Journal of Classification 18:57-80 (2001)
DOI: 10.1007/s00357-001-0005-2

An Evaluation of Two Algorithms for Hierarchical Classes Analysis

Iwin Leenen

K.U. Leuven, Belgium

Iven Van Mechelen

K.U. Leuven, Belgium

Abstract: The family of hierarchical classes models is a distinctive collection of direct two-sided clustering models for two-way two-mode binary data. In the associated data analysis, a data matrix **D** is approximated by a binary matrix **M** that can be represented by a hierarchical classes model of a prespecified rank k. In this procedure, a least-squares loss function in terms of discrepancies between **D** and **M** is minimized. The present paper describes the original hierarchical classes algorithm proposed by De Boeck and Rosenberg (1988), which is based on an alternating greedy heuristic, and proposes a new algorithm, based on an alternating branch-and-bound procedure. An extensive simulation study is reported in which both algorithms are evaluated and compared according to goodness-of-fit to the data and goodness-of-recovery of the underlying true structure. Furthermore, three heuristics for selecting models of different ranks for a given **D** are presented and compared. The simulation results show that the new algorithm yields models with slightly higher goodness-of-fit and goodness-of-recovery values.

Keywords: Hierarchical classification; Binary data; Combinatorial data analysis; Branch-and-bound algorithm; Simulation

---

The research reported in this paper was partially supported by the Research Council of K.U.Leuven (PDM/99/037 and GOA/2000/02) and NATO (Grant CRG.921321 to Iven Van Mechelen and Seymour Rosenberg).

We are grateful to Dean McKenzie, Seymour Rosenberg, and Paul De Boeck for their interesting suggestions and helpful comments on an earlier draft of this paper.

Authors' Address: Iwin Leenen and Iven Van Mechelen, Department of Psychology, University of Leuven, Tiensestraat 102, B-3000 Leuven, Belgium; email: iwin.leenen@psy.kuleuven.ac.be and iven.vanmechelen@psy.kuleuven.ac.be.

The family of hierarchical classes models as developed by De Boeck, Rosenberg, and Van Mechelen (De Boeck and Rosenberg 1988; Van Mechelen, De Boeck, and Rosenberg 1995) is a distinctive collection of two-sided clustering models for binary two-way two-mode data. The hierarchical classes approach has been successfully applied in various substantive contexts. Summaries of the approach with several empirical applications can be found in De Boeck, Rosenberg, and Van Mechelen (1993) and Rosenberg, Van Mechelen, and De Boeck (1996).

De Boeck and Rosenberg (1988; De Boeck 1986) described an algorithm for fitting the hierarchical classes model to a data set. Those authors concluded from a simulation study that their algorithm succeeds well in recovering the true structure underlying a data matrix. This paper proposes a new HICLAS algorithm and reports the results of a more extensive simulation in which the original and the new algorithm are evaluated and compared.

Section 1 recapitulates the original hierarchical classes model as proposed by De Boeck and Rosenberg (1988), and Section 2 describes the associated original algorithm. Section 3 presents the new algorithm and Section 4 the simulation study. Section 5 contains some concluding remarks.

1. The Disjunctive Hierarchical Classes Model

The family of hierarchical classes models comprises a disjunctive and a conjunctive type, each having four variants (Van Mechelen et al. 1995). At this point we will focus on the original disjunctive model only (De Boeck and Rosenberg 1988).

Hierarchical classes analysis approximates an $h \times n$ data matrix $\mathbf{D}$ with an $h \times n$ binary matrix $\mathbf{M}$ that can be represented by a hierarchical classes model (of a prespecified complexity). As a guiding example in this section we will make use of the matrix $\mathbf{M}$ presented in Table 1; a graphical representation of a hierarchical classes model for this matrix is given in Figure 1, and the algebraic form of the same model is given in Table 2.

A hierarchical classes model of $\mathbf{M}$ implies a representation of three structural aspects of $\mathbf{M}$:

(1) The first aspect is a clustering (viz., a partition) of both the rows and the columns, that is naturally defined by $\mathbf{M}$. In particular, the row clusters comprise rows associated with identical sets of columns. Analogously, column clusters comprise columns that are associated with the same set of rows. For example, rows o_1, o_2, and o_5 in Table 1 are associated with the same columns, a_1, a_2, a_4, a_5, and a_6, and, hence, they constitute a row cluster. In the graphical representation, the row (respectively column) clusters appear as boxes containing the row and column labels. For example, the cluster $\{o_1, o_2, o_5\}$ shows up as one of the boxes in Figure 1.

Table 1: *Hypothetical two-way two-mode array*

	Columns					
Rows	a_1	a_2	a_3	a_4	a_5	a_6
o_1	1	1	0	1	1	1
o_2	1	1	0	1	1	1
o_3	1	0	0	1	0	1
o_4	1	0	0	1	0	1
o_5	1	1	0	1	1	1
o_6	0	1	0	1	1	0
o_7	0	0	0	0	0	0

Table 2: *Disjunctive hierarchical classes model for the data in Table 1*

	Row Bundles			Column Bundles	
Rows	I	II	Columns	I	II
o_1	1	1	a_1	1	0
o_2	1	1	a_2	0	1
o_3	1	0	a_3	0	0
o_4	1	0	a_4	1	1
o_5	1	1	a_5	0	1
o_6	0	1	a_6	1	0
o_7	0	0			

(2) The second aspect is two structural relations that may be defined on the clusterings of the rows and of the columns, respectively. The specific relations are partial orders and can be considered hierarchies (Van Mechelen, Rosenberg, and De Boeck 1997). The partial order on the row clusters is defined according to the subset-superset relations among their associated sets of columns. Analogously, the partial order on the column clusters is defined according to the subset-superset relations among their associated sets of rows. For example, the column cluster $\{a_2, a_5\}$ is hierarchically below the column cluster $\{a_4\}$ because the set of rows associated with the former, $\{o_1, o_2, o_5, o_6\}$, is a subset of the set of rows associated with the latter, $\{o_1, o_2, o_3, o_4, o_5, o_6\}$. In the graphical representation, the partial orders are represented by Hasse-diagrams. In Figure 1, the upper half represents the row hierarchy and the lower half the column hierarchy. (For reasons that will soon become clear, the Hasse-diagram of the column hierarchy is depicted inverted.)

