Journal of Classification 11:171-193 (1994)

Pyramidal Classification Based on Incomplete Dissimilarity Data

Wolfgang Gaul
University of Karlsruhe

Martin Schader
University of Mannheim

Abstract: Two algorithms for pyramidal classification — a generalization of hierarchical classification — are presented that can work with incomplete dissimilarity data. These approaches — a modification of the pyramidal ascending classification algorithm and a least squares based penalty method — are described and compared using two different types of complete dissimilarity data in which randomly chosen dissimilarities are assumed missing and the non-missing ones are subjected to random error. We also consider relationships between hierarchical classification and pyramidal classification solutions when both are based on incomplete dissimilarity data.

Keywords: Cluster analysis, Missing values, Monte Carlo evaluation, Penalty approach, Pyramidal classification.

1. Introduction

1.1 Motivation

The motivation of this paper is twofold. First, the problem as to how missing values in dissimilarity data can be handled is of general interest (as a

The authors are indebted to anonymous referees who helped to improve an earlier version of this paper and to the editor for considerable assistance in ''polishing'' their English.

Authors' Addresses: Wolfgang Gaul, Institute of Decision Theory and Operations Research, University of Karlsruhe, Germany, and Martin Schader, Department of Information Systems, University of Mannheim, Germany.

Of course, condition (3b) is weaker than condition (3a). Additionally, for every hierarchy H on I there always exists a total order $\leq$ on I such that every $K \in H$ is an interval with respect to that order. For example, consider any corresponding tree representation of H and order the leaves/singletons, e.g., from left to right. Thus, one can state

Proposition 1: *Every hierarchy on I is a pyramid on I.*

To produce graphical representations of hierarchies and pyramids, an index for the elements of a hierarchy or a pyramid is needed. Let $f: S \rightarrow \boldsymbol{R}$ be a mapping that assigns real numbers to the elements of $S \subset 2^I$. Potentially useful conditions for f as an index function are

$$f(K) = 0 \Leftrightarrow |K| = 1, \quad \forall K \in S.$$

$$(|K| \text{ denotes the cardinality of } K.) \tag{5}$$

$$K \subset L \Rightarrow f(K) \leq f(L), \quad \forall K, L \in S. \tag{6}$$

$$K \subset L, K \neq L \Rightarrow f(K) < f(L), \quad \forall K, L \in S. \tag{7a}$$

$$K \subset L, K \neq L, f(K) = f(L) \Rightarrow \exists J_1, J_2 \in S:$$

$$K \neq J_1, K \neq J_2, \text{ and } K = J_1 \cap J_2. \tag{7b}$$

Standardization of f according to condition (5) is no restriction because addition of a suitable constant is always possible.

Condition (7b) prevents chaining of clusters on the same index level (see Appendix A for an example). Two non-nested clusters J_1 and J_2 with the same index may, however, generate K via conditions (3a,b).

Definition 2:

(2.1) (H,f) is an *indexed hierarchy* on I if H is a hierarchy on I and f satisfies conditions (5) and (6).

(2.2) (P,f) is an *indexed pyramid* on I if P is a pyramid on I and f satisfies conditions (5) and (6).

(2.3) (S,f) is *strictly indexed* if (S,f) is an indexed hierarchy/pyramid and f satisfies (7a).

(2.4) (S,f) is *semi-strictly indexed* if (S,f) is an indexed hierarchy/pyramid and f satisfies (7b).

The graphical representation of indexed hierarchies by dendrograms is well-known (e.g., Kruskal, Landwehr, and McKae 1985). For the graphical representation of indexed pyramids a similar procedure can be used.

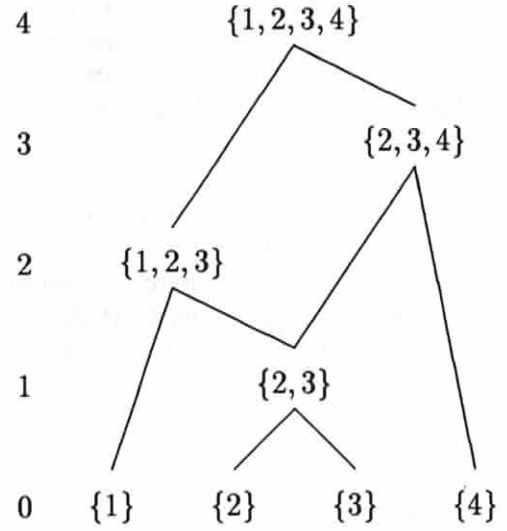

Figure 1a. Graphical Representation of an Indexed Pyramid.

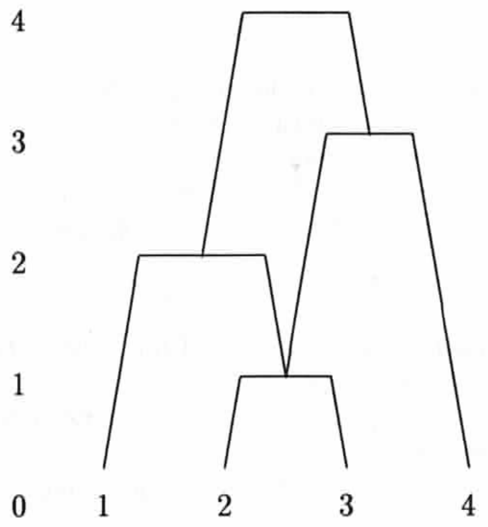

Figure 1b. Simplified Graphical Representation of the Indexed Pyramid of Figure 1a.

