression (9) $pl < t+1$. Because the lowest (highest label) of the parents of the top nodes at level $t+1$ always is one level higher than the highest top node (lowest label) at level $t+1$, $pl < t+1$ is equal to $pl \leq t$. For example, on dimension 2 the object scores have to be restricted to the cluster points of the top nodes at level 3; in Figure 1 (page 4) these nodes would be S_3, S_{05} and S_{06}, with parents S_1 and S_2 respectively.

So, the cluster points on dimension t are given by $\sum_{w=1}^{pl} e_{iw}\hat{c}_{wt}$, with $i \in S_l$, S_l top node at level $t+1$ and $pl \leq t$. By setting $\hat{c}_{wt}$ to 0 for $w > t$, the term $\mathbf{e}_i'\hat{\mathbf{c}}_t$, with $i \in S_w$, gives the cluster point of S_w on dimension t. Then for $\underline{\mathbf{C}} = \widehat{\mathbf{C}}$ with $\underline{c}_{wt} = 0$ if $w > t$, $\mathbf{E}\underline{\mathbf{C}} = \underline{\mathbf{Z}}$ gives the restricted (unnormalized) object scores. That this choice of $\mathbf{C}$ minimizes the last term of (4) can be seen by decomposing this term into

$$\|\mathbf{Z} - \mathbf{E}\mathbf{C}\|^2 = \|\mathbf{Z} - \mathbf{E}\widehat{\mathbf{C}}\|^2 + \|\mathbf{C} - \widehat{\mathbf{C}}\|^2. \tag{10}$$

Expression (10) follows from inserting the identity $\mathbf{C} = \widehat{\mathbf{C}} + (\mathbf{C} - \widehat{\mathbf{C}})$, giving

$$\begin{aligned}
\|\mathbf{Z} - \mathbf{E}\mathbf{C}\|^2 &= \|(\mathbf{Z} - \mathbf{E}\widehat{\mathbf{C}}) - \mathbf{E}(\mathbf{C} - \widehat{\mathbf{C}})\|^2 \\
&= \|\mathbf{Z} - \mathbf{E}\widehat{\mathbf{C}}\|^2 + \|\mathbf{E}(\mathbf{C} - \widehat{\mathbf{C}})\|^2 - 2\text{tr}\,[(\mathbf{Z} - \mathbf{E}\widehat{\mathbf{C}})'\mathbf{E}(\mathbf{C} - \widehat{\mathbf{C}})] \\
&= \|\mathbf{Z} - \mathbf{E}\widehat{\mathbf{C}}\|^2 + \|\mathbf{C} - \widehat{\mathbf{C}}\|^2 - 2\text{tr}\,[(\mathbf{Z}'\mathbf{E} - \widehat{\mathbf{C}}')(\mathbf{C} - \widehat{\mathbf{C}})] \\
&= \|\mathbf{Z} - \mathbf{E}\widehat{\mathbf{C}}\|^2 + \|\mathbf{C} - \widehat{\mathbf{C}}\|^2,
\end{aligned} \tag{11}$$

where the cross product vanishes because $\mathbf{Z}'\mathbf{E} = \widehat{\mathbf{C}}'$. When minimizing (10) over $\mathbf{C}$, the first term is constant, so we have to minimize the last term. This minimization can be done columnwise as follows:

$$\begin{aligned} \|\mathbf{C} - \widehat{\mathbf{C}}\|^2 &= \sum_t \|\mathbf{c}_t - \hat{\mathbf{c}}_t\|^2 \\ &= \sum_t \sum_{w=1}^{t} (c_{wt} - \hat{c}_{wt})^2 + \sum_t \sum_{w=t+1}^{N-1} \hat{c}_{wt}^2, \end{aligned} \tag{12}$$

where the c_{wt} are the restricted node quantifications, with $c_{wt} = 0$ if $w > t$. Clearly, (12) is minimized by choosing $\underline{c}_{wt} = \hat{c}_{wt}$ if $w \leq t$ and $\underline{c}_{wt} = 0$ if $w > t$. So, the matrix of restricted node quantifications $\underline{\mathbf{C}}$ is equal to the upper triangular part of $\widehat{\mathbf{C}}$.

3.5 Hierarchical Decomposition of the Object Scores

Because $\mathbf{E}$ forms an orthonormal (N-1)-dimensional basis of $\mathbf{Z}$, a hierarchical decomposition of $\mathbf{Z}$ can be applied (Mirkin, 1995), as was mentioned in section 3.2. This decomposition has the form $\mathbf{Z} = \mathbf{EMN}$, with $\mathbf{M}$ defined as a (N-1)×(N-1) diagonal matrix with diagonal entries μ_w, and $\mathbf{N}$ as the normed matrix $\mathbf{N} = \mathbf{M}^{-1}\widehat{\mathbf{C}}$, where μ_w, called the cluster value, is the Euclidean norm of the p-dimensional vector of node quantifications $\hat{\mathbf{c}}_w$. Using equation (5) the cluster value can be expressed as

$$\mu_w = \sqrt{\frac{n_{w_1} n_{w_2}}{n_w}} \sqrt{\textstyle\sum_t (M_{w_1 t} - M_{w_2 t})^2} = \sqrt{\frac{n_{w_1} n_{w_2}}{n_w}}\, d(S_{w_1}, S_{w_2}), \tag{13}$$

where $d(S_{w_1}, S_{w_2})$ is the Euclidean distance between the centroids of S_{w_1} and S_{w_2}. The term μ_w is positive if $S_{w_1} \neq S_{w_2}$ and zero if $S_{w_1} = S_{w_2}$. Thus, the decomposition $\mathbf{Z} = \mathbf{EMN}$ is analogous to the singular value decomposition (SVD) of $\mathbf{Z}$. The decomposition is not identical to a SVD since the vectors $\mathbf{n}_w$ are, in general, not mutually orthogonal.

The hierarchical decomposition of the object scores of the form $\mathbf{Z} = \mathbf{E}\widehat{\mathbf{C}}$ provides a set of cluster indices in terms of $\mathbf{Z}$, since the variances and covariances and the elements of $\mathbf{Z}$ can be decomposed by the clusters. Because $\mathbf{E}'\mathbf{E} = \mathbf{I}$ it is true that $\mathbf{Z}'\mathbf{Z} = \widehat{\mathbf{C}}'\widehat{\mathbf{C}}$. Since $\mathbf{Z}$ is centered, the elements of $\mathbf{Z}'\mathbf{Z}$ are the variances and covariances of $\mathbf{Z}$ (up to a factor N). Using equation (5), the elements of $\mathbf{Z}'\mathbf{Z}$ can be written as

$$(\mathbf{Z}'\mathbf{Z})_{st} = \mathbf{z}_s'\mathbf{z}_t = \textstyle\sum_w \frac{n_{w_1} n_{w_2}}{n_w} (M_{w_1 s} - M_{w_2 s})(M_{w_1 t} - M_{w_2 t}). \tag{14}$$

This equality can be interpreted as a decomposition of the variances (if $s = t$) and covariances of $\mathbf{Z}$ by the clusters of the hierarchy S_W. The total variance of $\mathbf{Z}$ is equal to the sum of the

squared cluster values (up to a factor N), as can be seen by summing (14) with $s = t$ over t=1,...,N-1, which yields the square of expression (13) summed over w=1,...,N-1.

The elements of $\mathbf{Z}$ can be decomposed into a sum of differences between the mean of a cluster and the mean of its child from which object i is a member. This difference indicates how well clusters fit into their parent cluster for clusters S_w, $i \in S_w$. Also, comparing this difference for one child of S_w with the difference for the other child gives an indication how symmetric the split of S_w is. The difference between the means of the children of S_w is an indication of the internal cohesiveness of S_w.

The elements of $\widehat{\mathbf{C}}$ can be interpreted (if the columns of $\mathbf{Z}$ would be standardized) as cluster loadings as can be seen by squaring expression (5). This gives the criterion used in Ward's method of hierarchical clustering (Ward, 1963): the increase in within-groups SS on dimension t when S_{w_1} and S_{w_2} are joined to form S_w. Thus $\hat{c}^2_{wt}$ indicates the contribution of S_w to the variance of $\mathbf{z}_t$.