(3) The third structural aspect comprises the association between the two hierachically organized clusterings. This association reflects the set of 1-entries in the matrix **M**. For example, from Table 1, one may derive that the row clus-

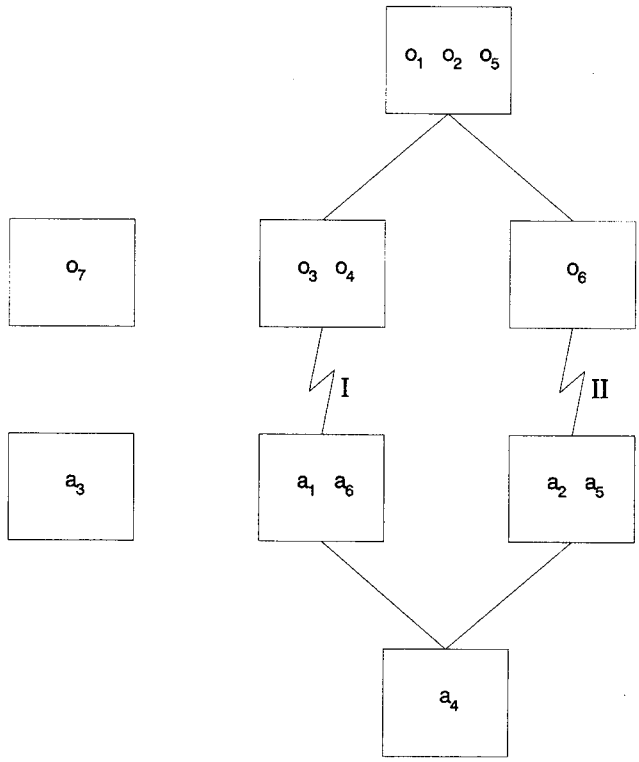


Figure 1. Graphical representation of the disjunctive hierarchical classes model of the matrix in Table 1.

ter $\{o_3, o_4\}$ is associated with the column cluster $\{a_1, a_4, a_6\}$. In the graphical representation, the association relation is represented by a set of zigzags between the hierarchies of row and column clusters. A row cluster is associated with a column cluster if and only if the row cluster and the column cluster are connected by a downward path of links and a zigzag.

Hierarchical classes models can be algebraically formalized as a decomposition of the $h \times n$ binary model matrix **M** into an $h \times k$ binary matrix **S** and an $n \times k$ binary matrix **P**, with k being the Schein rank of **M** (Kim 1982, pp. 36–37). **S** (resp. **P**) is called the row (resp. column) bundle matrix and the k columns of **S** (resp. **P**) are called row (resp. column) bundles. The set of bundles for which $s_{ir} = 1$ (resp. $p_{jr} = 1$) ($r = 1, \ldots, k$) is called the bundle pattern of row i (resp. column j). The bundle matrices **S** and **P** of a hierarchical classes model represent the three structural aspects of **M** mentioned above: (1) Rows (resp. columns) that belong to the same cluster have identical bundle patterns and vice versa. For example, columns of the column cluster $\{a_1, a_6\}$

have the same bundle pattern, {I}, in Table 2; (2) Two row (resp. column) clusters are hierarchically related if and only if the associated bundle patterns are in a subset-superset relation. For example, cluster $\{a_2, a_5\}$ is hierarchically below cluster $\{a_4\}$, and, hence, the bundle pattern {II} of the former is a subset of the bundle pattern {I, II} of the latter; (3) A row cluster is associated with a column cluster if and only if the intersection of their bundle patterns is nonempty. The latter implies that **M** can be fully reconstructed from **S** and **P**:

$$\mathbf{M} = \mathbf{S} \otimes \mathbf{P}', \tag{1}$$

with $\mathbf{P}'$ denoting the transpose of **P** and $\otimes$ denoting the Boolean matrix product (Kim 1982, p. 8). For example, $\{o_3, o_4\}$ is associated with $\{a_4\}$ as their bundle patterns have bundle I in common.

From the above, four distinctive features of the hierarchical classes approach should be clear: (a) Hierarchical classes models are *direct* clustering models in that they directly represent the binary two-way two-mode data rather than derived two-way one-mode proximities; (b) The models imply a two-sided clustering, that is, a simultaneous clustering of rows and columns, similar to such approaches as nodal analysis (Lambert and Williams 1962), two-way splitting/joining (Hartigan 1975, pp. 260–298) and blockmodeling (Arabie, Boorman, and Levitt 1978); (c) The models represent hierarchical relations between row and column clusters. Note that the hierarchy of row (resp. column) clusters is defined using set inclusion (as usual), but using associated sets of columns (resp. rows). Note further that the concept of hierarchy is formalized here using a partial order, consistent with, for instance, the general definition proposed by Sneath and Sokal (1973, pp. 205–207) rather than with the more restrictive use of the term in hierarchical clustering (for an indepth discussion of this topic, see Van Mechelen, Rosenberg, and De Boeck 1997). (d) The hierarchical classes models are based on a (Boolean) decomposition of a reconstructed data matrix (or model matrix) **M**. Note that this decomposition is a common feature of the hierarchical classes and Boolean factor analysis models (Mickey, Mundle, and Engelman 1983), but the latter models do not include a representation of the hierarchies of row and column clusters in **M**, which hierarchical classes models do. As shown by De Boeck and Rosenberg (1988) the latter representation implies a restriction on the model for Boolean factor analysis: Only a subset of the decompositions meeting Equation (1) simultaneously represents the clustering and the hierarchical relations. Otherwise, De Boeck and Rosenberg (1988) showed that if a decomposition of **M** into **S** and **P** consistent with (1) exists then $\mathbf{S}^*$ and $\mathbf{P}^*$ matrices of the same size always exist, that also combine to **M** by (1) and that, in addition, represent the partially ordered clusterings.

2. De Boeck and Rosenberg's HICLAS Algorithm

Any data matrix **D** can be perfectly represented by a hierarchical classes model if the rank of the model is sufficiently high (Kim 1982, p. 38; Leenen, Van Mechelen, and De Boeck 1999). In practice, however, for reasons of parsimony, low rank models are sought. Within this context, the algorithm proposed by De Boeck and Rosenberg (1988; De Boeck 1986) for fitting a rank k hierarchical classes model to a binary data matrix **D**, seeks matrices **S** and **P**, which by Equation (1) yield a matrix **M** such that the loss function

$$f(\mathbf{S},\mathbf{P}) = \sum_{i=1}^{h}\sum_{j=1}^{n}(d_{ij} - m_{ij})^2 \tag{2}$$

has minimal value, subject to the constraint that **S** and **P** represent the partially ordered clusterings of rows and columns as implied by **M**. Because **D** and **M** are binary, (2) can be considered both a least-squares and a least-absolute-deviations loss function (or, more generally, a least L_p-norm ($p > 0$) loss function). We note that a least absolute deviations view on the loss function may be particularly attractive, because it could be given a maximum likelihood justification within the context of a minimal stochastic extension of the model.

De Boeck and Rosenberg's algorithm successively executes two main routines (see Figure 2): The first routine, which is an improved version of algorithms for Boolean factor analysis (De Boeck and Rosenberg 1988), seeks matrices **S** and **P** such that (2) is minimal; in the second routine, **S** and **P** are modified so as to represent the respective partially ordered clusterings implied by **M**. The latter routine relies on the abovementioned proposition in De Boeck and Rosenberg (1988) that any pair of **S** and **P** matrices which by (1) yield a matrix **M** can be modified to represent the partially ordered clusterings. Otherwise, we note that this proposition implies that the representation of the hierarchical classicifations is an aspect of the model that does not affect the model's loss function. Hence, the second routine needs no further consideration.

