Journal of Classification 28:363-389 (2011)
DOI: 10.1007/s00357-011-9089-5

The Real-Valued Model of Hierarchical Classes

Jan Schepers

K.U.Leuven, Belgium and Maastricht University, The Netherlands

Iven Van Mechelen

K.U.Leuven, Belgium

Eva Ceulemans

K.U.Leuven, Belgium

Abstract: We propose a non-negative real-valued model of hierarchical classes (HICLAS) for two-way two-mode data. Like the other members of the HICLAS family, the non-negative real-valued model (NNRV-HICLAS) implies simultaneous hierarchically organized classifications of all modes involved in the data. A distinctive feature of the novel model is that it yields continuous, non-negative real-valued reconstructed data, which considerably expands the application range of the HICLAS family. The expansion implies a major algorithmic challenge as it involves a move from the typical discrete optimization problems in HICLAS to a mixed discrete-continuous one. To solve this mixed discrete-continuous optimization problem, a two-stage algorithm combining a simulated annealing and an alternating local descent stage is proposed. Subsequently it is evaluated in a simulation study. Finally, the NNRV-HICLAS model is applied to an empirical data set on anger.

Keywords: HICLAS modeling; Real-valued HICLAS; Two-mode clustering; Simulated annealing; Alternating least squares.

J. Schepers and I. van Mechelen were supported by the Fund for Scientific Research-Flanders (Belgium), Project No. G.0146.06 awarded to Iven van Mechelen and by the Research Council KULeuven (GOA/2005/04). The authors would like to thank Peter Kuppens for providing the data set on anger.

Corresponding Author's Address: Jan Schepers, Faculty of Psychology and Neuroscience, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands, email: jan.schepers@maastrichtuniversity.nl

Published online 6 October 2011

1. Introduction

Data sets collected in many research domains can be regarded as a mapping from the Cartesian product of N sets of entities to some value set (for example, the set of reals $\mathbb{R}$ or the Boolean set $\mathbb{B} = \{0, 1\}$). If all sets in the Cartesian product are distinct, the data set is referred to as N-way N-mode (Carroll and Arabie 1980). For example, a rectangular objects by attributes data matrix is a typical example of a *two*-way *two*-mode data set in which the *set of objects* and the *set of variables* each comprise one of the two distinct modes.

A challenge for the data analyst is to capture the structural information that is present in N-way N-mode data. A powerful family of models that aim at capturing this information is the HICLAS family. As data-analytic techniques, HICLAS models combine two important aspects. Firstly, they are *powerful* because they represent the internal structure of all modes involved in the data, as well as the interrelations between these modes. Secondly, they are quite *unique* as well since they include the representation of psychologically often important (generalized) if-then relations. For example, *if* a patient suffers from hallucinations, *then* that patient also suffers from anxiety. Both of these aspects of HICLAS models can be included in a single comprehensive graphical representation. Some examples of where HICLAS models have been applied successfully in a variety of research domains in psychology can be found in Van Mechelen and De Boeck (1989), De Boeck and Van Mechelen (1990), Reid and Deaux (1996), Gara, Silver, Escobar, Holman, and Waitzkin (1998), Vansteelandt and Van Mechelen (2006), and Stirratt, Meyer, Ouellette, and Gara (2008).

A limitation of HICLAS models is that, up to now, they only imply binary or rating-valued two- or three-way reconstructed data (De Boeck and Rosenberg 1988; Leenen, Van Mechelen, De Boeck, and Rosenberg 1999; Ceulemans, Van Mechelen, and Leenen 2003; Van Mechelen, Lombardi, and Ceulemans 2007). Yet, often it is desirable to have models at one's disposal that include real-valued reconstructed data values. Indeed, in numerous research domains, the observed data are not of a categorical type, but rather are values on a continuous scale; examples include response times, muscle tensions and gene expression strength. To deal with such data properly it is desirable to approximate them by reconstructed data that take values from the same set, $\mathbb{R}$. In addition, in some cases one may wish to allow for positively real-valued reconstructed data, even when the actual data are not real-valued; for example, modeling a binary data set with reconstructed data values in the interval [0,1] may allow the reconstructed data to be interpreted in terms of probabilistic association strength.