Example 1:

Let $I = \{1, 2, 3, 4\}$, $P = \{ \{1\}, \{2\}, \{3\}, \{4\}, \{2, 3\}, \{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\} \}$. Clearly, $1 < 2 < 3 < 4$ is a total order on I such that every element of P is an interval with respect to this order. Let f be given by $f(\{i\}) = 0$, $\forall\, i \in I$, and $f(\{2, 3\}) = 1$, $f(\{1, 2, 3\}) = 2$, $f(\{2, 3, 4\}) = 3$, $f(\{1, 2, 3, 4\}) = 4$.

Now (P,f) is a strictly indexed pyramid and its graphical representation is displayed in Figure 1a. As can be seen in this example, pyramidal classification allows non-nested overlapping of clusters. Furthermore, note that each cluster can have at most two successors with respect to the order relation "$\subset$" on 2^I. Figure 1b shows a simplified version of Figure 1a. This type of simplified output will be used in the following.

To provide an agglomerative algorithm using dissimilarities between pairs of objects and to represent the structure underlying such data via hierarchies and pyramids, further notation and definitions are needed.

Let $d: I^2 \rightarrow \mathbb{R}_+$ be a mapping that assigns non-negative real numbers to pairs of objects of the underlying set I. Potentially useful conditions for d are (d_{ij} is used as an abbreviated notation for $d(i,j)$)

$$d_{ij} = 0 \Leftrightarrow i = j\,. \tag{8}$$

$$d_{ij} = d_{ji}, \quad \forall\, i, j \in I\,. \tag{9}$$

$$d_{ik} \leq \max\,\{d_{ij}, d_{jk}\}, \quad \forall\, i, j, k \in I\,. \quad \text{(ultrametric condition)} \tag{10}$$

There exists a total order $\leq$ on I such that $d_{ik} \geq \max\,\{d_{ij}, d_{jk}\}$,

$$\forall\, i, j, k \in I \text{ with } i < j < k\,. \quad \text{(pyramidal condition)} \tag{11}$$

As it is known that potentially negative-valued measures of association between pairs of objects are sometimes used as input to clustering applications, one may ask whether the non-negativity of d is an important restriction. Note that we have standardized the index f according to condition (5) and that there are interrelationships between f and corresponding dissimilarity data that we will have to take into consideration in the algorithms to be described later. Thus, any measure of association one wants to use as input should first be transformed to non-negative dissimilarities. Note, additionally, that "dissimilarity (data)" or "dissimilarities" are used as generic terms throughout the paper. We add specific adjectives, e.g., "empirical" or "ultrametric", to denote different types of dissimilarity data.

Definition 3:

(3.1) d is an *ultrametric dissimilarity* on I if d satisfies conditions (8), (9), and (10).

	1	2	3	4	5	6
1	0					
2	7	0				
3	1	7	0			
4	5	4	5	0		
5	7	2	7	5	0	
6	4	6	4	3	6	0

Table 1a: Dissimilarity Data Between Pairs of Objects for $I=\{1,...,6\}$

	3	1	6	4	2	5
3	0					
1	1	0				
6	4	4	0			
4	5	5	3	0		
2	7	7	6	4	0	
5	7	7	6	5	2	0

Table 1b: Rearranged Dissimilarity Data of Table 1a According to the Total Order $3 \prec 1 \prec 6 \prec 4 \prec 2 \prec 5$

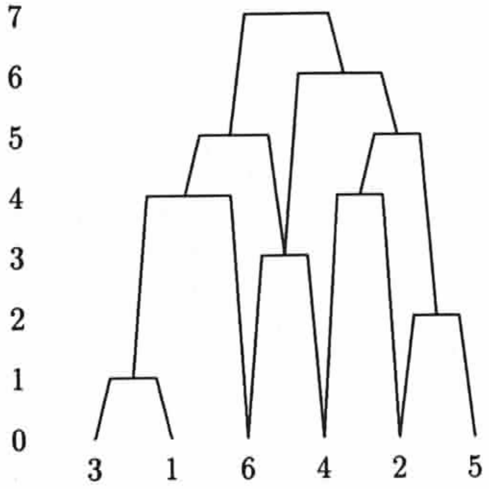

Figure 2a. Pyramidal Classification (Indexed Pyramid) of the Dissimilarity Data of Tables 1a,b.

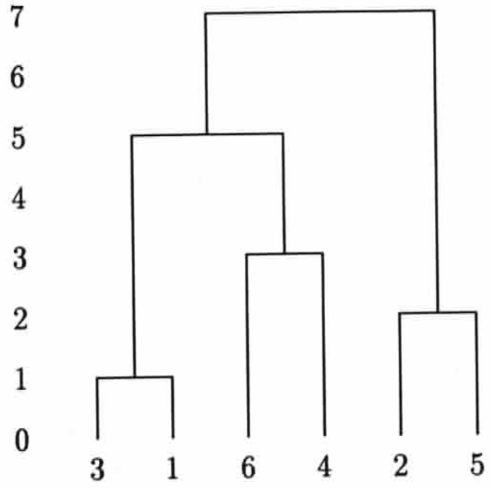

Figure 2b. Hierarchical Classification (Indexed Hierarchy, Complete-Linkage) of the Dissimilarity Data of Tables 1a,b.