The first routine of the De Boeck and Rosenberg algorithm can be considered an alternating greedy procedure, which starts by constructing an initial configuration for one of the bundle matrices. Assuming, without loss of generality, that an initial configuration, $\mathbf{P}^{(0)}$, is obtained for the column bundle matrix, the procedure looks, conditionally upon $\mathbf{P}^{(0)}$, for the optimal row bundle matrix $\mathbf{S}^{(0)}$, which is such that (2) is minimal. In the next steps, $\mathbf{P}^{(w)}$ is reestimated conditionally upon $\mathbf{S}^{(w-1)}$, and $\mathbf{S}^{(w)}$ is reestimated conditionally upon $\mathbf{P}^{(w)}$ ($w = 1, 2, \ldots$). This alternating procedure continues until no further improvement in the loss function (2) is observed.

An initial configuration can be obtained either (pseudo-)randomly, rationally by a built-in heuristic in the algorithm, or from the user. In a random

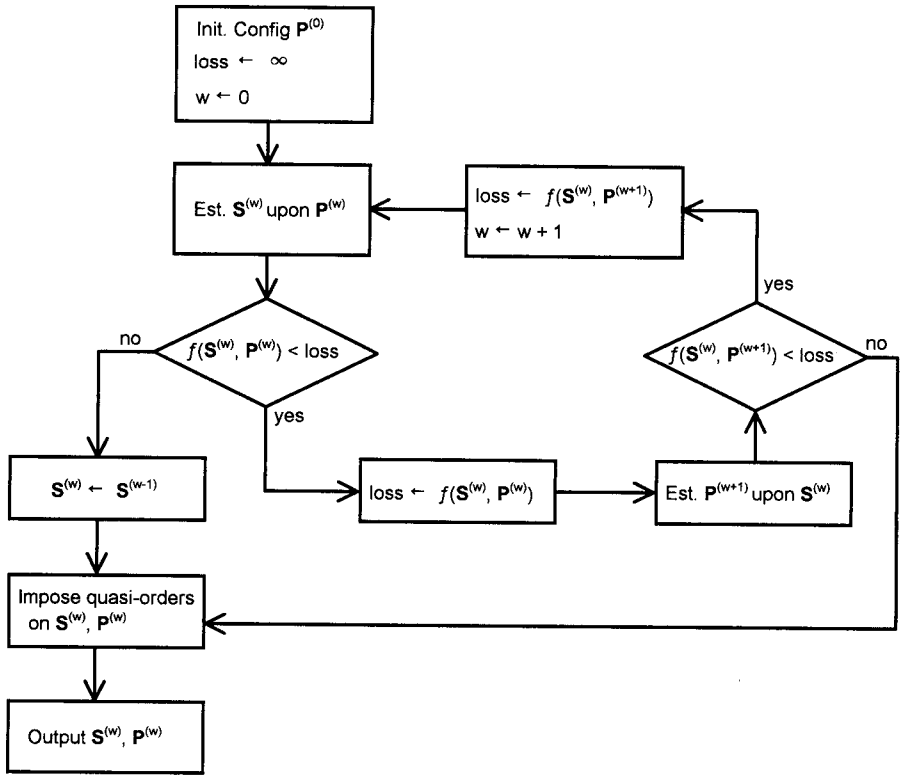


Figure 2. Flow chart of a hierarchical classes algorithm.

initial configuration for the row (resp. column) bundle matrix $\mathbf{S}^{(0)}$ (resp. $\mathbf{P}^{(0)}$), the values are independent realizations of a Bernoulli variable (with some parameter π). The rational heuristic for obtaining an initial rank k row (resp. column) bundle matrix first constructs n (resp. h) candidates, $\mathbf{S}_1^{(0)}, \ldots, \mathbf{S}_n^{(0)}$ (resp. $\mathbf{P}_1^{(0)}, \ldots, \mathbf{P}_h^{(0)}$), where the first $k-1$ bundles of $\mathbf{S}_j^{(0)}$ $(j = 1, \ldots, n)$ (resp. $\mathbf{P}_i^{(0)}$ $(i = 1, \ldots, h)$) are the bundles of the row (resp. column) bundle matrix of the solution in the previous rank $k-1$, and the k-th bundle is the data vector $\mathbf{m}_{\cdot j}$ (resp. $\mathbf{m}_{i\cdot}$). Next, the candidate $\mathbf{S}_j^{(0)}$ (resp. $\mathbf{P}_i^{(0)}$) is selected for which $f(\mathbf{S}_j^{(0)}, \mathbf{P}_j^{(0)})$ (resp. $f(\mathbf{S}_i^{(0)}, \mathbf{P}_i^{(0)})$) has minimal value, $\mathbf{P}_j^{(0)}$ (resp. $\mathbf{S}_i^{(0)}$) being the conditional estimate for the column (resp. row) bundle matrix given $\mathbf{S}_j^{(0)}$ (resp. $\mathbf{P}_i^{(0)}$).

A conditional estimate for a bundle matrix is obtained by a greedy heuristic which successively estimates each row (i.e., bundle pattern) of the matrix. The heuristic starts from the zero row vector and replaces in each of a num-

ber of steps a zero-entry by a one-entry, selecting that zero-entry for which the modification yields the largest gain in the loss function. The procedure stops if modifying any zero-entry no longer improves the loss function. The same greedy heuristic is also used by Van Mechelen (1988).

3. A Branch-and-bound HICLAS Algorithm

Greedy heuristics are known to yield efficient but possibly suboptimal solutions. In this section, we will therefore propose a new HICLAS algorithm in which the greedy heuristic is replaced by a procedure that necessarily yields optimal conditional estimates of the bundle matrices in each step of the algorithm.

We start from an analysis of the loss function (2). This function appears to be separable (Carroll and Chaturvedi 1995): From the Boolean product in (1), it is clear that the bundle pattern of row i, $\mathbf{s}_{i.}$, only affects $\mathbf{m}_{i.}$ and, vice versa, $\mathbf{m}_{i.}$ only depends on $\mathbf{s}_{i.}$ (given $\mathbf{P}$). Analogously, the bundle pattern $\mathbf{p}_{j.}$ of column j only affects $\mathbf{m}_{.j}$, and vice versa. Hence, the contribution of the bundle pattern of row i (resp. column j) to the loss function can be separated from the contribution of the bundle patterns of other rows (resp. columns). In particular, for the row bundle matrix $\mathbf{S}$,

$$f(\mathbf{S}, \mathbf{P}) = \sum_{i=1}^{h} f(\mathbf{s}_{i.}, \mathbf{P}),$$

with $f(\mathbf{s}_{i.}, \mathbf{P}) = \sum_{j=1}^{n} (d_{ij} - \mathbf{s}_{i.} \otimes \mathbf{p}'_{j.})^2$ being the contribution of bundle pattern $\mathbf{s}_{i.}$ to the loss function. As a result, a conditional optimal estimate of $\mathbf{S}$ can be found by successively estimating the optimal value for $\mathbf{s}_{i.}$ $(i = 1, \ldots, h)$ in each of the h terms of the sum on the right.