3.6 Number of Dimensions

In the theory as described, the number of dimensions is $N-1$. For a solution in p dimensions with p less than N-1, $\mathbf{Z}$ is a $N \times p$ matrix, and $\mathbf{C} = \mathbf{E}'\mathbf{Z}$ is a $(N\text{-}1) \times p$ matrix, with the elements below (t,t), $t = 1,\ldots,p$, set to zero. When $p < N$-1, the object scores are restricted on only the first p levels of the tree. This does not make the lower levels of the tree redundant, because the higher levels are based upon the lower levels. When $p < N-1$, the change in overall SS for a local tree operation is computed over all levels and, if the operation lowers the overall SS, the change in overall SS for the first p levels is computed. Operations that lower the overall SS are executed only if the overall SS over the first p levels does not increase.

4 Estimating the Tree

For the restriction of the object scores to $\mathbf{X} = \mathbf{EC}$ we need to estimate $\mathbf{E}$ and $\mathbf{C}$. In chapter 3 the structure of $\mathbf{E}$ and the estimation of $\mathbf{C}$ from $\mathbf{E}$ and $\mathbf{Z}$ was discussed. This chapter deals with the problem of estimating the values of $\mathbf{E}$ to minimize the overall SS, which means that the optimal tree has to be found.

Some concepts in this chapter and the appendix are explained by reference to a tree. Until now we labeled the root with 1. To avoid confusion when talking about nodes higher or lower in the tree, from here on the order of the labels of the internal nodes is reversed: S_1 is the lowest internal node, S_{N-1} is the root.

Finding the optimal tree is a discrete optimization problem, which has no global solution unless N is very small. However, a locally optimal tree can be found in the sense that the tree cannot be improved by any of a set of local operations on the tree, operations that change the tree structure slightly (Hartigan, 1967). To find a locally optimal tree, we start with an initial tree using Ward's method (Ward, 1963). This method optimizes a tree successively. That this not necessarily optimizes the total tree can be seen in Figure 3: tree A is the Ward tree with overall SS equal to 4.114 (within-groups SS per level, summed over levels 1 to 3: .50 on level 1, 1.167 on level 2, 1.167+1.28= 2.447 on level 3). In tree B the overall SS is 4.045 (.50 on level 1, .50+.725=1.225 on level 2, .50+1.82=2.32 on level 3). Tree B can not be obtained by Ward's method because the increase in SS when joining objects 3 and 4 (.725) is higher than the increase in SS when adding object 3 to cluster 1 (.667). It is possible to obtain tree B from tree A by removing the second level of A, that is, by removing the edge or branch between node 1 and object 3, and adding a branch from object 3 to object 4.

To ensure that the total tree is locally optimal, we use a set of two local tree operations:

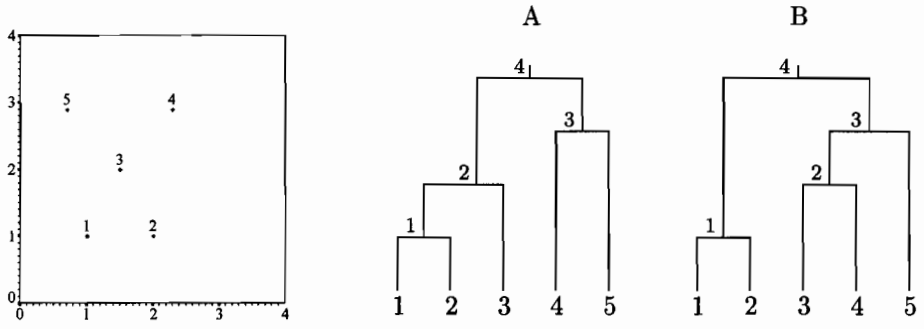


Figure 3: Ward tree (A) and optimal tree (B) for the objects in the plot

reordering of levels and rebranching, defined in section 4.1. In section 4.2 the algorithm to evaluate these operations is described.

4.1 Local Tree Operations

In this section the reordering and the rebranching operations are discussed. The notation used in this discussion is:

S_{w_k} $k = 1, 2$ the children of S_w

S_{pw} the parent of S_w

S_w $w = a$ the node operated on

$w = c$ the node S_{a_k} is attached to

e the level on which S_{a_k} is attached to S_c

b the level S_a is reordered to

SN the new node formed by rebranching S_{a_k} to S_c

τ number of times a node is top node

ι increase in overall SS when executing a operation

δ decrease in overall SS when executing a operation

Δ total change in overall SS

1. Reordering

   The level of S_a changes to b, with $0 < a < N$-1, $\max(a_1, a_2) < b < S_{pa}$,and $b \neq a$. Thus, the level of S_a can change to levels between its parent and the higher of its children.

   For example see Figure 4, tree A: if the cluster operated on is cluster 4 ($a = 4$), the cluster can be reordered to levels between 5 (parent) and 0 (child), so $b = 1$, 2 or 3. When the cluster is reordered to level 3, the level of cluster 4 becomes 3, which has the consequence that the level of cluster 3 becomes 4. This reordering operation results in tree C.

2. Rebranching (adjusted from Hartigan, 1967)

   S_a is removed from the tree and S_{a_k} is attached to S_c, with $c \neq a, a_k$ or a_{3-k}, on level $e = c$. Attaching S_{a_k} to S_c on level e means that SN is inserted in the tree between S_e and S_{e+1}. When $S_c < S_{a_k}$, e is equal to a_k (in the sequel if c refers to the attachment level, for "c", read "c or a_k"). For example see Figure 4, tree A with $a = 2$, $a_k = 0$,

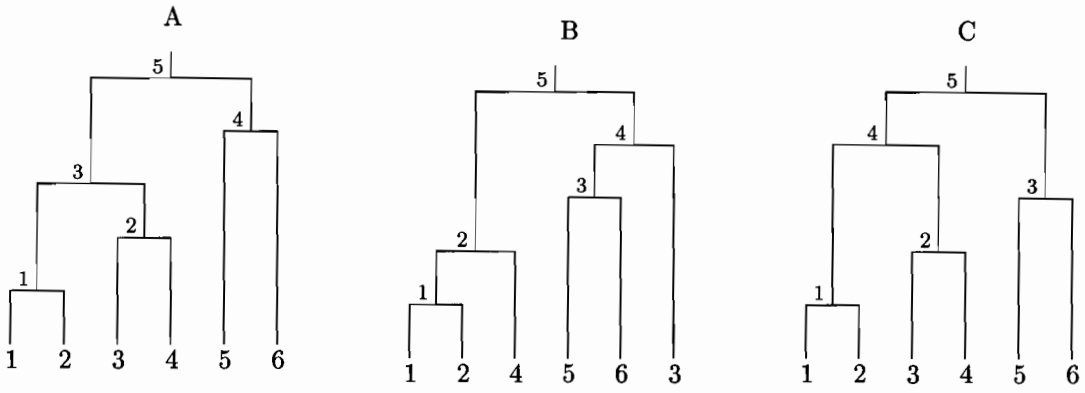


Figure 4: Illustration of the rebranching operation (resulting in tree B) and the reordering operation (resulting in tree C) on tree A

$i = 3$, and $c = e = 4$: level 2 is removed and object 3 is attached to cluster 4 between levels 4 and 5, resulting in tree B. Removing level 2 lowers the levels of clusters higher than 2 with one and inserting a level between 4 and 5 raises the levels higher than 4 with one. Thus, level numbers under 2 and above 4 do not change and the levels between 2 and 5 are lowered with one: cluster 3 in A is cluster 2 in B and cluster 4 in A is cluster 3 in B. Then the level of the new node formed by adding object 3 to cluster 3 in B (which was 4 in A) is 4 (note that e refers to level numbers in the tree operated on, not to level numbers in the new tree).

If $\tau_c > 1$ the rebranching operation is followed by the reordering operation on SN for $b = e+1,\ldots,pc-1$, that is, the attachment of S_{a_k} to S_c can take place on levels $c \leq e < pc$ with $e \neq a$. In the rebranching example given above $\tau_4 = 1$, so attachment can only take place on level 4. In tree C $\tau_3 = 2$, then if object 1 is rebranched to cluster 3, for example, the attachment can take place on levels 3 and 4, resulting in trees D and E respectively in Figure 5.

There is a restriction on rebranching to c when $a_k \neq 0$ and $c < a_k$: then S_{a_k} can only be rebranched to S_c if S_c is top node on level a_k, because if c is not top node on level a_k, a more complicated reordering is required (consider for example tree B in Figure 4 with $a = 5$, $a_k = 4$, and $c = 1$).