2. As the model on which pyramidal classification is based is weaker than hierarchical classification models, one can expect that in general pyramids will fit dissimilarity data "better" than dendrograms. However, "better" has to be explained, e.g., in terms of differences of goodness-of-fit values. Of course, whenever a stronger model fits the data equally well or the difference of corresponding fit values is "small", the stronger model has a good chance to be preferred.
3. The dendrogram solution of Figure 2b will not change if such values as $\delta_{15}, \delta_{51}, \delta_{24}, \delta_{42}, \delta_{45}, \delta_{54}$ are missing in the dissimilarity data (i.e., 20% of the given data are missing), whereas the pyramidal representation of this first example of incomplete dissimilarity data would be affected by such missing values.

We postpone discussing the differential effects on hierarchical and pyramidal classification of incomplete dissimilarity data. Instead, we consider different approaches to pyramidal clustering and their relation to the problem of missing data.

2. Two Approaches Using Pyramidal Classification to Handle Missing Values in Dissimilarity Data

2.1. The PACII (Pyramidal Ascending Classification with Incomplete Information) Modification

We consider whether the PAC (Pyramidal Ascending Classification) algorithm as described, e.g., in Diday (1986, 1987) and Diday and Bertrand (1986) can be modified to cope with missing values in the input data. PACII (Pyramidal Ascending Classification with Incomplete Information) will be used as a label to distinguish our modified version from the original PAC algorithm. Our outline of the PACII algorithm emphasizes two points: First, the PACII algorithm starts with clusters each consisting of a single element of the set I. Using the available dissimilarities, the closest pair of clusters is joined to form a new cluster. (Ties can be arbitrarily broken.) This procedure is repeated until the set I forms a cluster or no more clusters can be joined. Second, in the PACII algorithm, a total order has to be generated simultaneously on I. For this purpose, a starting partial order on I defined by $i \leq j$ if and only if $i = j$ is used when the PACII algorithm begins execution. During all intermediate stages of PACII this partial order is now successively completed, using the current set of clusters until, finally, $\leq$ is a complete order on I or no more clusters can be joined.

By way of a more formal description, first note that each cluster K that is generated by PACII will be subset of one of the equivalence classes of the transitive closure of the reflexive symmetric relation $\approx$ defined by $i \approx j \Leftrightarrow (i \leq j \text{ or } j \leq i)$. This equivalence class will be denoted by $[K]$. Recall that δ_{ij} is a notation for the given dissimilarity value between objects i and j. Denote by $M \subset I^2$ the set of pairs of objects for which the dissimilarity values are missing. Then, a brief sketch of the PACII algorithm must emphasize the following steps (where D is a real-valued function on $2^I \times 2^I$, and min and max denote minimal and maximal elements with respect to $\leq$):

(S1) Initialize $D(\{i\},\{j\}) = \delta_{ij}, \quad \forall (i,j) \in I^2 - M; \quad P = \{\{i\}: i \in I\}$; $Q = \{(\{i\},\{j\}):(i,j) \in M\} \cup \{(J,J): J \in 2^I\}$; and $f(\{i\}) = 0,\ i \leq i$, $\forall\, i \in I; N(J) = 0, \forall\, J \in 2^I$. ($N(\cdot)$ counts the number of successors.)

(S2) If $I \in P$ or $P^2 - Q = \varnothing$ then stop, otherwise find $(K,L) \in P^2 - Q$ with minimal $D(K,L)$.

(S3) If K and L are *linkable* (more precisely, if the Boolean-valued function *linkable* defined in Appendix B returns *linkable* (K,L) = true) then

If $[K] \neq [L]$ then complete $\leq$ such that $[K] \cup [L]$ is totally ordered (see Remark 1). Update $P = P \cup \{K \cup L\}$, $f(K \cup L) = D(K,L)$, $N(K) = N(K) + 1$, and $N(L) = N(L) + 1$. For all $J \in P$ compute $D(K \cup L,J)$ (see Remark 2) and let $D(J,K \cup L) = D(K \cup L,J)$; if $D(K \cup L,J)$ is missing then update $Q = Q \cup \{(K \cup L,J), (J,K \cup L)\}$.

(S4) Update $Q = Q \cup \{(K,L), (L,K)\}$, and go back to (S2).

Remark 1: There are always two plausible strategies for generating $\leq$ on $[K] \cup [L]$. Let us analyze the first case, $\max K = \max\,[K]$ and $\max L = \max\,[L]$. Here, the maximal elements of K and L are also maximal elements of their respective classes $[K]$ and $[L]$. Strategy 1 would be to reverse the existing (possibly partial) order on $[L]$ and to put $i < j$ for all $i \in K$ and $j \in L$. Strategy 2 would be to reverse the existing order on $[K]$ and to put $i < j$ for all $i \in L$ and $j \in K$. Analogously, in the second case, $\max K = \max\,[K]$ and $\min L = \min\,[L]$, we can put $i < j$ for all $i \in K$ and $j \in L$, or first reverse both orders (on $[K]$ and $[L]$) and then put $i < j$ for all $i \in L$ and $j \in K$. The other cases are similar and are omitted here.