In this paper we will propose a novel hierarchical classes model (NNRV-HICLAS) that yields continuous, non-negative real-valued reconstructed data. This model expansion is significant at two levels. Firstly, on the model side, the development of NNRV-HICLAS implies a transition from a discrete binary/rating-valued modeling to one that is hybrid continuous-discrete. This transition is reflected in a change of mathematical structure in which the decomposition rule of the model takes place. Substantively, it implies that continuous real-valued observed data need not be preprocessed in order to obtain discrete values (making them amenable to a HICLASR analysis) which would lead to loss of information. Instead, the model allows for much more flexibility in approximating the observed data matrix. One may argue that this flexibility may facilitate revealing underlying processes/structures in the data. Secondly, the estimation of NNRV-HICLAS models implies a major algorithmic challenge because it is characterized by a move from the typical discrete optimization problems in HICLAS to a mixed discrete-continuous one. We will propose and evaluate an algorithm to solve this optimization problem.

The remainder of this paper is organized as follows: In Section 2, we will discuss the novel non-negative real-valued model of hierarchical classes. In Section 3 we will show how this real-valued expansion can be estimated. Section 4 presents the results of a simulation study, an illustrative application is discussed in Section 5 and some concluding remarks are given in Section 6.

2. NNRV-HICLAS

2.1 Model

The non-negative real-valued model of hierarchical classes approximates a positively real-valued two-mode $I \times J$ object by attribute data matrix **D** by a two-mode $I \times J$ matrix **M** of positively real-valued reconstructed data entries m_{ij} that can be represented by a rank (R, S) NNRV-HICLAS model. As a guiding example, we will make use of the hypothetical reconstructed data matrix **M** in Table 1.

Three types of relations that are included in **M** are captured by the NNRV-HICLAS model: (a) equivalence, (b) hierarchy, and (c) association. (a) Two objects (attributes) are *equivalent* iff they correspond to identical rows (columns) in **M**. For example, suppose that the data entries in **M** represent the amount of time that is required by a student (i.e., a row element) to solve an item (i.e., a column element). Two students are then equivalent iff they solve each item in the same amount of time. Similarly, two items are equivalent iff each student solves both items in the same amount of time.

Table 1. Hypothetical Reconstructed Data Matrix **M**.

	attributes			
objects	a_1	a_2	a_3	a_4
o_1	3.12	2.17	3.12	1.17
o_2	1.11	3.16	3.16	1.19
o_3	3.12	3.16	3.16	1.19
o_4	0.16	1.04	2.07	2.07
o_5	1.11	3.16	3.16	1.19
o_6	3.12	3.16	3.16	2.07
o_7	3.12	2.17	3.12	1.17

(b) Object i (attribute j) is *hierarchically below* object i' (attribute j') iff all entries of the row (column) in **M** pertaining to object i (attribute j) are less than or equal to the corresponding entries of the row (column) in **M** pertaining to object i' (attribute j'). In particular, object i is hierarchically below object i' iff $\forall j = 1...J : m_{ij} \leq m_{i'j}$ (attribute j is hierarchically below attribute j' iff $\forall i = 1...I : m_{ij} \leq m_{ij'}$). These generalized if-then type relations are often psychologically interesting. For example, *if* a student i is hierarchically below a student i' *then* student i solves all the items at least as quickly as student i'. (c) The *association relation* is the mapping that links each object-attribute pair to its corresponding value in **M**. In our example, this is the amount of time that is required by a student i to solve an item j. To illustrate these three relations, in Table 1, objects o_2 and o_5 are equivalent, object o_1 is hierarchically below object o_3, attribute a_4 is hierarchically below attribute a_3, and object o_4 is associated to attribute a_1 by the value 0.16. One may note that the three relations captured by NNRV-HICLAS are generalizations of the ones captured in HICLASR, which in turn were generalizations of the ones captured in HICLAS. Indeed, the binary set is a subset of the set of rating values, and the latter is a subset of the set of positive reals.