An optimal conditional estimate for $\mathbf{s}_{i.}$ (resp. $\mathbf{p}_{j.}$) can be found by the branch-and-bound algorithm for Boolean regression proposed by Leenen and Van Mechelen (1998). In general, this branch-and-bound algorithm finds, given a binary criterion and a set of binary predictors, the subset of predictors whose disjunctive combination optimally predicts the binary criterion (with optimality defined according to a least-squares loss function). For the implementation of the algorithm within a HICLAS context, the k bundles of the column (row) bundle matrix are considered as predictors and row i (column j) of the data matrix as the criterion.

In particular, the Boolean regression algorithm considers a tree with 2^k nodes corresponding to the 2^k possible bundle patterns for row i. This tree has $k + 1$ levels, labeled 0 through k, with level v $(v = 0, \ldots, k)$ having all bundle patterns with v elements. The children of the root node at level 0 are the k singletons, and the children of a node W at level v $(v = 1, \ldots, k)$ are the sets $W \cup C^{(l)}$, for all $C^{(l)}$ that are a sibling (i.e., a child of the parent) of W

and that succeed W in some predefined order on the set of W and its siblings. The algorithm passes through the tree by a recursive procedure which makes extensive use of the bounding function

$$f^{+}(\mathbf{s}_{i.}, \mathbf{P}) = \sum_{j|d_{ij}=0} (d_{ij} - \mathbf{s}_{i.} \otimes \mathbf{p}'_{j.})^2.$$

The function f^{+} returns the number of false positives associated with row i and gives a lower bound on f for all possible extensions (i.e., the child nodes) of $\mathbf{s}_{i.}$, because false positives cannot be corrected by adding bundles to the bundle pattern.

Given the separability of the loss function and the use of the branch-and-bound procedure, the HICLAS algorithm is turned into an alternating least-squares procedure, which in a binary framework is equivalent with an alternating least-absolute-deviations algorithm (Chaturvedi and Carroll 1994, 1997). It must be noted that, although in each step of this algorithm an optimal conditional estimate of a bundle matrix is obtained, the procedure does not necessarily yield a global optimum for the overall optimization problem (viz., the pair of bundle matrices with lowest value on (2)).

4. Simulation Study

The aim of the present simulation study is twofold:

First, we want to evaluate and compare the original greedy algorithm and the new branch-and-bound algorithm. The greedy algorithm has been evaluated by De Boeck and Rosenberg (1988) in a rather small simulation study. In the present study (a) a greater range in the size of the matrices is included, (b) data matrices are constructed from different true matrices, rather than from the same true matrix, (c) the rank of the true matrices is varied, (d) within each cell of the design more data matrices are analyzed, and (e) not only the goodness-of-recovery of the association relation will be reported on, but also the goodness-of-recovery of the clusterings and of the partial orders.

Second, some heuristics for selecting a model rank will be presented and evaluated. In practice, models of increasing rank are typically fitted to a given data set and then a model is selected on the basis of some heuristic. Three such heuristics will be presented and compared.

This section is divided into four subsections: In the first subsection, the design and the procedure of the simulation study are discussed. The second and third subsections evaluate and compare both algorithms with respect to goodness-of-fit and goodness-of-recovery, respectively. Subsection 4 discusses the heuristics for rank selection.

4.1 Design and Procedure

Three different types of $h \times n$ matrices are to be distinguished in this simulation study: the true matrix **T**, for which a hierarchical classes model in some low rank q exists; the data matrix **D**, which is the matrix **T** perturbed with error; and the reconstructed model matrix **M** of rank k, yielded by the analysis of **D**.

A split-plot factorial design is used with five independent variables, three of which are between-block and two of which are within-block variables (Kirk 1982, 489–569). The three between-block variables are:

(1) the Size $h \times n$ of **T**, **D** and **M** having six levels: 15×25, 20×20, 80×20, 40×40, 500×50 and 150×150;

(2) the True rank q of the hierarchical classes model for **T** having three levels: 3, 5 and 8;

(3) the Error level e (or badness of data), which is the proportion of discrepancies between **D** and **T**, or,

$$e = \frac{\sum_{i=1}^{h} \sum_{j=1}^{n} (t_{ij} - d_{ij})^2}{h \times n}$$

having six levels: .00, .05, .10, .15, .20 and .25.

The two within-block variables are:

(4) the Analysis rank k of the matrix **M** having eight levels from 1 through 8;

(5) the Algorithm used to find **M** having two levels: greedy and branch-and-bound.

The expected proportion of one-entries in the matrices **T** and **D** was kept constant at the value 1/2 as follows: For each combination of Size $h \times n$, True rank q and Error level e, 25 pairs of an $h \times q$ matrix **S** and an $n \times q$ matrix **P** were generated with

$$s_{ir}, p_{jr} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\pi)$$

where $\pi = \sqrt{1 - \sqrt[3]{1/2}}$. Matrices T were calculated by:

$$\mathbf{T} = \mathbf{S} \otimes \mathbf{P}'.$$

Using the independence of s_{ir} and p_{jr}, $\Pr(t_{ij} = 1)$ can be shown to be equal to:

$$\Pr(t_{ij} = 1) \quad = \quad \Pr(\exists r : s_{ir} = 1 \wedge p_{jr} = 1)$$

$$
\begin{aligned}
&= 1 - \Pr(\forall r : s_{ir} = 0 \vee p_{jr} = 0) \\
&= 1 - \prod_{r=1}^{q} [1 - \Pr(s_{ir} = 1 \wedge p_{jr} = 1)] \\
&= 1 - (1 - \pi^2)^q,
\end{aligned}
$$

which equals 1/2 for the given value of π.

From each true matrix **T**, a data matrix **D** was constructed, with $d_{ij} = 1 - t_{ij}$ for a randomly selected set of cells, which contained a proportion e of the total number of cells, and $d_{ij} = t_{ij}$ otherwise. This strategy resulted in 25 $\times$ 6 (error levels) $\times$ 3 (true ranks) $\times$ 6 (sizes) = 2,700 different couples (**T**, **D**).

Each data matrix **D** was analyzed by both algorithms in ranks $k = 1, \ldots, 8$ using the rational heuristic for the initial configuration; as a consequence, for each **D** 2 (algorithms) $\times$ 8 (analysis ranks) = 16 model matrices **M** were obtained. Hence, there were 2,700 $\times$ 16 = 43,200 different triples (**T**, **D**, **M**).

4.2 Goodness-of-fit

In this section, the original and new HICLAS algorithm are evaluated with respect to goodness-of-fit, that is, with respect to how well they succeed in minimizing the loss function (2). To compare analyses of matrices of different size, the following badness-of-fit (BOF) statistic is used:

$$
BOF = \frac{\sum_{i=1}^{h} \sum_{j=1}^{n} (d_{ij} - m_{ij})^2}{h \times n} \tag{3}
$$

Only analyses in the true rank (i.e., $k = q$) will be considered; because, in that case, it is known that the error level e is an upper bound for the BOF-value of the best fitting model. Table 3 displays the mean differences between BOF and e for both algorithms at every level of the between-block variables. The variables True rank and Analysis rank, being equal, are renamed *Rank*.