Remark 2: Depending on how $D(\cdot,\cdot)$ is determined between clusters, different versions of PACII are available. The compete-linkage version, e.g., would use

$$D(K \cup L,J) = \begin{cases} \max\{\delta_{ij}:(i,j) \in (K \cup L) \times J - M\} & \text{, if } (K \cup L) \times J - M \neq \varnothing\,, \\ \text{missing} & \text{, otherwise}\,. \end{cases}$$

	1	2	3	4
1	0			
2	x	0		
3	2	1	0	
4	4	3	x	0

Table 2: Example Dissimilarity Data (Missing Values Denoted by x) Between Pairs of Objects for $I = \{1, 2, 3, 4\}$.

K	L	$D(K, L)$	relation $\preceq$
$\{2\}$	$\{3\}$	1	$2 \prec 3$
$\{1\}$	$\{3\}$	2	$1 \prec 3,\ 2 \prec 3$
$\{2, 3\}$	$\{1, 3\}$	2	$2 \prec 3 \prec 1$
$\{2\}$	$\{4\}$	3	$2 \prec 3 \prec 1,\ 2 \prec 4$
$\{2, 3\}$	$\{2, 4\}$	3	$2 \prec 3 \prec 1,\ 4 \prec 2 \prec 3$
$\{1, 2, 3\}$	$\{2, 3, 4\}$	4	$4 \prec 2 \prec 3 \prec 1$

Table 3: Intermediate Results of PACII when Applied to the Data of Table 2.

Note that because of the possibility of missing values, we do not recommend usage of recurrence formulas (Lance and Williams 1967).

Example 3:

If PACII is applied to the dissimilarity data of Table 2 where missing values are denoted by x then, e.g., the results presented in Table 3 show how PACII works.

2.2. The PLSC (Pyramidal Least Squares Classification) Technique

Another approach to pyramidal classification based on incomplete dissimilarity data could be the following: Solve a constrained optimization problem to find a pyramidal dissimilarity d on I that ''best'' fits the given dissimilarity data δ (with possibly missing values).

Since penalty formulations of constrained optimization belong to the well-known approaches in cluster analysis and can be adapted to handle missing values, we will briefly describe the following algorithm that we have called PLSC (Pyramidal Least Squares Classification). (See, e.g., Arabie and Carroll (1980), and Carroll and Arabie (1983) for their experiences with penalty formulations of clustering problems, and De Soete (1984), and Schader and Gaul (1992) for penalty approaches to hierarchical classification based on incomplete dissimilarity data.)

Two types of iterations merit description here. First, in an outer iteration the PLSC technique starts with an initial total order on I and subsequently updates the actual total order using the DD (Doubles Décalages, or Double Swapping) method (see e.g., Marcotorchino and Michaud (1979, pp. 166-172) or our short description in Appendix C). Second, in an inner iteration, based on the actual total order on I, the PLSC technique solves a penalty approach to fit a pyramidal dissimilarity d to the given dissimilarity data δ using Powell's (1977) conjugate gradient procedure with automatic restarts. (Also see De Soete (1984) for an application of this procedure within a penalty approach for hierarchical classification.)

A brief outline of the PLSC technique emphasizes the following steps:

(S1) Choose an initial total order $\leq$ on I. Describe this total order and the total orders generated in the following steps by a vector $\mathbf{x} = (\ldots,x_{ij},\ldots)$ with

$$x_{ij} \in \{0,1\}, \ \forall\, i,j \in I,$$

$$x_{ii} = 1, \ \forall\, i \in I \text{ (reflexivity)},$$

$$x_{ij} + x_{ji} = 1, \ \forall\, i,j \in I \text{ (antisymmetry and completeness)},$$

$$x_{ij} + x_{jk} - x_{ik} \leq 1, \ \forall\, i,j,k \in I \text{ (transitivity)}.$$

Set $\mathbf{y} = \mathbf{x}$, and $F = \infty$.

(S2) Solve the constrained optimization problem
Minimize $F(d^x) = \sum_{(i,j) \in I^2 - M} (\delta_{ij} - d^x_{ij})^2$ subject to the constraints

$$d^x_{ik} \geq d^x_{ij}\, x_{ij}\, x_{jk} \text{ and } d^x_{ik} \geq d^x_{jk}\, x_{ij}\, x_{jk}, \ \forall\, i,j,k \in I\,,$$

$$d^x_{ij} \in \mathbb{R}_+, \ d^x_{ij} = 0 \Leftrightarrow i = j, \text{ and } d^x_{ij} = d^x_{ji}, \ \forall\, i,j \in I\,;$$

e.g., via a penalty approach.
If $F(d^x) < F$ then update $\mathbf{y} = \mathbf{x}$, $d^y = d^x$, $F = F(d^x)$ and go to (S3); otherwise go to (S4).

(S3) Take $\mathbf{y}$ and (re)start the DD method creating a new total order $\mathbf{x}_{DD}$ from $\mathbf{y}$. Set $\mathbf{x} = \mathbf{x}_{DD}$ and go back to (S2).