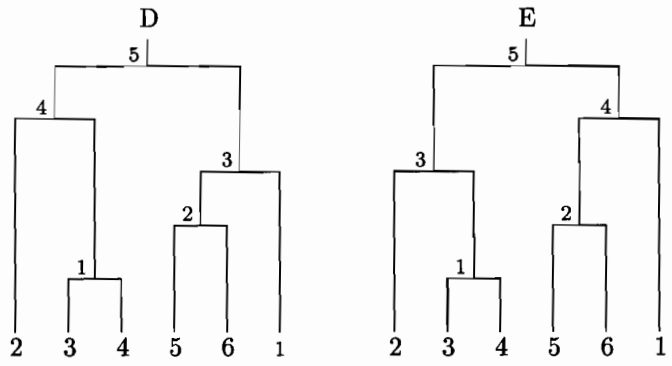


Figure 5: Illustration of rebranching object 1 to cluster 3 in tree C from Figure 4, resulting in tree D if $e = 3$ and in tree E if $e = 4$

4.2 Algorithm for Local Tree Operations

The algorithm starts with $a = 1$. First the reordering operation on S_a is evaluated for the values of b, next the rebranching operation on S_{a_1} for the values of c and e is evaluated, then the rebranching operation on S_{a_2} is evaluated for the values of c and e (in evaluating the rebranching operation, some values are the same for both the children of a node; working on the children of a node successively avoids the need for an array to store these values).

An operation that lowers the overall SS can be executed immediately (next first method) after which the algorithm continues with S_{a+1}, or the operation parameters can be stored and after a pass through all nodes the operation that lowers the overall SS most is executed, after which the algorithm starts again with S_1 (best first method). The algorithm stops when during a pass through all nodes no operation is found that lowers the overall SS. In the best first method a pass through all nodes is a pass from S_1 through S_{N-1}. In the next first method a pass trough all nodes is a pass from S_w through S_{w-1} (w is initialized on 1 and set to $a + 1$ if an operation is executed; $w - 1$ is initialized on $N - 1$).

The overall SS can be computed as:

$$\begin{aligned} \text{overall SS} &= \textstyle\sum_l \left(\sum_w \mathrm{SS}(S_w)\right) \text{ for } S_w \text{ is top node on level } l \\ &= \textstyle\sum_w \tau_w \mathrm{SS}(S_w) \end{aligned} \tag{15}$$

In the rebranching operation both $\mathrm{SS}(S_w)$ and τ_w can change. The total change Δ for the rebranching operation is divided into 4 components:

1. Removing S_a from the tree also means removing level a: the overall SS decreases with $\mathrm{SS}(S_a)$ and with the sum of the SS of the top nodes on level a.

2. By attaching S_{a_k} to S_c a new level is inserted in the tree between c and pc: the overall SS increases with the sum of the SS of the top nodes on level c and with the SS of the new node SN.

3. When S_{a_k} is rebranched to S_c, S_{a_k} has to be removed from S_w, for w on the path from S_a to the root. Removing S_{a_k} from S_w, with $w \neq a$, decreases the overall SS with δ_w on levels w to pw: removing S_{a_k} from S_w decreases the overall SS with $\tau_w \delta_w$; for the levels between a and $a + \tau_a$ the decrease is $(\tau_a - 1)\delta_a$.

4. When S_{a_k} is rebranched to S_c, S_{a_k} has to be added to S_w, for w on the path from S_c to the root. Adding S_{a_k} to S_w, with $w \neq c$, increases the overall SS with ι_w on levels w to pw: adding S_{a_k} to S_w increases the overall SS with $\tau_w \iota_w$; for the levels between e and $c + \tau_c$ the increase is $((c + \tau_c) - (e + 1))\iota_c$.

So, Δ for the rebranching operation is computed as:

$$\begin{array}{rcll} \Delta & = & -\,\mathrm{SS}(S_a) - \sum_w \mathrm{SS}(S_w) & \text{for } S_w \text{ is top node at level } a \\ & & +\,\mathrm{SS}(SN) + \sum_w \mathrm{SS}(S_w) & \text{for } S_w \text{ is top node at level } e, \text{ with} \\ & & & w \neq c \text{ or } a_k \\ & & -\,(\tau_a - 1)\delta_a - \sum_w \tau_w \delta_w & \text{for } S_w \text{ on path from } S_a \text{ to root} \\ & & +\,((c + \tau_c) - (e + 1))\iota_c + \sum_w \tau_w \iota_w & \text{for } S_w \text{ on path from } S_c \text{ to root.} \end{array} \tag{16}$$

If in the last component of (16) $e < a < c + \tau$, then τ is set to $\tau - 1$.

In the reordering operation only τ_w can change. The term τ_w can be computed as $pw - w$. When a changes to b, τ_a changes from $pa - a$ to $pa - b$, and τ_{a_k} changes from $a - a_k$ to $b - a_k$. Changing a to b can be seen as removing S_a from the tree and inserting S_a in the tree at level b. Then τ_w decreases with 1 for nodes that are top node at level a and τ_w increases with 1 for nodes that are top node at level b.
So, Δ for the reordering operation on S_a is computed as:

$$\begin{array}{rcll} \Delta & = & (a - b)\mathrm{SS}(S_a) & \\ & & +\,(b - a)[\mathrm{SS}(S_{a_1}) + \mathrm{SS}(S_{a_2})] & \\ & & -\,\sum_w \mathrm{SS}(S_w) & \text{for } S_w \text{ is top node at level } a, \text{ with} \\ & & & w \neq a, a_1 \text{ or } a_2 \\ & & +\,\sum_w \mathrm{SS}(S_w) & \text{for } S_w \text{ is top node at level } b, \text{ with} \\ & & & w \neq a, a_1 \text{ or } a_2. \end{array} \tag{17}$$

The terms δ_w and ι_w are computed using Ward's criterion as the squared Euclidean distance between the means of S_{a_k} and S_w, multiplied by a factor weighting n_{a_k} and n_w:

$$\begin{aligned} \delta_w &= \frac{n_w n_{a_k}}{n_w - n_{a_k}} d^2(M_w, M_{a_k}) \\ \iota_w &= \frac{n_w n_{a_k}}{n_w + n_{a_k}} d^2(M_w, M_{a_k}). \end{aligned} \tag{18}$$

A more detailed description of the algorithm is given in appendix B.

5 Examples

To be able to compare the Hierarchical GROUPALS method to another hierarchical method, we have also implemented a divisive method in the clustering component of GROUPALS. The divisive algorithm performs successive K-means clustering with K=2. To find the best partition on each level, the algorithm computes a specified number of partitions, each starting from a different random allocation. The order of the splits is determined by the cluster values. In Mirkin (1995) this divisive method is used for illustration purposes.

5.1 Body Data

The data are the same data Mirkin (1995) uses for illustration: a subset of the dissimilarity data concerning body parts as collected by Miller (1968) and reported in Rosenberg (1982) (see Table 2). These data were analyzed in 4 dimensions and treating the variables as single numerical. In Figure 6 the divisive solution is presented on the left side and the Hierarchical GROUPALS solution on the right side (where the Hierarchical GROUPALS method refers to the optimization method as described in chapter 3). Because the squared cluster values μ^2 sum to the total variance of $\mathbf{Z}$ (up to a factor N), they are given as percentages of the total variance. The cluster values, multiplied by a factor such that the highest value is 100, are used as the heights of the clusters.