A rank (R, S) NNRV-HICLAS model includes a binary $I \times R$ matrix **A**, a binary $J \times S$ matrix **B** and a $R \times S$ positively real-valued linking matrix **G**. **A** and **B** are called object and attribute bundle matrices and define probably overlapping clusters of objects and attributes, respectively. **G** is called the core matrix and contains the strengths of the associations between object and attribute bundles. In general one may note that there is no one-to-one correspondence between object and attribute bundles, with the number of object bundles R further possibly being different from the number of attribute bundles S. The $(3, 3)$ NNRV-HICLAS model for the reconstructed data in Table 1 is in Table 2.

The three types of relations mentioned above are represented by the NNRV-HICLAS model as follows: (a) Equivalence relations are represented by the bundle matrices in terms of identical binary row patterns for two

Table 2. NNRV-HICLAS model for **M** in Table 1.

Bundle matrices											
A				**B**				**G**			
	Object bundle				Attribute bundle						
objects	A_1	A_2	A_3	attributes	B_1	B_2	B_3		B_1	B_2	B_3
o_1	1	0	0	a_1	1	0	0	A_1	3.12	1.17	2.17
o_2	0	1	0	a_2	0	0	1	A_2	1.11	1.19	3.16
o_3	1	1	0	a_3	1	1	1	A_3	0.16	2.07	1.04
o_4	0	0	1	a_4	0	1	0				
o_5	0	1	0								
o_6	1	1	1								
o_7	1	0	0								

equivalent objects (attributes). By the representation of the equivalence relations and their implied equivalence classes (i.e., groups of equivalent students/groups of equivalent items), the NNRV-HICLAS model includes a partition of the two modes of the data. (b) Hierarchical relations are represented by subset-superset relations between the binary patterns corresponding to two objects (attributes). In particular, in the object (attribute) bundle matrix, the row pattern of ones of a hierarchically lower object (attribute) is a subset of the row pattern of ones of a hierarchically higher object (attribute). By the representation of the hierarchical relations, the classifications included in a NNRV-HICLAS model turn into hierarchically organized partitions because these relations are not only defined on the individual objects (attributes), but also on the equivalence classes, that is, the groups of equivalent objects (attributes). They can, hence, be considered as quasi-orders on the individual objects (attributes), or as partial orders on the corresponding equivalence classes. (c) The association relation, finally, is represented by the following decomposition rule:

$$m_{ij} = \max_{r=1}^{R} \max_{s=1}^{S} a_{ir} b_{js} g_{rs} \quad \forall i, j \tag{1}$$

where $\max_{r=1}^{R}$ denotes the maximum of the subsequent expression over all values of r ranging from 1 to R, a_{ir} is the entry in row i and column r of the object bundle matrix **A**, b_{js} is the entry in row j and column s of the attribute bundle matrix **B** and g_{rs} is the entry in row r and column s of the core matrix **G**. Rule (1) implies that object i is associated to attribute j with strength g_{rs} iff g_{rs} is the largest value in **G** for which it holds that object i and attribute j belong to bundles r and s, respectively. Van Mechelen et al. (2007) showed how the decomposition rule of the original binary HICLAS model can be rewritten as in (1). Hence, the NNRV-HICLAS decomposition rule includes that of the original binary HICLAS model as a special case.

Interestingly, for non-negative continuous data, and in case both bundle matrices **A** and **B** constitute partition matrices (i.e., all rows sum to one and no zero columns are present), decomposition rule (1) is equivalent to a two-mode partitioning (see e.g., Rosmalen, Groenen, Trejos, and Castillo 2009; Schepers and Hofmans 2009). Hence, it follows that for a given data matrix of positive values **D**, the optimal NNRV-HICLAS solution of rank (R, S) will approximate **D** at least as well as the optimal two-mode partitioning solution with R row clusters and S column clusters. To prevent misunderstanding, it must be emphasized that the advantage of NNRV-HICLAS over two-mode partitioning is the fact that the former can also capture (and represent) hierarchical relations between the object and attribute equivalence classes that imply a partition of both modes in the data.