(S4) Take **x** and check whether the DD method can be continued.
If not then stop, with the results **y** and d^y,
otherwise continue the DD method creating a new total order $\mathbf{x}_{DD}$ from **x**, set $\mathbf{x} = \mathbf{x}_{DD}$ and go back to (S2).

One possible initialization for PLSC is the total order resulting from a PACII application. In that case, the least squares based goodness-of-fit yielded by PLSC will never be worse than the corresponding PACII value. Within the outer iteration of PLSC any other sub-algorithm suitable for generating and updating total orders on I can be incorporated instead of using the DD method. Note that the decision on how to update/generate total orders on I is one of the crucial points in PLSC (as well as in PACII).

3. Monte Carlo Evaluation

Both the PACII and PLSC algorithm have been applied to several data sets. Additionally, hierarchical classification techniques adapted to tackle the problem of missing values in dissimilarity data were used for comparison. Consistent with the labels PACII and PLSC, we use the notation HACII (Hierarchical Ascending Classification with Incomplete Information) and HLSC (Hierarchical Least Squares Classification) for the hierarchical counterparts of the pyramidal techniques. HACII is described in Schader and Gaul (1992) as modification of the well-known linkage procedures that can handle missing values in dissimilary data. HLSC refers to our version of De Soete's (1984) algorithm.

From the many results obtained only some selected findings can be presented. In this paper results are explained on the basis of an experimental design in which the following factors were varied: (a) The underlying type of error-free data (complete ultrametric dissimilarity or complete pyramidal dissimilarity data) from which the simulation data sets were generated; (b) the percentage p of missing values removed from the complete data (p = 0%, 10%, 20%, 30%, 40%); (c) the variance σ^2 of random error added to the non-missing data ($\sigma^2 = 0.0,\ 0.25,\ 0.5$); (d) the data analysis technique used (HACII, HLSC, PACII, PLSC). Note that prior to embarking on this design, the complete data were normalized to unit variance.

In the following example the set of objects to be clustered consisted of $n = 10$ elements for each underlying type of data. For each combination of p and σ^2 twenty dissimilarity data sets were randomly generated for each underlying model and evaluated by each of the techniques HACII, HLSC, PACII, and PLSC. Corresponding goodness-of-fit values and their means were calculated. Note that computational efforts for these calculations increase from HACII to PACII to HLSC to PLSC, i.e., PLSC is the most complex algorithm.

Tables 4a and 4b summarize the results for two different types of fit measures. First, the product moment correlation between the non-missing δ_{ij} and the corresponding d_{ij} values — denoted by ''modified product moment correlation'' (mpmc for short) is given. These results show how the algorithms perform with respect to the available part of the underlying data. Second, in brackets, the product moment correlation (pmc for short) between the complete, error-free δ_{ij} — before randomly labeling missing entries — and the complete d_{ij} values is displayed. With respect to the mpmc values it should be mentioned that the task of fitting the non-missing part of predefined data becomes easier when the percentage of missing values increases, and that, therefore, mpmc approaches to the value 1 for increasing p. This effect can be compensated by introducing suitable corrections. However, in Tables 4a, b the uncorrected fit values of mpmc are shown to avoid subjective adjustments. The pmc values are added in brackets to allow comparisons with results published in De Soete (1984). In Appendix D an example is described which demonstrates that different and, perhaps, misleading judgments could result if mpmc is not used.

A last point to be mentioned is that dependent on the pattern and percentage of missing values HACII and PACII might stop with an output consisting of two or more disjoint subhierarchies or subpyramids, a problem that cannot occur if HLSC or PLSC is used. In our study — where the percentage of missing values was restricted to 0%, 10%, 20%, 30%, and 40% — neither HACII nor PACII applications resulted in subhierarchy or subpyramid solutions. Thus, all outcomes are comparable in this sense.

In the following, we mainly comment those values of Tables 4a and 4b which are based on the mpmc calculations. Considering some of the results, we note that if the method used and the percentage p of missing values are fixed, an increase of the variance σ^2 of random error — which is one indicator for the degree of deviation from the underlying hierarchical or pyramidal structure — leads to a decrease in mpmc independent of the underlying type of data used in this study. This tendency also holds for pmc. Independent of the combination of p and σ^2, the difference in fit between the hierarchical algorithms HACII and HLSC, respectively, is always marginal. If, additionally, the output of HACII would have been used as input for HLSC, further improvements of HLSC's fit may be possible.

The most complex algorithm, PLSC, always shows the best results in terms of mpmc values.