Table 2: Extract from Miller's sorting data (1968): number of subjects out of 50 who did not put any given row terms into the column categories

No.	Term	Head	Arm	Chest	Leg
1	Body	45	50	37	50
2	Cheek	19	50	49	50
3	Ear	18	49	50	49
4	Elbow	49	8	50	47
5	Face	14	48	47	48
6	Hand	48	14	50	46
7	Knee	49	47	50	8
8	Lip	18	49	50	49
9	Lung	48	49	17	49
10	Mouth	19	49	50	49
11	Neck	31	45	38	45
12	Palm	50	16	49	48
13	Thigh	47	45	48	5
14	Toe	49	47	50	13
15	Trunk	42	46	19	45
16	Waist	44	45	26	46

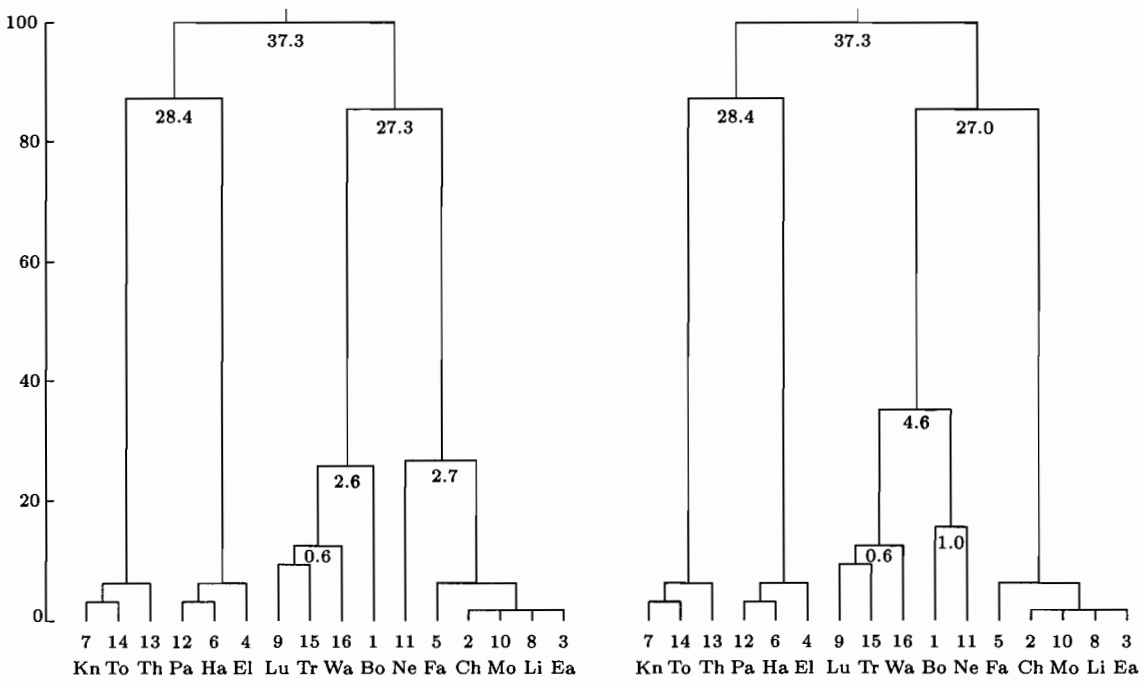


Figure 6: Tree and squared cluster values as percentages of the total variance resulting from the divisive method (left), and from the Hierarchical GROUPALS method (right). The heights of the vertical edges represent the cluster values

The squared node quantifications $\hat{c}^2_{wt}$, or cluster "loadings", for the most contributing clusters and the component loadings and fit for the Hierarchical GROUPALS solution are given in Table 3. The uniformly low values of the component loadings on the fourth dimension indicate that most of the variance is already accounted for by the first 3 dimensions. The lower levels of the tree only account for local effects. The squared cluster "loadings" can be used to identify the variables that contribute to the split of a cluster. For example, the split of cluster 14 is due to dimension 3, on which dimension only the variables arm and leg have high loadings. Figure 7 presents a plot of the first 2 dimensions of the object space. The object scores are given in Table 4 (the object scores GROUPALS uses for output are the unrestricted, unnormalized scores $\mathbf{Z}$). Figure 7 clearly shows the split of the objects into the head-chest cluster and the arm-leg cluster on the first dimension, and the split of the

Table 3: Squared cluster "loadings", component loadings and fit for the Hierarchical GROUPALS solution

			Dimension				
			1	2	3	4	
Squared cluster							$\Sigma = \mu^2$
"loadings"	cluster:	15	21.30	2.57	0.02	0.00	23.89
		14	0.04	0.00	18.16	0.00	18.20
		13	3.55	13.66	0.01	0.03	17.25
		12	0.07	2.68	0.00	0.19	2.94
		11	0.07	0.16	0.00	0.41	0.64
Component loadings	variable:	head	-0.89	-0.40	0.02	-0.07	
		arm	0.65	-0.18	0.72	-0.06	
		chest	-0.13	0.98	-0.01	-0.05	
		leg	0.58	-0.19	-0.78	-0.05	
Fit			0.39	0.30	0.28	0.00	$\Sigma = 0.97$

head/chest cluster into a head and a chest cluster on the second dimension.

The results from the divisive method and the Hierarchical GROUPALS method are very similar. For the divisive method the fit is .957 and the overall SS is 69.1, for the Hierarchical GROUPALS method this is .973 and 68.4 respectively. The only difference between the two trees concerns the clustering of body and neck. The divisive method joins neck with the "face" cluster, while Hierarchical GROUPALS joins neck with body. The reason for this is that the overall SS is minimal if neck joins body: in the divisive solution the within-group SS summed over levels 11 en 12 is higher then the within-group SS summed over levels 11 en 12 in the Hierarchical GROUPALS solution. The squared cluster value for cluster 12 is relatively high, so, besides the 4-cluster partition also a 5-cluster partition could be considered when the Hierarchical GROUPALS method is used.

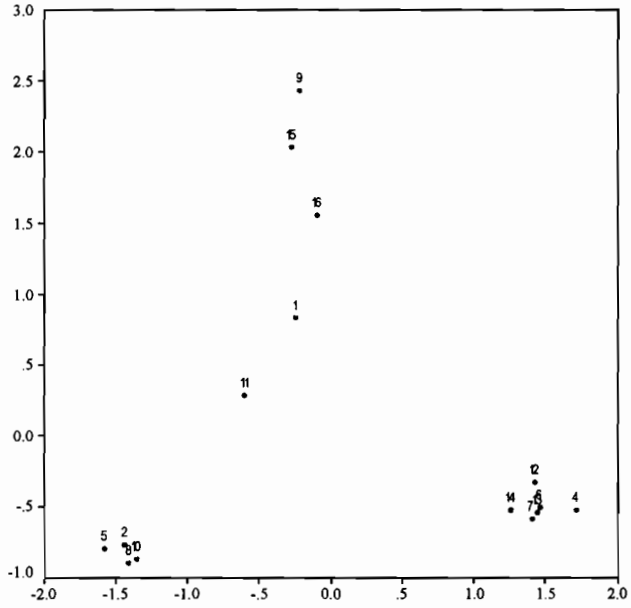


Figure 7: Plot of the object scores for the body data

Table 4: Coordinates for the objects in Figure 7

	Dimension			
Object	1	2	3	4
1	-0.25	0.84	0.02	0.71
2	-1.44	-0.76	-0.07	0.16
3	-1.41	-0.89	-0.08	0.09
4	1.71	-0.52	-1.87	-0.27
5	-1.58	-0.79	-0.08	-0.24
6	1.46	-0.50	-1.54	-0.12
7	1.40	-0.58	1.79	0.01
8	-1.41	-0.89	-0.08	0.09
9	-0.22	2.44	0.07	0.03
10	-1.36	-0.86	-0.08	0.13
11	-0.61	0.29	-0.03	-0.11
12	1.42	-0.32	-1.53	0.05
13	1.44	-0.54	1.83	-0.30
14	1.25	-0.52	1.56	0.15
15	-0.28	2.04	0.10	-0.35
16	-0.10	1.56	-0.01	-0.03

5.2 Whales Data

The whales data consist of 36 different types of Cetacea (whales, porpoises and dolphins), described by 15 variables regarding physical forms, bone structures and behaviour as presented in Table 6 (from Vescia, 1985a). There exist various classifications of Cetacea, of which the one by Grasse (1955; cited in Vescia, 1985b) is presented in Table 5.