2.2 Graphical Representation

The NNRV-HICLAS model can be given a comprehensive graphical representation. Figure 1 is a graphic representation of the NNRV-HICLAS model in Table 2. The upper half of the figure represents the partition of the objects, and the lower half that of the attributes. Object classes that are hierarchically below another object class are situated below the latter in this figure. Conversely, attribute classes that are hierarchically below another attribute class are situated above the latter. Hence, in the partition of the objects, hierarchical relations are represented by downward paths between the boxes that denote the equivalence classes, whereas in the partition of the attributes, they are represented by upward paths between the equivalence classes. The association relation between object i and attribute j is represented by the largest value contained in the hexagons which are included in a path between that object-attribute pair. For example, the value associated to object o_7 and attribute a_3 is the maximum of 3.12, 1.17 and 2.17, that associated to object o_2 and attribute a_1 is simply 1.11, because there is only one path connecting o_2 and a_1. Note that one object bundle may be linked, via a hexagon, to more than one attribute bundle, and vice versa.

As one of the reviewers remarked, distance information (e.g., between two objects) is obscured in the typical HICLAS graphical representation as described here. This may be undesirable in the context of real-valued data. It is no problem, however, to complement the NNRV-HICLAS representation with such distance information. For instance, Euclidean distances can be computed between the reconstructed data vectors corresponding to all pairs of equivalence classes that are connected by a link in the graphical representation. These distances, d, can then be displayed next to the links in question. Note that as NNRV-HICLAS aims to reconstruct observed data values, one may expect that observed distances between, say, two objects, are preserved

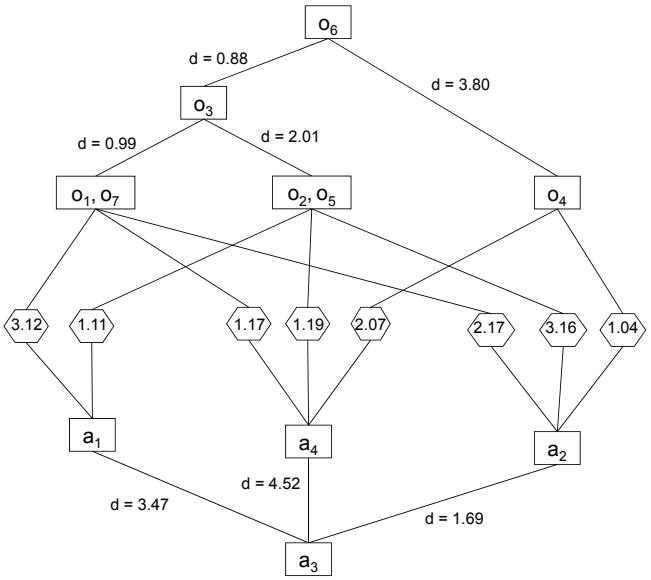


Figure 1. Graphic representation of the rank $(3, 3)$ NNRV-HICLAS solution of Table 2.

reasonably accurate as well in the reconstructed data. In Figure 1 we have included such distance information.

An alternative graphical representation of the NNRV-HICLAS solution is given in Figure 2, where the response profiles of the different object classes are displayed. From the profile of the object class consisting of object o_6, for instance, it can be seen that the reconstructed data value corresponding to that object class is, in all attribute classes, equal to or larger than that of the other object classes. This reflects the fact that, in Figure 1, object o_6 is hierarchically higher than all other objects. In general, this alternative graphical representation can be useful in cases where the number of object and attribute classes is modest. In other cases, interpretation of the solution will be more challenging, and one will usually rely on representations as in Figure 1.