If $\sigma^2 = 0$ and the underlying type of data are hierarchical dissimilarities, all methods attain maximal fit with respect to mpmc values, i.e., in this study the available part of the underlying hierarchical structure is recovered

	$\sigma^2 = 0$				
	p=0%	p=10%	p=20%	p=30%	p=40%
HACII	1.0	1.0 (.99)	1.0 (.98)	1.0 (.92)	1.0 (.85)
HLSC	1.0	1.0 (.99)	1.0 (1.0)	1.0 (.91)	1.0 (.86)
PACII	1.0	1.0 (.99)	1.0 (.98)	1.0 (.95)	1.0 (.86)
PLSC	1.0	1.0 (.99)	1.0 (.98)	1.0 (.93)	1.0 (.84)
	$\sigma^2 = 0.25$				
HACII	.95	.91(.94)	.92(.90)	.92(.86)	.92(.84)
HLSC	.95	.91(.94)	.92(.87)	.92(.77)	.92(.69)
PACII	.93	.92(.81)	.93(.75)	.94(.69)	.93(.60)
PLSC	.96	.96(.84)	.97(.75)	.98(.71)	.98(.62)
	$\sigma^2 = 0.5$				
HACII	.89	.85(.86)	.86(.78)	.87(.63)	.87(.60)
HLSC	.90	.85.(.86)	.86(.79)	.88(.66)	.86(.60)
PACII	.88	.88(.71)	.88(.66)	.88(.55)	.91(.51)
PLSC	.94	.94(.73)	.95(.69)	.96(.59)	.96(.52)

Table 4a: Results of Monte Carlo Evaluation Starting from Known Hierarchical Dissimilarity Data: Mpmc and Pmc (in Brackets) Values of Fit.

	$\sigma^2 = 0$				
	p=0%	p=10%	p=20%	p=30%	p=40%
HACII	.93	.93(.85)	.94(.84)	.94(.83)	.94(.76)
HLSC	.93	.93(.85)	.94(.84)	.94(.83)	.94(.76)
PACII	1.0	.99(.96)	.99(.95)	.99(.92)	.99(.85)
PLSC	1.0	1.0 (.98)	1.0 (.96)	1.0 (.94)	1.0 (.84)
	$\sigma^2 = 0.25$				
HACII	.86	.86(.78)	.87(.75)	.88(.66)	.89(.63)
HLSC	.86	.86(.76)	.87(.75)	.88(.65)	.88(.61)
PACII	.91	.91(.80)	.92(.78)	.92(.69)	.91(.59)
PLSC	.96	.96(.83)	.97(.80)	.98(.71)	.97(.61)
	$\sigma^2 = 0.5$				
HACII	.80	.80(.71)	.83(.70)	.84(.59)	.84(.55)
HLSC	.81	.80(.68)	.84(.70)	.83(.58)	.84(.52)
PACII	.85	.85(.65)	.87(.68)	.87(.56)	.87(.52)
PLSC	.93	.93(.70)	.96(.69)	.96(.69)	.96(.49)

Table 4b: Results of Monte Carlo Evaluation Starting from Known Pyramidal Dissimilarity Data: Mpmc and Pmc (in Brackets) Values of Fit.

regardless of the value of p. For $\sigma^2 > 0$ the available part of the original hierarchical dissimilarities does not fulfill the hierarchical constraints any longer, and a gap between the mpmc fit values computed for the hierarchical and the pyramidal solutions appears. This gap widens as σ^2 is increased. The mpmc fit values obtained from the pyramidal algorithms are always better than those from the hierarchical counterparts but it should be noted that conclusions with respect to the underlying hierarchy still have to be drawn from these pyramidal solutions (if one knows that the underlying dissimilarity data actually describe a hierarchy).

If the underlying type of data describes pyramidal dissimilarities for all combinations of p and σ^2, there is a remarkable difference in fit with respect to the solutions obtained by hierarchical versus pyramidal techniques. PACII yields better mpmc fit results than the hierarchical algorithms (which perform similarly as already stated above), and PLSC still exceeds PACII. Additionally, it should be recalled that the hierarchical solutions obtained by HACII and HLSC cannot fit underlying pyramidal dissimilarities. Thus, also for $\sigma^2 = 0$ and $p = 0\%$, maximal fit to the available part of the data can only be yielded by PACII and PLSC, as Table 4b shows.

In practice, nearly all empirical dissimilarity data are of course neither ultrametric nor pyramidal. Therefore, the random error pertubated data, where σ^2 indicates the degree of deviation from an underlying hierarchical or pyramidal structure, and the results of the methods applied to these data are mainly of interest. Here, the fit values of the first column of Tables 4a and 4b give hints as to how the algorithms "behave" on complete data. In this case mpmc and pmc are equal. However, empirical dissimilarity data may be incomplete. The remaining columns of Tables 4a and 4b give first impressions of the behavior of the algorithms for different percentages of missing values in the underlying data. Although the question which measure of fit would be appropriate when missing values occur (see also Appendix D) may need further discussion, the Monte Carlo results are of interest per se and give first insights with respect to the ability of the pyramidal approach to cope with the data situation discussed.

4. Conclusions

In this paper, the PACII and PLSC algorithms for pyramidal classification based on incomplete dissimilarity data have been proposed. Their effectiveness has been evaluated comparing them to each other as well as to our implementations of their hierarchical counterparts in a Monte Carlo study. The ability of the pyramidal approach to process such parts of the available data which hierarchical classification cannot take into account has been demonstrated, especially in the case of incomplete dissimilarity data which are neither hierarchical nor pyramidal.

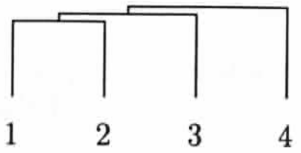

Figure 3a. Chaining within a Dendrogram.