Table 5: Classification of Cetacea according to Grasse

		family	No.
Baleen whales		Baleen whales	1- 3
		Grey whale	4
		Finback whales	5- 7
Toothed whales	Physeteroidea	Sperm whales	8- 9
		Beaked whales	28-32
	Delphinoidea	Dolphins	10-23
		Porpoises	24-25
		White whales	26-27
	Platanistoidea	River dolpins	33-36

In Van Buuren (1986) the data were analysed with K-means GROUPALS using K=9 and a dimensionality of 8. Except for variables 8 and 13, which were treated as single ordinal, the variables were considered to be multiple nominal. Missing values were treated as single category (see Gifi, 1990). The same analysis of the data with Hierarchical GROUPALS results in a solution that has a fit that is slightly higher (.004) than the fit of the K-means solution. The squared cluster "loadings" are given in Table 7 and the fit per variable in Table 8. As explained in section 5.1 these tables can be used to identify the variables that contribute to the split of a cluster. The tree resulting from Hierarchical GROUPALS is presented in Figure 8. Cutting the tree at level 28 gives a 9-cluster partition. This partition is much the same as the K-means partition. The difference concerns the clustering of the Delphinoidea. In the K-means partition the Dolphin cluster has 7 members, 2 Dolphins join the Porpoises and 5 Dolphins join the White Whales. In the partition resulting from the hierarchical clustering, the Dolphin cluster has 9 members, the Porpoises and White Whales are in one cluster, together with 4 Dolphins, and 1 Dolphin is in a cluster on his own; this Dolphin differs from most of the other Dolphins on the variables habitat (cold seas) and

Table 6: Whales data

Var.:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	fam.	No.
	1	6	1	1	1	2	3	1	4	5	2	3	1	3	4	1	1
	1	6	1	1	1	2	3	1	4	2	2	3	1	3	4	1	2
	1	2	1	1	4	1	3	1	4	1	2	3	1	3	4	1	3
	1	2	1	1	1	4	4	1	4	4	1	3	1	3	4	2	4
	1	5	1	1	4	1	5	3	4	1	1	3	1	5	4	3	5
	1	5	1	1	4	4	5	3	4	4	1	3	1	5	4	3	6
	1	5	1	1	4	4	5	3	4	4	1	3	1	6	4	3	7
	1	1	1	1	2	1	1	1	1	1	1	3	4	5	1	4	8
	1	1	1	1	1	1	1	1	1	1	1	1	4	2	1	4	9
	1	2	2	2	3	3	2	1	2	5	2	2	2	3	1	5	10
	1	3	2	3	3	3	2	1	2	1	2	2	2	2	2	5	11
	1	4	2	1	3	4	2	1	2	4	2	2	2	2	1	5	12
	1	4	2	1	3	4	1	1	2	1	2	2	2	5	1	5	13
	1	3	2	3	3	3	2	1	2	1	2	2	2	3	2	5	14
	1	2	2	2	1	3	2	1	2	1	2	2	2	2	2	5	15
	2	4	2	1	3	1	2	1	2	2	2	2	2	4	2	5	16
	1	4	2	1	3	2	2	1	2	5	2	2	2	3	3	5	17
	1	4	2	1	3	4	2	1	2	2	2	2	2	5	1	5	18
	1	3	2	3	3	3	2	1	2	3	2	2	2	4	2	5	19
	1	3	2	3	3	3	2	1	2	1	2	2	2	4	2	5	20
	1	3	2	3	3	3	2	1	2	4	2	2	2	5	2	5	21
	1	2	2	3	3	3	2	1	2	1	2	2	2	2	2	5	22
	1	3	2	3	3	3	2	1	2	1	2	2	2	5	2	5	23
	1	2	2	1	1	1	2	1	2	4	2	2	2	5	2	6	24
	1	2	2	1	2	1	2	1	2	1	2	2	2	5	2	6	25
	2	4	2	1	1	2	2	1	2	3	1	2	3	3	2	7	26
	2	4	2	1	1	2	1	1	2	4	1	2	3	3	1	7	27
	2	3	2	3	3	1	1	2	3	1	2	2	3	3	1	8	28
	2	3	2	3	3	1	1	2	3	1	2	2	3	5	1	8	29
	2	2	2	3	3	1	1	2	3	1	2	2	3	5	1	8	30
	1	4	2	3	3	1	1	2	3	1	2	2	3	3	1	8	31
	1	2	2	3	3	1	1	2	3	4	2	2	3	5	1	8	32
	2	3	2	4	2	2	2	1	3	1	1	1	5	1	2	9	33
	2	3	2	4	2	2	2	1	1	1	1	1	5	1	2	9	34
	2	3	2	4	2	2	2	1	3	2	1	1	2	1	2	9	35
	2	3	2	4	2	2	2	1	3	2	1	1	5	1	2	9	36

Variables		Categories
1	Neck	1 does not exist 2 exists
2	Form of head	1 cylindrical 2 conical 3 curved forehead 4 globular 5 flat 6 convex
3	Size of head	1 very big 2 medium size
4	Beak	1 missing 2 large 3 narrow and short 4 narrow and long
5	Dorsal fin	1 missing 2 triangular 3 falciform 4 backward and falciform
6	Flippers	1 small 2 large and short 3 medium 4 long and narrow
7	Set of teeth	1 lower jaw 2 lower and upper jaw 3 without, long baleens 4 without, thick baleens 5 without, large baleens
8	Furrows on throat	1 do not exist 2 small number 3 big number
9	Blow hole	1 left side 2 right side 3 middle 4 middle, two holes
10	Color	1 ventral parts clearer than dorsal parts 2 blackish 3 no pigmentation 4 spotted 5 missing
11	Cervical vertebrae	1 free 2 partly or completely welded
12	Lachrymal and jugal bones	1 form one piece 2 are independent 3 missing
13	Head bones	1 symmetrical 2 slightly unsymmetrical 3 unsymmetrical 4 very unsymmetrical 5 missing
14	Habitat	1 rivers 2 temperate or warm seas 3 cold seas 4 coasts 5 variable 6 missing
15	Feeding	1 squid 2 fish 3 seal 4 plankton
	Family	1 Baleen whale 2 Grey whale 3 Finback whale 4 Sperm whale 5 Dolphin 6 Porpoise 7 White whale 8 Beaked whale 9 River dolphin

Table 7: Squared cluster "loadings"

	dimension								
Cluster	1	2	3	4	5	6	7	8	$\Sigma = \mu^2$
35	45.73	0.01	1.00	0.23	0.00	0.29	0.06	0.01	47.33
34	0.50	27.08	0.34	1.55	0.03	0.13	0.02	0.01	29.66
33	2.40	0.26	15.68	1.24	0.05	0.11	0.70	0.00	20.44
32	1.10	2.86	0.16	8.64	3.67	0.94	0.29	0.05	17.71
31	0.55	0.72	0.02	4.94	8.05	0.28	0.00	0.13	14.70
30	0.22	0.00	0.93	1.85	0.00	8.99	0.02	0.72	12.73
29	0.01	0.00	0.88	0.00	0.00	1.01	5.78	1.77	9.45
28	0.00	0.08	0.82	0.21	0.43	0.41	2.66	3.97	8.58

Table 8: Fit per variable

		dimension							
Var.	Row sum	1	2	3	4	5	6	7	8
Multiple fit									
1	0.57	0.15	0.22	0.02	0.00	0.12	0.03	0.02	0.01
2	3.52	0.81	0.21	0.45	0.79	0.43	0.58	0.18	0.08
3	1.00	0.75	0.07	0.01	0.08	0.04	0.05	0.00	0.01
4	2.09	0.44	0.82	0.13	0.16	0.11	0.37	0.01	0.05
5	1.95	0.77	0.70	0.12	0.20	0.05	0.01	0.08	0.03
6	2.07	0.25	0.41	0.39	0.33	0.18	0.38	0.11	0.01
7	3.76	0.93	0.03	0.86	0.69	0.21	0.11	0.63	0.30
9	2.66	0.89	0.55	0.21	0.23	0.60	0.09	0.05	0.04
10	1.07	0.12	0.14	0.24	0.03	0.03	0.14	0.20	0.15
11	0.81	0.06	0.56	0.02	0.01	0.12	0.02	0.01	0.01
12	1.84	0.83	0.88	0.04	0.02	0.01	0.04	0.00	0.01
14	1.97	0.38	0.76	0.31	0.21	0.20	0.04	0.02	0.05
15	2.58	0.90	0.04	0.33	0.29	0.07	0.24	0.31	0.41
Single fit									
8	1.00	0.33	0.01	0.46	0.08	0.11	0.00	0.01	0.01
13	0.81	0.40	0.30	0.06	0.04	0.00	0.00	0.00	0.00