In this and the following section, the two algorithms are first considered separately: An analysis of variance is conducted for each algorithm with the performance statistic under consideration as the dependent variable and the between-block variables as the independent variables. Next, the two algorithms are compared: The overall difference between them is tested with a pairwise t-test, and an analysis of variance is further used to check whether the difference is affected by the between-block variables and/or their interactions. Because of the high power of the analysis-of-variance tests, only effects that have an effect size, as measured by the intraclass correlation $\hat{\rho}_I$ (Haggard 1958, pp.44–72; Kirk 1982, pp. 162-163), above .10 will be considered. As pointed out by

Table 3: *Mean difference between Badness-of-fit and Error level for different levels of Size, Rank, Error level and Algorithm*

		Greedy algorithm							Branch-and-bound algorithm						
		Error Level							Error Level						
		.00	.05	.10	.15	.20	.25	Overall	.00	.05	.10	.15	.20	.25	Overall
Size	15 × 25	.012	.003	−.010	−.028	−.053	−.087	−.027	.008	.000	−.012	−.032	−.058	−.091	−.031
	20 × 20	.015	.006	−.006	−.023	−.048	−.080	−.023	.011	.002	−.010	−.026	−.054	−.085	−.027
	80 × 20	.013	.008	.005	−.002	−.011	−.030	−.003	.010	.006	.002	−.006	−.015	−.035	−.006
	40 × 40	.013	.010	.010	.007	−.004	−.020	.003	.010	.007	.006	.003	−.008	−.024	−.001
	500 × 50	.007	.003	.001	.002	.006	.009	.005	.005	.002	.000	.002	.004	.006	.003
	150 × 150	.011	.005	.000	.000	.001	.000	.003	.012	.004	−.000	.000	−.000	.001	.003
Rank	3	.020	.007	.002	.002	−.003	−.013	.002	.018	.005	.002	−.001	−.006	−.015	.001
	5	.008	.006	.003	−.005	−.016	−.034	−.006	.005	.002	−.001	−.007	−.020	−.037	−.010
	8	.008	.004	−.005	−.018	−.036	−.057	−.017	.005	.003	−.008	−.021	−.040	−.062	−.021
Overall		.012	.006	.000	−.007	−.018	−.035	−.007	.009	.004	−.002	−.010	−.022	−.038	−.010

Hays (1994, p. 534), $\hat{\rho}_I$ may be interpreted as the proportion of variance in the dependent variable accounted for by a particular factor in the model.

Concerning the separate evaluation of both algorithms, the mean value for $(BOF - e)$ is respectively -0.0071 and -0.0099, for the original greedy algorithm and the branch-and-bound version, indicating that the algorithms, on the average, succeed in finding a model that is at least as close to the data as the true model. The analyses of variance yield similar results for both algorithms: There is a sizeable effect of Error level ($\hat{\rho}_I$ = .22, .23 for the original and new algorithm, respectively), $(BOF - e)$ generally decreasing with increasing values of e. (In this context, it was further found that BOF increases with e.) The effect of Size ($\hat{\rho}_I$ = .12, .14) further needs to be considered, with large and square matrices yielding worse fits, BOF being close to e. Finally, there is an important interaction between Size and Error level ($\hat{\rho}_I$ = .23, .21): $(BOF - e)$ largely decreases with increasing e for small data sets, whereas e does not have a systematic effect for large data sets. This interaction effect becomes larger with increasing Rank as indicated by the important triple interaction ($\hat{\rho}_I$ = .11, .12).

Concerning the comparison of both algorithms, the BOF of the new algorithm is significantly smaller than the BOF of the original algorithm ($t(2699) = 18, p < .01$). The analysis of variance does not reveal sizeable effects of Size, Rank, or Error level on the difference between both algorithms.

4.3 Goodness-of-recovery

In this section, we evaluate and compare the algorithms with respect to how well the true hierarchical classes model underlying the data is recovered. Because the truth cannot be expected to be fully recovered by a model of rank k lower than the true rank q, again only analyses in the true rank ($k = q$) will be considered. As mentioned above, goodness-of-recovery can be considered with respect to (a) the clustering of rows and columns, (b) the hierarchies defined on the clusterings, and (c) the association relation as represented by **T**.

With respect to the goodness-of-recovery of the clusterings, the partition of the row (resp. column) set as represented by the underlying true model is to be compared with the partition of the row (resp. column) set as represented by the model obtained from the analysis. For this purpose, Hubert and Arabie's (1985) Rand index corrected for chance is used. For a given set S with n elements and two partitions U and V of S with classes $\{u_1, \ldots, u_i, \ldots, u_R\}$ and $\{v_1, \ldots, v_j, \ldots, v_C\}$, respectively, the original Rand Index (Rand 1971; Hubert

and Arabie 1985) is defined as

$$\frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{ij}}{2}}{\binom{n}{2}},$$

where n_{ij} is the number of elements in S that belong both to u_i and v_j [$\binom{n_{ij}}{2}$ is defined to be 0 for $n_{ij} < 2$]. To take into account chance correspondence, Hubert and Arabie proposed a corrected Rand Index using the general formula for chance correction

$$\frac{\text{Index} - \text{Expected Index}}{\text{Maximum Index} - \text{Expected Index}}.$$

As a result, the corrected Rand Index assumes a maximum value of 1 if both partitions are identical and a value of 0 if there is no more correspondence than expected by chance. The corrected Rand index was calculated for both the row and the column clustering and, subsequently, averaged (taking into account the number of rows and columns) to obtain a single corrected Rand index for each analysis. Table 4 displays the mean corrected Rand index for both algorithms at every level of the between-block variables.

Concerning the separate evaluation of both algorithms, the mean corrected Rand index across all analyses equals respectively .653 and .664 for the original greedy algorithm and the branch-and-bound method. The results of the analyses of variance are similar for both algorithms. The corrected Rand index is considerably affected by Size ($\hat{\rho}_I$ = .34, .32) with large and square matrices yielding near perfect Rand values. Also, Error level has a large effect ($\hat{\rho}_I$ = .31, .33): High error levels generally yield lower Rand values. The Size by Error level interaction effect ($\hat{\rho}_I$ = .11, .11) qualifies this main effect, though, as the effect of Error level is strong for smaller matrices only and almost disappears for larger and square matrices.

Concerning the comparison of both algorithms, the difference in Rand index between both algorithms is significant, $t(2699) = 5.4$, $p < .01$. This difference is almost not affected by Size, Rank, or Error level.