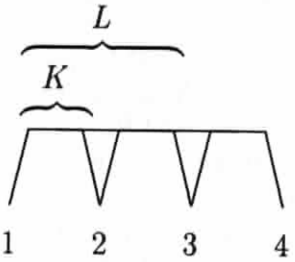

Figure 3b. Chaining on the Same Index Level within a Pyramid.

Appendix A

"Chaining" is a well-known property of clustering algorithms creating dendrograms of the form shown in Figure 3a, a situation that can also happen when pyramids are determined. Chaining on the same index level would occur if tied index values are assigned to the nested clusters. Condition (7b) prevents chaining of clusters on the same index level because if a situation as depicted in Figure 3b occurs, one could choose sets K and L as indicated in Figure 3b but not sets $J_1 \neq K$ and $J_2 \neq K$ which fulfill $K = J_1 \cap J_2$.

Appendix B

The Boolean-valued function *linkable* used in Section 2.1 indicates whether two clusters K and L can be linked together without destroying the pyramidal structure of the current set P: *linkable* $(K,L) := K \cup L \notin P$ and $N(K) < 2$ and $N(L) < 2$ and *border* (K) and *border* (L) and (*rightNeighbor* (K,L) or *rightNeighbor* (L,K) or $([K] \neq [L]$ and $(\min K = \min [K]$ or $\max K = \max [K])$ and $(\min L = \min[L]$ or $\max L = \max [L])))$,
order $(K) := \nexists J \in P{:}K \subset J$ and $\min J < \min K < \max K < \max J$,
rightNeighbor (K,L):=($\min L < \min K$ and $\max L < \max K$) and ($\min K = \min \{\min J{:}J \in P$ and $\min L < \min J$ and $\max L < \max J\}$) and ($\max L = \max \{\max J{:}J \in P$ and $\min J < \min K$ and $\max J < \max K\}$).

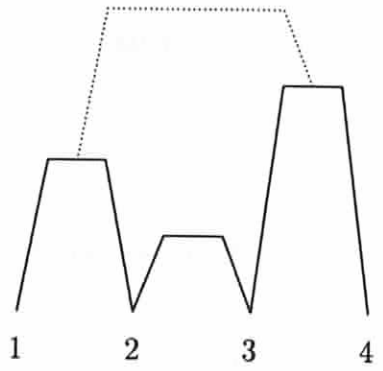

Figure 4a

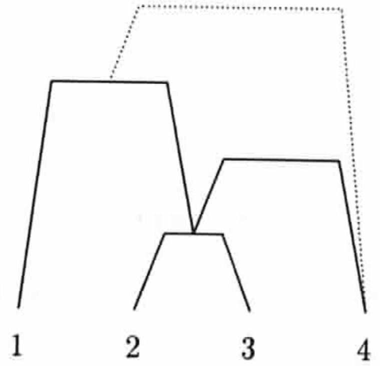

Figure 4b

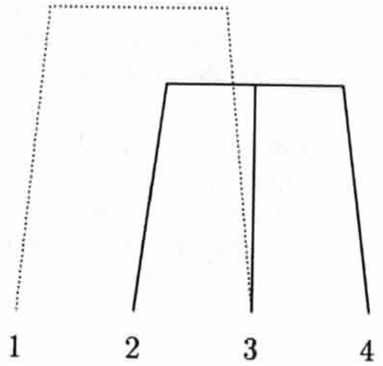

Figure 4c

Three examples with four objects, where the links indicated by the dotted lines (if considered) are prevented by a ''false'' value of *linkable*, are shown below in Figures 4a-4c.

Appendix C

The DD (Doubles Décalages) method used in Section 2.2 successively to generate new total orders from a given total order on $I = \{1, \ldots, n\}$ can be described as follows:

```
(1)  Select a maximal stepsize $m \in \{1, \ldots, n-1\}$. Begin with a total
     order $i_1 < i_2 < \cdots < i_n$ on $I$.
(2)  For $j = 1, 2, \ldots, n-1$:
     (2a) (''forward'' décalage) For $k = j+1,\ j+2, \ldots,$ min
          $\{j+m, n\}$: Swap $i_j$ and $i_k$ to obtain a new total order, then
          re-swap $i_j$ and $i_k$.
     (2b) (''backward'' décalage) For $k = j-1,\ j-2, \ldots,$ max
          $\{j-m, 1\}$: Swap $i_j$ and $i_k$ to obtain a new total order, then
          re-swap $i_j$ and $i_k$.
```

Appendix D

If incomplete dissimilarity data have to be used as input for algorithms designed to tackle the missing values problem one has also to consider which criteria for the comparison of the solutions obtained one should select. In the Monte Carlo study described in this paper mpmc (modified product moment correlation) was used as a fit measure operating on the non-missing part of the data. For complete data mpmc is equal to pmc (product moment correlation) and in a simulation situation where the ''true'' underlying, i.e., the complete data are known one could argue that the fit should be determined with respect to the ''true'' data structure.

The following example describes a simulation experiment where the ''true'' data structures change in the following manner: Take the dissimilarities δ_{ij} of Table 5a as starting point. Here, 40% of the data are missing. The corresponding entries are denoted by x. Assume that different ''true'' data structures can be described by successively replacing all missing values by $x = 0, 1, \ldots, 20$. The non-missing part remains the same regardless of the value assigned to x.