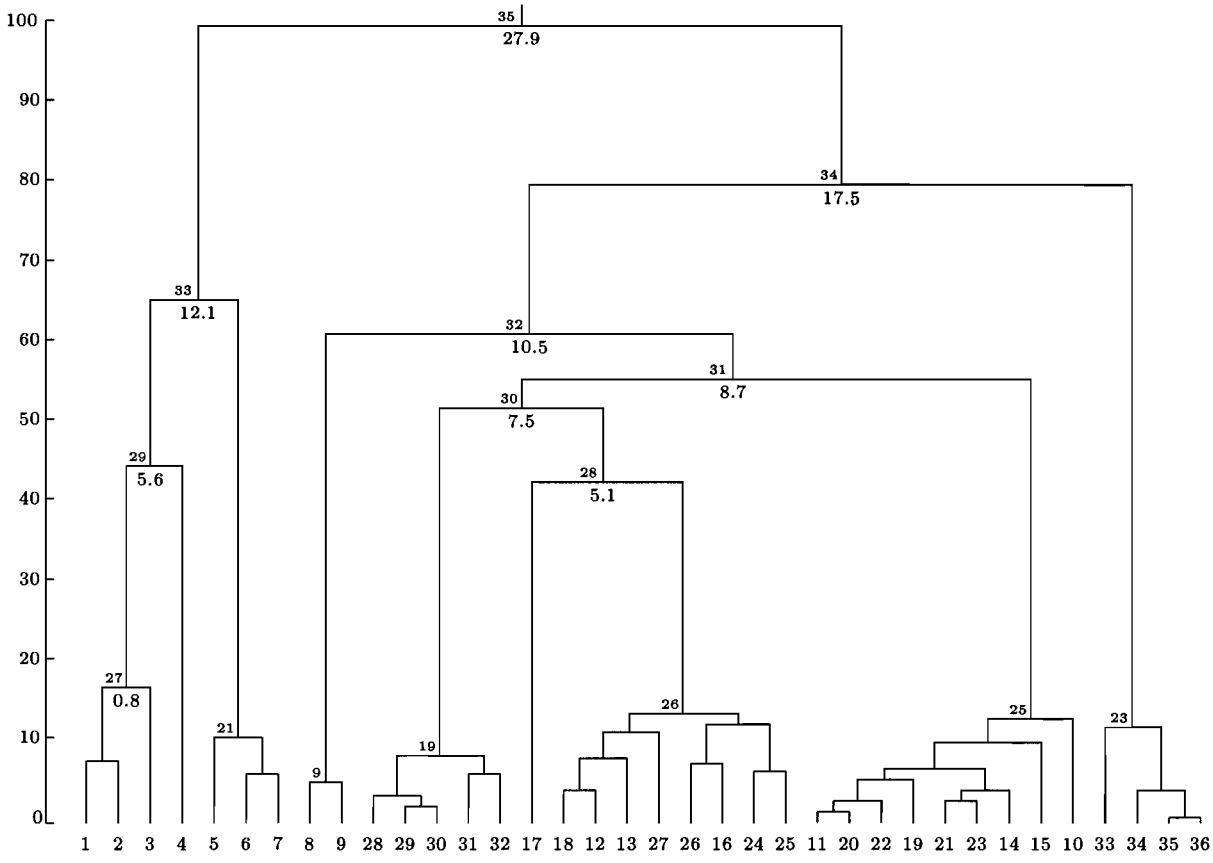


Figure 8: Tree and squared cluster values as percentages of the total variance, resulting from the Hierarchical GROUPALS method. The heights of the vertical edges represent the cluster values

color (missing) and differs from all other cetacea because it is the only one that eats seal. The Porpoises, the White Whales and the 4 Dolphins joining them all have a missing value on the variable Beak and, except for the Porpoises, have a globular form of the head. The two Dolphins joining the Porpoises in the K-means solution stay in the Dolphin cluster in the hierarchical solution because they do not have a globular form of the head and they do not have a missing value on the variable beak.

Figure 9 presents a plot of the first 2 dimensions of the object space. The objects are labeled with cluster numbers that correspond to the family numbers, except for object 17 (labeled with 7) and the objects of cluster 26 in the tree (labeled with 6). This representation combines the attractive features of a principal component analysis with optimal scaling with a simultaneously fitted hierarchical clustering. The object scores are given in Table 9.

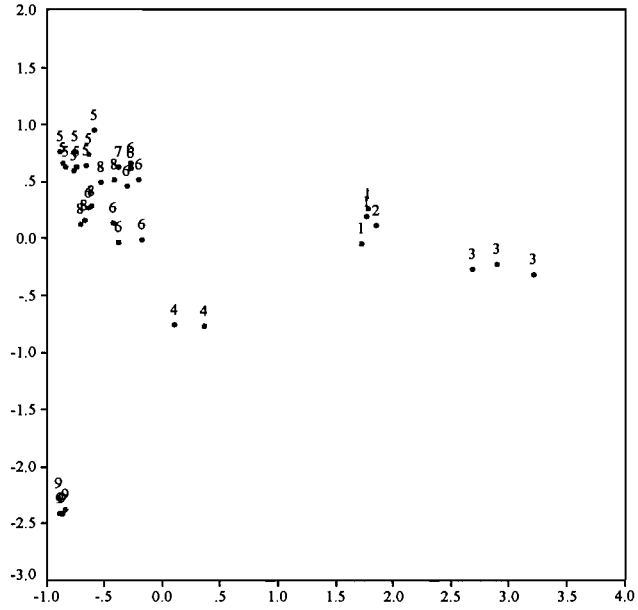


Figure 9: Plot of the object scores for the whales data

Table 9: Coordinates for the objects in Figure 9

Object	Cluster label	dimension							
		1	2	3	4	5	6	7	8
1	1	1.76	0.19	-2.35	-0.03	0.22	-0.46	-0.61	0.07
2	1	1.72	-0.04	-2.22	-0.08	0.23	-0.51	-0.25	0.68
3	1	1.77	0.26	-1.01	-0.20	0.42	-0.66	-0.13	0.16
4	2	1.85	0.11	-0.81	-0.13	-0.15	0.53	2.43	-1.05
5	3	2.68	-0.27	1.26	0.34	0.18	-0.25	-0.52	-0.02
6	3	2.89	-0.22	1.36	0.65	-0.17	0.07	-0.05	0.03
7	3	3.21	-0.31	1.65	0.91	-0.15	0.07	-0.20	-0.05
8	4	0.35	-0.76	0.37	-2.11	0.83	0.96	-0.41	-0.20
9	4	0.10	-0.75	0.20	-2.12	1.05	1.25	-0.34	-0.21
10	5	-0.59	0.96	-0.35	0.44	0.31	0.13	-0.28	-0.54
11	5	-0.84	0.63	0.18	0.58	0.83	0.14	-0.12	-0.08
12	6	-0.28	0.67	0.26	-0.06	-0.67	0.69	0.25	0.39
13	6	-0.28	0.62	0.39	-0.57	-0.74	0.19	0.07	0.36
14	5	-0.75	0.63	-0.02	0.59	0.57	-0.11	-0.06	-0.21
15	5	-0.64	0.74	-0.12	0.54	0.89	0.58	0.17	0.14
16	6	-0.64	0.27	-0.06	0.17	-0.70	0.29	-0.05	0.78
17	7	-0.38	0.63	-0.89	0.38	-1.60	0.73	-1.45	-1.65
18	6	-0.31	0.46	0.12	-0.07	-0.92	0.46	0.10	0.65
19	5	-0.89	0.77	0.15	0.83	0.62	0.22	0.01	0.19
20	5	-0.86	0.67	0.16	0.72	0.72	0.06	-0.06	0.00
21	5	-0.67	0.64	0.32	0.60	0.38	0.02	0.12	-0.01
22	5	-0.76	0.77	0.11	0.46	0.72	0.16	0.04	-0.05
23	5	-0.77	0.60	0.26	0.54	0.58	-0.06	-0.09	-0.03
24	6	-0.21	0.52	0.01	-0.07	-0.31	0.38	0.38	0.45
25	6	-0.43	0.14	0.07	-0.05	-0.09	0.26	0.03	0.37
26	6	-0.38	-0.03	-0.39	0.22	-0.84	0.69	0.16	0.58
27	6	-0.18	-0.01	-0.14	-0.57	-1.07	0.27	0.30	0.31
28	8	-0.68	0.16	0.32	-0.74	-0.10	-1.28	0.03	-0.32
29	8	-0.71	0.13	0.60	-0.78	-0.09	-1.23	0.00	-0.14
30	8	-0.62	0.28	0.53	-0.90	-0.20	-1.21	0.15	-0.12
31	8	-0.54	0.50	0.26	-0.87	-0.39	-0.90	-0.09	-0.23
32	8	-0.42	0.52	0.56	-0.81	-0.23	-0.95	0.29	-0.18
33	9	-0.90	-2.27	-0.11	0.58	-0.02	-0.32	0.07	-0.14
34	9	-0.84	-2.38	-0.14	0.29	0.33	0.36	-0.11	-0.13
35	9	-0.86	-2.41	-0.28	0.68	-0.22	-0.29	0.11	0.09
36	9	-0.89	-2.41	-0.27	0.67	-0.22	-0.28	0.10	0.09