To evaluate the goodness-of-recovery of the row and column hierarchies, we will focus on the quasi-order relations on the rows and the columns implied by the hierarchies rather than on the partial orders on the row and column clusters, because the clusters in the reconstructed model may not fully correspond to those with the clusters in the true model. For a given $h \times n$ binary matrix **A**, we define the associated row quasi-order matrix **U** as the $h \times h$ binary matrix with $u_{ii'} = 1$ if $A_i \subseteq A_{i'}$ and 0 otherwise, with A_i and $A_{i'}$ being the sets of columns that row i and i' are associated with in **A**. Row quasi- order matrices **U** and **U*** were constructed for the matrix **T** and the matrix **M**, respectively, and,

Table 4: *Mean recovery of the true clusterings, as measured by the Corrected Rand Index, for different levels of Size, Rank, Error level and Algorithm*

		Greedy algorithm							Branch-and-bound algorithm						
		Error Level							Error Level						
		.00	.05	.10	.15	.20	.25	Overall	.00	.05	.10	.15	.20	.25	Overall
Size	15 × 25	.865	.659	.479	.328	.199	.108	.440	.908	.673	.498	.324	.187	.119	.452
	20 × 20	.843	.683	.482	.317	.223	.126	.446	.870	.704	.493	.336	.227	.135	.461
	80 × 20	.903	.816	.692	.509	.326	.204	.575	.925	.820	.706	.536	.337	.203	.588
	40 × 40	.909	.891	.762	.597	.446	.250	.642	.933	.904	.792	.642	.476	.272	.670
	500 × 50	.966	.972	.957	.886	.739	.534	.842	.974	.974	.965	.878	.732	.525	.841
	150 × 150	.938	.972	.997	.997	.985	.940	.972	.932	.976	1.000	.997	.992	.928	.971
Rank	3	.864	.906	.865	.741	.621	.473	.745	.876	.917	.869	.772	.644	.485	.761
	5	.935	.841	.726	.618	.490	.361	.662	.951	.862	.745	.610	.494	.378	.673
	8	.912	.750	.593	.457	.348	.247	.551	.944	.746	.613	.475	.338	.228	.557
Overall		.904	.832	.728	.606	.486	.360	.653	.924	.842	.742	.619	.492	.364	.664

Table 5: *Mean recovery of the true hierarchies, as measured by GOR, for different levels of Size, Rank, Error level and Algorithm*

		Greedy algorithm							Branch-and-bound algorithm						
		Error Level							Error Level						
		.00	.05	.10	.15	.20	.25	Overall	.00	.05	.10	.15	.20	.25	Overall
Size	15×25	.971	.929	.880	.829	.784	.743	.856	.979	.934	.884	.835	.784	.751	.861
	20×20	.966	.932	.886	.826	.790	.755	.859	.971	.939	.889	.829	.795	.761	.864
	80×20	.972	.956	.929	.878	.821	.770	.888	.977	.961	.932	.887	.827	.775	.893
	40×40	.972	.970	.944	.907	.866	.804	.911	.979	.976	.952	.921	.878	.815	.920
	500×50	.987	.991	.989	.971	.939	.884	.960	.989	.992	.990	.972	.942	.893	.963
	150×150	.976	.989	.999	.998	.995	.986	.991	.974	.990	1.000	.998	.997	.985	.991
Rank	3	.950	.967	.957	.916	.872	.819	.913	.953	.970	.958	.927	.884	.826	.920
	5	.983	.959	.930	.895	.855	.814	.906	.988	.968	.936	.896	.857	.821	.911
	8	.988	.957	.926	.894	.870	.838	.912	.993	.958	.930	.899	.870	.842	.915
Overall		.974	.961	.938	.902	.866	.824	.911	.978	.965	.941	.907	.871	.830	.915

subsequently, the proportion of cells for which $u_{ii'} = u^*_{ii'}$ $(i, i' = 1, \ldots, h)$ was calculated yielding a goodness-of-recovery statistic for the row quasi-order. Similarly, to evaluate the recovery of the column hierarchy, we constructed column quasi-order matrices **V** and **V***, and calculated the proportion of cells for which $v_{jj'} = v^*_{jj'}$ $(j, j' = 1, \ldots, n)$. Finally, an overall goodness-of-recovery value for each analysis was obtained by averaging the recovery values for rows and columns (weighted by the numbers of rows and columns, respectively). Table 5 displays, for both algorithms, the mean GOR at each level of the between-block variables.

Concerning the separate evaluation of each algorithm, the mean GOR amounts to .911 and .915 for the original and new algorithm, respectively, which indicates that, on the average, more than 90% of the hierarchical relations between the elements are correctly recovered. Again, the analyses of variance show similar results for both algorithms; moreover, these results are very similar to the results for the corrected Rand index. GOR is considerably affected by Error level ($\hat{\rho}_I$ = .36, .36), and Size ($\hat{\rho}_I$ = .32, .31), while the Size by Error level interaction ($\hat{\rho}_I$ = .14, .15) qualifies these main effects: Error level affects the goodness-of-recovery in small data sets only; for large matrices, GOR is close to perfect even for the highest error levels.

Concerning the comparison of both algorithms, the overall difference between both algorithms is significant ($t(2699) = 8.9$, $p < .01$), though small. An analysis of variance does not show any appreciable effects of the between-block variables on this difference.

With respect to the recovery of the association relation, we compare the proportion of discrepancies between **M** and **T** with the proportion of discrepancies between **D** and **T**. For this purpose, we define the relative recovery gain:

$$RRG = \frac{e - \frac{\sum_{i=1}^{h}\sum_{j=1}^{n}(t_{ij}-m_{ij})^2}{h \times n}}{e}.$$

This statistic takes a maximum value of 1 in the case of perfect recovery, and a value of 0 if the model is as far from the truth as the data are. As recovery and fit are equivalent for error-free data sets, and RRG is not defined if $e = 0$, analyses with $e = 0$ are omitted from the calculations below. Table 6 displays the mean RRG for both algorithms at every level of the between-block variables.

Concerning the separate evaluation of both algorithms, the overall mean RRG equals respectively .465 and .485 for the original and new algorithm. Hence, the models resulting from the algorithms are, on the average, about halfway between the data and the truth. Again, the effect sizes of the between-block variables and their interactions are similar for both algorithms: There is a huge main effect of Size ($\hat{\rho}_I$ = .55, .53); Large matrices have a RRG close

Table 6: *Mean recovery of the association relation, as measured by the Relative Recovery Gain, for different levels of Size, Rank, Error level and Algorithm*

		Greedy algorithm						Branch-and-bound algorithm					
		Error Level						Error Level					
		.05	.10	.15	.20	.25	Overall	.05	.10	.15	.20	.25	Overall
Size	15 × 25	.140	.162	.089	.061	.001	.090	.213	.201	.116	.037	.012	.116
	20 × 20	.183	.180	.105	.080	.046	.119	.282	.208	.108	.087	.043	.145
	80 × 20	.511	.524	.389	.214	.161	.360	.524	.546	.421	.233	.136	.372
	40 × 40	.637	.570	.456	.347	.181	.438	.706	.626	.506	.381	.204	.484
	500 × 50	.907	.940	.888	.770	.604	.822	.931	.953	.884	.779	.610	.831
	150 × 150	.893	.993	.994	.979	.933	.959	.910	.999	.994	.988	.929	.964
Rank	3	.759	.797	.688	.600	.481	.665	.791	.807	.730	.627	.480	.687
	5	.547	.539	.477	.403	.316	.457	.650	.588	.478	.404	.326	.489
	8	.329	.349	.295	.223	.166	.272	.342	.371	.307	.221	.161	.280
Overall		.545	.562	.487	.409	.321	.465	.594	.589	.505	.417	.322	.485

to perfect, whereas small matrices have a (very) low RRG. Also, Rank has a considerable effect ($\hat{\rho}_I$ = .16, .17): the larger the rank, the worse the recovery.