With the input described in Table 5a, applications of, e.g., HACII and PACII result in the solutions depicted in Tables 5b und 5c. The mpmc fit values are 1.0 for PACII and 0.9408 for HACII (regardless of the ''true'' data structure), while the pmc fit values are shown in Figure 5.

	1	2	3	4	5
1	0				
2	1	0			
3	5	3	0		
4	8	x	x	0	
5	x	x	6	2	0

Table 5a: Example Dissimilarity Data with Missing Values

	1	2	3	4	5
1	0				
2	1	0			
3	5	3	0		
4	8	6	6	0	
5	6	6	6	2	0

Table 5b: PACII Result for the Data given in Table 5a

	1	2	3	4	5
1	0				
2	1	0			
3	4	4	0		
4	7	7	7	0	
5	7	7	7	2	0

Table 5c: HACII Result for the Data given in Table 5a

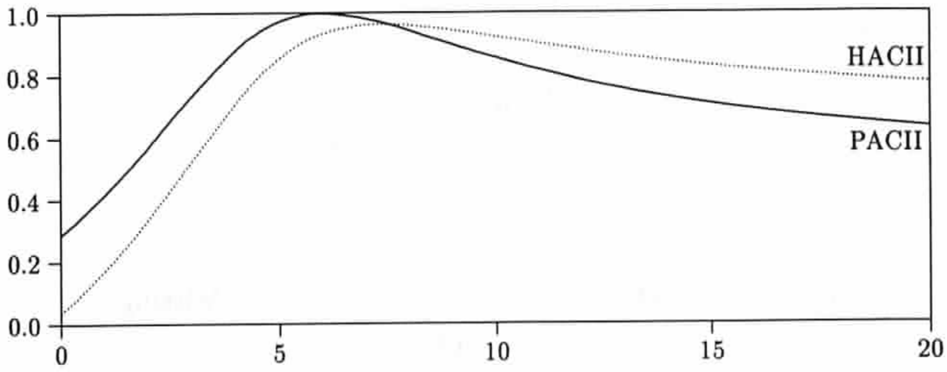

Figure 5. Pmc Values for different values of x.

Here, totally different judgments about the performance of both algorithms would result dependent on the "true" data structure which (normally) neither the user nor the algorithms "know". Sometimes the results of both algorithms would be judged as acceptable, sometimes the results of both algorithms would have to be rejected, sometimes PACII would be estimated better than HACII but sometimes worse.

References

AMBROSI, K. (1978), "Klassifikation und Identifikation," in *Numerische Taxonomie in der Marktforschung*, Ed., O. Opitz, München: Vahlen, 79-109.

ARABIE, P., and CARROLL, J. D. (1980), "MAPCLUS: A Mathematical Approach to Fitting the ADCLUS Model," *Psychometrika, 45*, 211-235.

CARROLL, J. D., and ARABIE, P. (1983), "INDCLUS: An Individual Differences Generalization of the ADCLUS Model and the MAPCLUS Algorithm," *Psychometrika, 48*, 157-169.

DE SOETE, G. (1984), "Ultrametric Tree Representations of Incomplete Dissimilarity Data," *Journal of Classification, 1*, 235-242.

DIDAY, E. (1986), "Orders and Overlapping Clusters by Pyramids," in *Multidimensional Data Analysis*, Eds., J. de Leeuw, W. Heiser, J. Meulman, and F. Critchley, Leiden: DSWO, 201-234.

DIDAY, E. (1987), "Orders and Overlapping Clusters by Pyramids," Rapports de Recherche No. 730, Octobre 1987, INRIA, Paris.

DIDAY, E., and BERTRAND, P. (1986), "An Extension of Hierarchical Clustering: The Pyramidal Presentation," in *Pattern Recognition in Practice II*, Eds., E.S. Gelsema and L.N. Kanal, Amsterdam: North-Holland, 411-424.

JARDINE, C. J., JARDINE, N., and SIBSON, R. (1967), "The Structure and Construction of Taxonomic Hierarchies," *Mathematical Biosciences, 1*, 173-179.

JOHNSON, S. C. (1967), "Hierarchical Clustering Schemes," *Psychometrika, 32*, 241-254.

KRUSKAL, J. B., LANDWEHR, J. M., and McKAE, J. E. (1985), ''ICICLE Plot Package for Hierarchical Clustering,'' *Journal of Classification*, 2, 131-132.

LANCE, G. N., and WILLIAMS, W. T. (1967), ''A general Theory of Classificatory Sorting Strategies. 1. Hierarchical systems,'' *Computer Journal*, 9, 337-380.

MARCOTORCHINO, J.-F., and MICHAUD, P. (1979), *Optimisation en Analyse Ordinale des Données*, Paris: Masson.

OPITZ, O. (1980), *Numerische Taxonomie*, Stuttgart: Fischer.

POWELL, M. J. D. (1977), ''Restart Procedures for the Conjugate Gradient Method,'' *Mathematical Programming*, 12, 241-254.

SCHADER, M., and GAUL, W. (1992), ''The MVL (Missing Values Linkage) Approach for Hierarchical Classification when Data are Incomplete,'' in *Analyzing and Modeling Data and Knowledge*, Ed., M. Schader, Berlin-Heidelberg: Springer, 107-115.