6 Conclusion

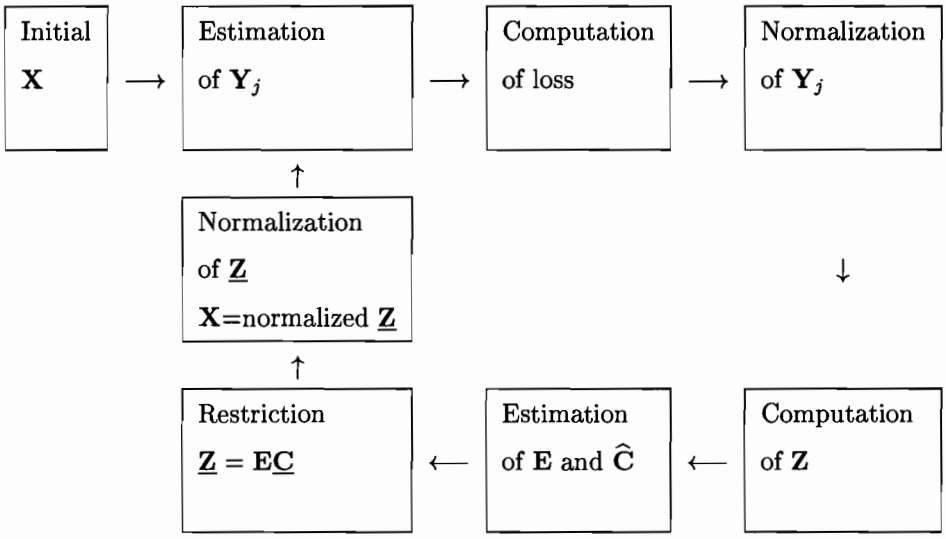
Hierarchical GROUPALS performs hierarchical clustering of objects measured on categorical variables under simultaneous scaling of the variables. The method is based on the alternating minimization of two components of the loss function; the scaling component and the clustering component. This minimization ensures monotone convergence of the algorithm. Because the clustering algorithm converges to a locally optimal tree, the method converges to a local minimum.

The method uses the orthonormal basis corresponding to the cluster hierarchy as the incidence matrix. This basis can also be used for a hierarchical decomposition of the object scores, resulting in a set of cluster indices that can be helpful in the interpretation and evaluation of the solution.

A FORTRAN 77 program has been developed to implement Hierarchical GROUPALS as described here. Because of time limits, as yet the method has not been tested extensively. Further research needs to be done to determine the sensitivity to local minima. Also, the next best method and the best first method should be compared with respect to processing time and performance.

APPENDIX

A Flow chart of the Hierarchical GROUPALS algorithm



B Steps of the algorithm for local tree operations

Preliminary remarks

1. When in the rebranching operation the path from S_a to the root meets the path from S_c to the root in S_p (for example, if $a = 28$, $a_k = 26$, and $c = 25$ in tree B of Figure 8, then $p = 31$), S_{a_k} should be removed from S_p and added to the same node S_p, so for $w \geq p$ nothing is changed by removing and adding S_{a_k}. In this case $\Sigma_w \ \tau_w \delta_w$, with w on the path from p to the root, is subtracted from the third component of (16) and the last term of (16) sums $\tau_w \iota_w$ over w on the path from c to $p-1$.

2. If evaluating the rebranching operation for $c = f$ and $e = f$ is followed by evaluating for $c = f$ and $e = f+1$, the last component of (16), i.e. $\Delta_\iota(c = f, e = f+1)$, can be easily computed by subtracting ι_f from $\Delta_\iota(c = f, e = f)$, and so on until $e = pf - 1$. Then if next the rebranching operation is evaluated for $c = pf$ and $e = c$, $\Delta_\iota(c, e = c)$ can be computed from $\Delta_\iota(c = f, e = pf - 1)$ by subtracting $\mathrm{SS}(SN_f)$ and adding $\mathrm{SS}(S_{a_k})$ and $\mathrm{SS}(S_c)$ ($\mathrm{SS}(SN_c)$ needs not to be added because $\mathrm{SS}(SN_c)$=$\mathrm{SS}(S_{a_k})$+$\mathrm{SS}(S_c)$+ι_c and ι_c is already included), and so on for all nodes on the path from S_f to the root. Because of this, in the step of evaluating the attachment, the algorithm starts with $c = f$, $n_f = 2$ (if $n_f = 2$ the path from S_f to the root is a path from the lowest to the highest level; these are the longest paths). First attachment to S_f on level $e = f$ is evaluated, then attachment to S_f on levels $e > f$ is evaluated (the reordering part) and then attachment to S_c (c being the next node on the path from S_f to the root) is evaluated, and so on for all nodes on the path. This process is repeated for all nodes f. A path from S_f to the root may meet a previously evaluated path in S_m (for example, if f is the right child of S_{19} in tree B of Figure 8 and the path from S_{26} to the root has previously been evaluated, then $m = 30$). In this case, if $c < m$, the increase in SS for adding S_{a_k} to S_w, with $w \geq m$, does not have to be computed (because this is already computed on the previously evaluated path), and, if $c = m$ the algorithm continues with the next f (if $c \geq m$, attachment to c is already evaluated on the previously evaluated path).

3. In the algorithm external nodes are stored as children of their parent, not as nodes. Because of this, in the rebranching operation c is an internal node. If rebranching to c is evaluated and one or both of the children of S_c is an external node, Δ for rebranching to S_{c_k} can be easily computed from Δ for rebranching to S_c, because the only difference between

S_{a_k} attached to S_c and S_{a_k} attached to S_{c_k} is a difference in the children of SN. As an example, consider the previously mentioned rebranching of object 3 in tree A of Figure 4 to S_4: rebranching object 3 to object 5 also results in tree B, except that the labels of the three right external nodes are 5 3 6 instead of 5 6 3. Because of this, evaluating the rebranching of S_{a_k} to an external node is done immediately after evaluating the rebranching of S_{a_k} to the parent of an external node. The difference in overall SS when S_{a_k} is rebranched to S_{c_k} is computed as:

$$\Delta_{c_k} = \mathrm{SS}(S_{a_k}) + \mathrm{SS}(S_c) - (\mathrm{SS}(S_{a_k} \text{ attached to } S_c) + \mathrm{SS}(S_{c_{(3-k)}})). \tag{19}$$

Then $\Delta = \Delta_c - (e-c+1)\Delta_{c_k}$ with Δ_c the difference in overall SS when S_{a_k} is rebranched to S_c.

Steps of the algorithm

FOR $a = 1$ to $N-1$

1. Determine change in overall SS when S_a would be reordered to b:

   FOR $b = \max(a_1,a_2)+1$ to $pa-1$

   $\Delta = (a-b)\mathrm{SS}(S_a)+(b-a)(\mathrm{SS}(S_{a_1})+\mathrm{SS}(S_{a_2}))$

   $\Delta = \Delta - \sum_{w=1}^{a-1} \mathrm{SS}(S_w)$ for S_w is top node at level a

   if $a > b$ then $\Delta = \Delta \sum_{w=1}^{b-1} \mathrm{SS}(S_w)$ for S_w is top node at level b

   if $a < b$ then $\Delta = \Delta \sum_{w=1}^{b} \mathrm{SS}(S_w)$ for S_w is top node at level b, and $w \neq a$