Concerning the comparison of both algorithms, higher RRG means are found for the branch-and-bound algorithm ($t(2249) = 5.7, p < .01$). But none of the between-block variables appreciably affects the difference between both algorithms.

4.4 Heuristics for rank selection

With respect to rank selection, three heuristics are evaluated. For each heuristic we presupposed that analyses should be carried out in ranks 1 through v and that the BOF of each model has been calculated. Table 7, which displays the BOF statistics for a particular data set in our study, serves as a guiding example to illustrate the three heuristics; the associated BOF by analysis rank plot is shown in Figure 3. Rule A, which may be called an "elbow criterion" or "scree test" (Cattell 1966), selects the rank k that maximizes the difference between the improvement in BOF from rank $k-1$ to rank k, and the improvement from rank k to rank $k+1$; this approach means that Rule A selects the rank k, at which the angle between two neighboring segments in the BOF plot is sharpest. For the example in Table 7, this situation arises for k equal to 5. Rule B may be considered a generalized scree test, in selecting the rank k with maximal difference between the mean improvement in BOF from rank 1 to rank k, and the mean improvement in BOF from rank k to rank v. In Table 7, this situation arises at rank 3. In rule C (McKenzie, Clarke, and Low 1992), a pseudo-binomial test is used, similar to the pseudo F-tests proposed by Hartigan (1975, pp. 89–91) to determine the number of clusters in k-means cluster analysis. The rule selects the smallest k for which the number of discrepancies of the model in rank $k+1$ exceeds the 10-th percentile of the binomial distribution Bin (hn, BOF_k), with hn equaling the number of cells in the data matrix and BOF_k as defined in (5). In the example in Table 7, this situation arises for rank 6.

In our simulation study, each data set was analyzed in ranks 1 through 8. For the comparison of the above rules for rank selection, only analyses from the branch-and-bound algorithm will be considered, because (a) the latter algorithm seems to do better than the greedy algorithm, and (b) the difference between both algorithms was not found to be mediated by other factors in the design. Furthermore, matrices with True rank equal to 8 were not taken into account, because, by construction, none of the three rules described above can ever select the highest analysis rank v. On the average, Rule C is superior with about 60% correct estimations (see Table 8). In particular, there is a strong difference between the rules for small matrices. For large matrices, Rule A even seems to

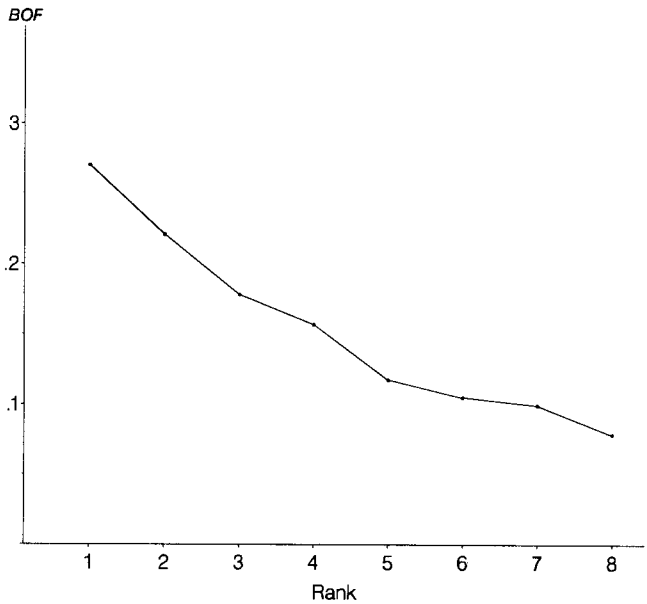


Figure 3. The BOF by rank plot for the example used in Table 7

do better than Rule C. It must be noted, however, that the range of possibly selected ranks is larger for Rule C than for both other rules: Rules A and B always select a rank within the range 2 to $v - 1$, whereas Rule C may select either a rank from 1 to $v - 1$ or no rank; the latter situation occurs when in each rank k the critical binomial test value at the 10-th percentile is not exceeded. To obtain a fairer comparison of the three rules, the goodness-of-recovery of the models selected by the respective rules are compared: The models selected by rule C generally have a considerably higher goodness-of-recovery than do the models selected by Rules A and B. For example, the mean RRG of the models selected by Rule C equals .60, whereas the mean RRG of the models of Rules A and B is .45 and .36, respectively.

As an aside, one may be interested in the relationship between goodness-of-fit and goodness-of-recovery. We calculated the Pearson product-moment correlation between the RRG and the BOF of the model selected by the superior Rule C. This correlation was calculated at each level of the independent variable Size separately, because the size of the matrix is known by the user (unlike the error level and true rank). These correlations are .27, .33, .53, .62, .70 and .43 for the 15×25, 20×20, 80×20, 40×40, 500×50 and 150×150 matrices, respectively. Hence, a good fit generally is indicative for a good recovery, particularly for larger matrices. The somewhat lower correlation for the 150×150 matrices probably stems from restriction of range, because the relative recovery gain for these matrices is close to perfect in most cases.

Table 7. *Badness-of-fit values for a data set of True rank 5, Size 80 × 20 and Error level .10 and statistics on which rank selection rules are based*

k	BOF_k	Rule A Statistic	Rule B Statistic	Rule C Statistic
1	.270			$< .001$
2	.221	.006	.025	$< .001$
3	.178	.022	.026	.013
4	.157	$-.018$	.018	$< .001$
5	.118	.026	.025	.063
6	.105	.007	.020	.246
7	.100	.014	.009	.003
8	.080			

Note. k: Analysis rank; BOF_k: Badness-of-fit value for rank k;
Rule A Statistic: $(BOF_{k-1} - BOF_k) - (BOF_k - BOF_{k+1})$;
Rule B Statistic: $(BOF_1 - BOF_k)/(k-1) - (BOF_k - BOF_8)/(8-k)$;
Rule C Statistic: $\mathrm{P}[X \leq (\mathrm{Size} \times BOF_{k+1}) | X \sim \mathrm{Bin}(\mathrm{Size}, BOF_k)]$.

Table 8. *Percentage of correct estimation of the true rank for each of the three heuristics*

	True rank = 3			True rank = 5		
	Rule A	Rule B	Rule C	Rule A	Rule B	Rule C
15 × 25	36	23	50	6	3	29
20 × 20	37	26	47	5	2	31
80 × 20	51	33	47	20	16	50
40 × 40	59	37	61	27	23	58
500 × 50	90	81	61	84	80	73
150 × 150	86	78	83	98	97	98
Overall	60	46	58	40	37	56

5. Discussion

This paper recapitulates the family of hierarchical classes models for two-way two-mode binary data. This family occupies a distinctive place within the classification domain because of the representation of three types of structural relations: equivalence relations, yielding a two-sided clustering, hierarchical relations, yielding a hierarchical organization of the clusterings, and the association relation. This paper further proposed a new, alternating least-squares algorithm for fitting a hierarchical classes model to a data set. Because of the duality and mathematical equivalences among the eight members of the hierarchical classes family, the algorithm can be used to fit any particular member (Van Mechelen et al. 1995).