2.3 Existence and Uniqueness

2.3.1 Existence

For each positively valued $I \times J$ data matrix $\mathbf{D}$, at least one NNRV-HICLAS decomposition exists that correctly represents the relations of equivalence and hierarchy. To prove this, we define a binary $I \times I$ matrix $\mathbf{A}$ which is such that $a_{ij} = 1$ iff row i in $\mathbf{D}$ is hierarchically higher or equivalent to

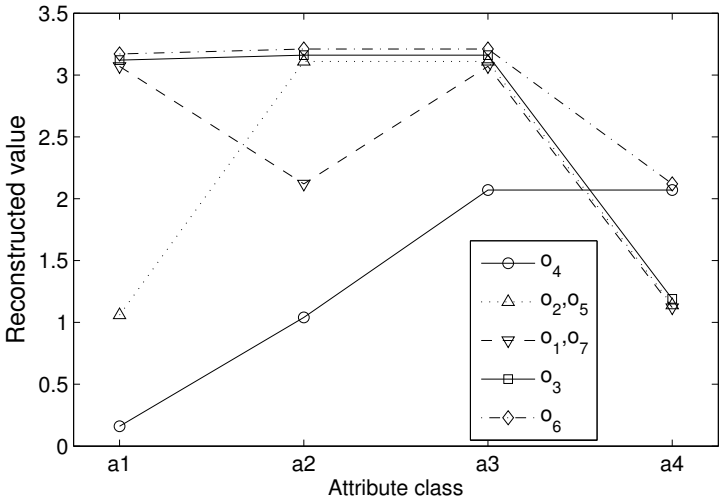


Figure 2. Alternative graphic representation of the rank $(3, 3)$ NNRV-HICLAS solution of Table 2.

row j in **D**; $a_{ij} = 0$ otherwise. A $J \times J$ matrix **B** can be defined with respect to the columns of **D** analogously. Assume further that $\mathbf{G} = \mathbf{D}$. From the definitions of **A** and **B** it then follows immediately that

$$d_{ij} = \max_{r=1}^{I} \max_{s=1}^{J} a_{ir} b_{js} d_{rs}, \tag{2}$$

with **A** and **B** representing the relations of equivalence and hierarchy between the rows and columns of **D**, respectively.

Closer study of decomposition rule (1) allows one to understand why NNRV-HICLAS can only be defined for non-negative real-valued data. Assume that in an observed data matrix the row vectors for two objects o_1 and o_2 correspond to $(-1\ -2)$ and $(-2\ -3)$, respectively. Clearly, o_1 is hierarchically higher than o_2. This implies that the bundle pattern of o_2 is a strict subset of that of o_1 and, hence, that the bundle pattern of o_2 must include at least one 0. However, it then follows from (1) that the reconstructed data entries of o_2 are at least equal to 0. Hence, in this case no solution exists that can represent the hierarchy relation correctly. Moreover, it can be deduced from (1) that in case some $m_{ij} = x < 0$, then the reconstructed data matrix **M** can include only two different values: 0 and x. The reason for restricting to the set of positive reals is because in case one considers the entire real set, then 0 – which is one of the two possible values for a_{ir} and b_{js} – is not neutral with respect to the maximum operator.

2.3.2 Uniqueness

For a given $I \times J$ matrix $\mathbf{M}$ that can be decomposed exactly according to (1), it holds that this decomposition is not necessarily unique, apart from the trivial nonuniqueness due to permutational freedom of the bundles. A significant part of the nonuniqueness, however, is removed by requiring the bundle matrices to correctly represent the hierarchical relations in $\mathbf{M}$. Only a limited number of cases then remain for which the NNRV-HICLAS decomposition is not unique. For these cases, however, it is possible to specify sufficient conditions for uniqueness, in analogy with earlier work by Van Mechelen, De Boeck, and Rosenberg (1995) and Ceulemans and Van Mechelen (2003). We give the result here, and refer to Schepers and Van Mechelen (2010) for a detailed proof: A rank (R, S) NNRV-HICLAS decomposition is unique if the reconstructed data matrix $\mathbf{M}$ contains a $R \times S$ submatrix $\mathbf{M}^*$ such that

$$\forall i, 1 \leq i \leq R, \exists j, 1 \leq j \leq S : m^*_{ij} > \max_{i'=1, i' \neq i}^{R} m^*_{i'j}, \tag{3}$$

and

$$\forall j, 1 \leq j \leq S, \exists i, 1 \leq i \leq R : m^*_{ij} > \max_{j'=1, j' \neq j}^{S} m^*_{ij'}. \tag{4}$$