2. Determine decrease in overall SS when S_a would be removed, for levels a to $pa-1$:

   $\delta = \mathrm{SS}(S_a)$

   $\delta = \delta + \sum \mathrm{SS}(S_w)$ for S_w is top node at level a

   $\delta_a = \mathrm{SS}(S_a) - (\mathrm{SS}(S_{a_1})+\mathrm{SS}(S_{a_2}))$

   $\delta = \delta + (\tau_a - 1)\delta_a$

   FOR $k = 1$ to 2

3. Determine decrease in overall SS when S_{a_k} would be removed from nodes on the path from S_a to the root:

   FOR $w = a+1$ to $N-1$, for S_w on the path from S_a to the root

   $\delta_w = \frac{n_w n_{a_k}}{n_w - n_{a_k}} d^2(M_w, M_{a_k})$

   $\delta = \delta + \tau_w \delta_w$

4. Determine increase in overall SS when S_{a_k} would be attached to S_f on level $e = f$, for levels f to $pf - 1$:

$m = N - 1$

FOR $w = 1$ to $N - 2$

$\iota_w = 0$

FOR $f = 1$ to $N - 2$

if $n_f \neq 2$ then next f

if $f = a$, a_1, a_2, or m then next f

if S_f is not top node at level a_k then f = next node on path from S_f to root

$\iota = \mathrm{SS}(SN) = \mathrm{SS}(S_f) + \mathrm{SS}(S_{a_k}) + \frac{n_f n_{a_k}}{n_f + n_{a_k}} d^2(M_f, M_{a_k})$

$\iota_f = \mathrm{SS}(SN) - \mathrm{SS}(S_f) - \mathrm{SS}(S_{a_k})$

$\iota = \iota + (\tau_f - 1)\iota_f$ (if $f < a < f + \tau_f$ then $\iota = \iota + (\tau_f - 2)\iota_f$)

$\iota_{top} = \sum \mathrm{SS}(S_w)$ for S_w is top node at level f

5. Determine whether the path from S_a to the root meets the path from S_f to the root:

find $i \in S_{a_k}$

FOR $w = a + 1$ to $N - 1$

if $i \in S_w$, with S_w on path from S_f to root, then $p = w$

jump out of loop over w

if $p = N - 1$ then go to step 6

else $\delta = \delta - \tau_w \delta_w$ for S_w on path from S_p (inclusive) to root

6. Determine increase in overall SS when S_{a_k} would be added to nodes on the path from S_f to S_p:

FOR $w = f + 1$ to $p - 1$, for S_w on the path from S_f to S_p

if $\iota_w \neq 0$ then $m = w$

else $\iota_w = \frac{n_w n_{a_k}}{n_w + n_{a_k}} d^2(M_w, M_{a_k})$

$\iota = \iota + \tau_w \iota_w$

$\Delta = \iota + \iota_{top} - \delta$

7. Determine change in overall SS when S_{a_k} would be attachted to S_{f_j}:

FOR $j = 1$ to 2

if $f_j = 0$ and $a_k < f$ (then S_{f_j} is an object and top node)

then $\mathrm{SS}(SN) = \mathrm{SS}(S_{a_k}) + \frac{n_{a_k}}{n_{a_k} + 1} d^2(M_{a_k}, z_i)$

$\Delta_{f_j} = [\mathrm{SS}(S_{a_k}) + \mathrm{SS}(S_f)] - [\mathrm{SS}(SN) + \mathrm{SS}(S_{f_{(3-j)}})]$

$\Delta = \Delta - \Delta_{f_j}$

8. Determine increase in overall SS when S_{a_k} would be attached to S_f and to S_{f_j} on levels $e > f$:

$\iota_s = \iota$

FOR $t = f + 1$ to $f + t_f - 1$

if $t = a$ then $t = t + 1$

$e = t$

$\iota_s = \iota_s - \iota_f$

$\iota_{top} = \sum \mathrm{SS}(S_w)$ for S_w is top node at level e

$\Delta = \iota_s + \iota_{top} - \delta$

FOR $j = 1$ to 2

if $\Delta_{f_j} > 0$ then $\Delta = \Delta - (e - f + 1)\Delta_{f_j}$

if $f < a < e$ then $\Delta = \Delta - (e - f)\Delta_{f_j}$

9. Determine increase in overall SS when S_{a_k} would be attached to S_c and to S_{c_j}, with c on the path from S_f to the root, for $e = c$:

FOR $c = f + 1$ to $N - 2$, for S_c on the path from S_f to the root,

with $c \neq a$, a_1, or a_2

if $c = m$ then next f

$e = c$

$\iota_s = \iota_s - \mathrm{SS}(SN)$

$\mathrm{SS}(SN) = \mathrm{SS}(S_c) + \mathrm{SS}(S_{a_k}) + \iota_c$

$\iota_s = \iota_s + \mathrm{SS}(SN) - \iota_c$

$\iota_{top} = \sum \mathrm{SS}(S_w)$ for S_w is top node at level e

$\Delta = \iota_s + \iota_{top} - \delta$

Repeat step 7 with c instead of f

10. Determine increase in overall SS when S_{a_k} would be attached to S_c and to S_{c_j} on levels $e > c$:

FOR $t = c + 1$ to $c + \tau_c - 1$

if $t = a$ then $t = t + 1$

$e = t$

$\iota_s = \iota_s - \iota_c$

$\iota_{top} = \sum \mathrm{SS}(S_w)$ for S_w is top node at level e

$\Delta = \iota_s + \iota_{top} - \delta$

FOR $j = 1$ to 2

if $\Delta_{c_j} > 0$ then $\Delta = \Delta - (e - c + 1)\Delta_{c_j}$

C Operating GROUPALS

Information on operating the GROUPALS program can be found in Van Buuren (1986). The only adjustment needed to perform Hierarchical GROUPALS concerns card number 3 of the GROUPALS control cards; between ILEV and METH, KMHI should be inserted:

0 = K-means

1 = hierarchical divise method

2 = hierarchical optimization method

References

Chandon, J. L., Lemaire, J., & Pouget, J. (1980). Construction de l'ultramétrique la plus proche d'une dissimilarité au sens des moindres carrés. *R.A.I.R.O. Recherche opérationelle*(14), 157-170.

De Soete, G., DeSarbo, W., & Carroll, J. D. (1985). Optimal variable weighting for hierarchical clustering: an alternating least-squares algorithm. *Journal of classification*(2), 179-192.

Everitt, B. S. (1980). *Cluster analysis* (2nd ed.). London: Heinemann Educational Books Ltd.

Gifi, A. (1990). *Nonlinear multivariate analysis.* Chichester: Wiley.

Gordon, A. D. (1996). Hierarchical classification. In P. Arabie, L. J. Hubert, & G. De Soete (Eds.), *Clustering and classification* (p. 65-121). River Edge, NJ: World Scientific Publishing.

Hartigan, J. A. (1967). Representation of similarity matrices by trees. *Journal of the American Statistical Association*(62), 1140-1158.

Miller, G. (1968). Algebraic models in psycholinguistics. In C. Vlek (Ed.), *Algebraic models in psychology.* Proceedings of the NUFFIC International Summer Session at "Het Oude Hof", The Hague.

Mirkin, B. (1995). A linear embedding of the binary hierarchy and its application to clustering and querying. In J. Albus, A. Meystel, D. Pospelov, & T. Reader (Eds.), *Proceedings of 1995 isic workshop (10th ieee international symposium on intelligent control)* (p. 259-269). Bala Cyncyd, Pa.: Adrem.

Rosenberg, S. (1982). The method of sorting in multiariate research with applications selected from cognitive psychology and person perception. In N. Hirchberg & L. Humphreys (Eds.), *Multivariate applications in the social sciences* (p. 117-142). Hillsdale, NJ: L. Erlbaum Assoc.

Späth, H. (1985). *Cluster dissection and analysis.* Chichester: Ellis Horwood Ltd.

Van Buuren, S. (1986). *Groupals: a method to cluster objects for variables with mixed measurement levels* (Research Report No. RR-86-10). Leiden: Department of Data Theory.

Van Buuren, S., & Heiser, W. J. (1989). Clustering n objects into k groups under optimal scaling of variables. *Psychometrika*(54), 699-706.

Vescia, G. (1985a). Automatic classification of cetaceans by similarity aggregation. In

J. F. Marcotorchino, J. M. Proth, & J. Janssen (Eds.), *Data analysys in real life environment: Ins and outs of solving problems* (p. 15-24). Amsterdam: North-Holland.

Vescia, G. (1985b). Descriptive classification of cetacea: Whales, porpoises and dolphins. In J. F. Marcotorchino, J. M. Proth, & J. Janssen (Eds.), *Data analysys in real life environment: Ins and outs of solving problems* (p. 7-13). Amsterdam: North-Holland.

Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association,*(58), 236-244.