A simulation study was conducted to evaluate both the new and the original hierarchical classes algorithm and to compare them. Both algorithms turned

out to succeed quite well in minimizing the least-squares loss function, as they seldom yield a model that fits the data worse than the underlying truth fits the data. With respect to the recovery of the underlying truth, it was found that the models yielded by the algorithms are considerably closer to the underlying truth than the data are. Moreover, this result holds for each of the three structural relations represented by the model. It was further found that with increasing size of the data matrix, goodness-of-fit decreased, whereas goodness-of-recovery increased. In particular, small data matrices, with a complex underlying truth, yielded a mean relative recovery gain close to zero. This finding generally implies that a HICLAS analysis in a high rank is useless for small-sized data sets. On the other hand, if the data matrix is large, the algorithm generally does succeed in finding the truth, even if the data are highly error-perturbed and the underlying truth is complex.

In De Boeck and Rosenberg's (1988) original simulation study an important effect of the size of the matrix on the goodness-of-recovery was also reported. Compared with the present study, De Boeck and Rosenberg's study revealed a better recovery for the original algorithm. This difference most probably results from the firm structure of De Boeck and Rosenberg's true matrices: There were only 5 row classes and 3 column classes, each class having either 5 entities (for the 25×15 matrix) or 10 entities (for the 50×30 matrix), and the bottom classes (i.e., classes of rows or columns of which the bundle pattern is a singleton) were nonempty. In the present study, the values on the bundles of the hierarchical classes model for the true matrices were randomly assigned.

Summarizing the comparison of both algorithms, the new algorithm was found to yield significantly higher goodness-of-recoveries and goodness-of-fits when compared to the original algorithm. But the difference is small. Furthermore, this difference is little or not affected by the size of the matrix, the error level, and the underlying true rank. The absence of any such effect implies that the new algorithm generally beats the original regardless of the properties of the data set analyzed. Whether the improvement compensates for the (slight) increase in runtime is to be decided by the user.

References

ARABIE, P., BOORMAN, S. A., and LEVITT, P. R. (1978), "Constructing Blockmodels: How and Why," *Journal of Mathematical Psychology, 17*, 21–63.

CARROLL, J. D., and CHATURVEDI, A. (1995), "A General Approach to Clustering and Multidimensional Scaling of Two-Way, Three-Way, or Higher-Way Data," in *Geometric Representations of Perceptual Phenomena*, Eds., R.D. Luce, M. D'Zmura, D. Hoffman, G.J. Iverson, and A.K. Romney, Mahwah, NJ: Erlbaum, 295–318.

CATTELL, R. B. (1966), "The Meaning and Stategic Use of Factor Analysis," in *Handbook of Multivariate Experimental Psychology*, Ed., R.B. Cattell, Chicago: Rand McNally, 174–243.

CHATURVEDI, A., and CARROLL, J. D. (1994), "An Alternating Combinatorial Optimization Approach to Fitting the INDCLUS and Generalized INDCLUS Models," *Journal of Classification, 11*, 155–170.

CHATURVEDI, A., and CARROLL, J. D. (1997), "An L_1 Norm Procedure for Fitting Overlapping Clustering Models to Proximity Data," in *L_1-Statistical Procedures and Related Topics, Volume 31*, Ed., Y. Dodge, Hayward, CA: IMS [Institute of Mathematical Statistics] Lecture Notes - Monograph Series [LNMS], 443–456.

DE BOECK, P. (1986), HICLAS *Computer Program: Version 1.0*, Department of Psychology, Katholieke Universiteit Leuven.

DE BOECK, P., and ROSENBERG, S. (1988), "Hierarchical Classes: Model and Data Analysis," *Psychometrika, 53*, 361–381.

DE BOECK, P., ROSENBERG, S., and VAN MECHELEN, I. (1993), "The Hierarchical Classes Approach: A Review," in *Categories and Concepts: Theoretical Views and Inductive Data Analysis*, Eds., I. Van Mechelen, J. Hampton, R.S. Michalski, and P. Theuns, London: Academic Press, 265–286.

HAGGARD, E. A. (1958), *Intraclass Correlation and the Analysis of Variance*, New York: Dryden.

HARTIGAN, J. A. (1975), *Clustering Algorithms*, New York: Wiley.

HAYS, W. L. (1994), *Statistics*, New York: Harcourt Brace.

HUBERT, L., and ARABIE, P. (1985), "Comparing Partitions," *Journal of Classification, 2*, 193–218.

KIM, K. H. (1982), *Boolean Matrix Theory and Applications*, New York: Marcel Dekker.

KIRK, R. E. (1982), *Experimental Design: Procedures for the Behavioral Sciences, 2nd Edition*, Belmont, CA: Brooks Cole.

LAMBERT, J. M., and WILLIAMS, W. T. (1962), "Multivariate Methods in Plant Ecology. IV. Nodal Analysis," *Journal of Ecology, 50*, 775–802.

LEENEN, I., and VAN MECHELEN, I. (1998), "A Branch-and-Bound Algorithm for Boolean Regression," in *Data Highways and Information Flooding, a Challenge for Classification and Data Analysis*, Eds., I. Balderjahn, R. Mathar, and M. Schader, Berlin: Springer-Verlag, 164–171.

LEENEN, I., VAN MECHELEN, I., and DE BOECK, P. (1999), "A Generic Disjunctive/Conjunctive Decomposition Model for n-ary Relations," *Journal of Mathematical Psychology, 43*, 102–122.

McKENZIE, D. P., CLARKE, D. M., and LOW, L. H. (1992), "A Method of Constructing Parsimonious Diagnostic and Screening Tests," *International Journal of Methods in Psychiatric Research, 2*, 71–79.

MICKEY, M. R., MUNDLE, P., and ENGELMAN, L. (1983), "Boolean Factor Analysis," in *BMDP Statistical Software*, Ed., W.J. Dixon, Berkeley, CA: University of California Press, 538–545, 692.

RAND, W. M. (1971), "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association, 66*, 846–850.

ROSENBERG, S., VAN MECHELEN, I., and DE BOECK, P. (1996), "A Hierarchical Classes Model: Theory and Method with Applications in Psychology and Psychopathology," in *Clustering and Classification*, Eds., P. Arabie, L.J. Hubert, and G. De Soete, Singapore: World Scientific, 123–155.

SNEATH, P. H. A., and SOKAL, R. R. (1973), *Numerical Taxonomy*, San Francisco: Freeman.

VAN MECHELEN, I. (1988), "Prediction of a Dichotomous Criterion Variable by Means of a Logical Combination of Dichotomous Predictors," *Mathématiques, Informatiques et Sciences Humaines, 102*, 47–54.

VAN MECHELEN, I., DE BOECK, P., and ROSENBERG, S. (1995), "The Conjunctive Model of Hierarchical Classes," *Psychometrika, 60*, 505–521.

VAN MECHELEN, I., ROSENBERG, S., and DE BOECK, P. (1997), "On Hierarchies and Hierarchical Classes Models," in *Mathematical Hierarchies and Biology*, Eds., B. Mirkin, F.R. McMorris, F.S. Roberts, and A. Rzhetsky, Providence: American Mathematical Society, 291–298.