Condition (3) means that every row of $\mathbf{M}^*$ contains a column element that is strictly larger than the corresponding column elements of the other rows of $\mathbf{M}^*$. Similarly, condition (4) means that every column of $\mathbf{M}^*$ contains a row element that is strictly larger than the corresponding row elements of the other columns of $\mathbf{M}^*$. It should be noted that conditions (3) and (4) are relatively mild. Indeed, as R is typically much smaller than I, and likewise S is typically much smaller than J, $I \times J$ matrices $\mathbf{M}$ typically contain a large set of different $R \times S$ submatrices. If for at least one of these submatrices the conditions (3) and (4) hold, then a rank (R, S) NNRV-HICLAS decomposition of $\mathbf{M}$ is unique. This means that, in practice, conditions (3) and (4) are often satisfied.

3. Estimation

The aim of an NNRV-HICLAS analysis is to look for a model matrix $\mathbf{M}$, that can be represented by an NNRV-HICLAS model of rank (R, S), and that minimizes the following L_p-norm based loss function:

$$f(\mathbf{M}) = (\sum_{i=1}^{I} \sum_{j=1}^{J} |d_{ij} - m_{ij}|^p)^{1/p}, \tag{5}$$

where d_{ij} denotes the actual observed data value for the pair of an object i and an attribute j. Standard choices for the loss function include $p = 1$ (least absolute deviations) and $p = 2$ (least squares). In the remainder, we will focus on the latter case. Note that the estimation problem in HICLASR is inherently linked to a least absolute deviations loss function. This is a consequence of the fact that one relies on a reparametrization into an equivalent binary Tucker3-HICLAS model (Ceulemans et al. 2003) in order to estimate a HICLASR solution (for more details, see Van Mechelen et al. 2007). Since in NNRV-HICLAS the estimation problem does not involve such a reparametrization, more flexibility is allowed with respect to the choice of loss function.

The HICLAS optimization problem in general implies that the best fitting $\mathbf{M}$ is looked for within the set of all possible $\mathbf{M}$ that can be represented by a certain type of HICLAS model of a prespecified rank. In all HICLAS models thus far, the model constituents included two or more bundle matrices, and, in some cases also a binary or rating-valued core matrix or array. This implies that up to now HICLAS optimization problems were fully discrete; algorithms developed for the estimation of binary and rating-valued HICLAS models all heavily relied on this fact. With respect to the estimation of real-valued HICLAS models, however, the optimization of (5) implies a new kind of problem in that the solution space now involves both discrete and real-valued variables. As a consequence it is not possible to use the existing HICLAS algorithms for the estimation of real-valued HICLAS models. In particular, in NNRV-HICLAS the estimation of $\mathbf{A}$ and $\mathbf{B}$ involves an optimization over the Boolean spaces $\mathbb{B}^{I\times R}$ and $\mathbb{B}^{J\times S}$, respectively, whereas the estimation of $\mathbf{G}$ involves an optimization over the real space ($\mathbb{R}^{R\times S}$).

Traditionally, HICLAS estimation problems have been tackled by making use of deterministic alternating local descent algorithms. In particular, these algorithms start from some initial configuration and then alternate between the estimation of the bundle matrices (and possibly a core array), at each step searching for an optimal conditional estimate. In order to find the solution that minimizes (5), we have developed such a deterministic alternating local descent algorithm (see below). However, it was found that this algorithm (using different random initial configurations) only occasionally yielded reasonable estimates. Moreover, the solutions resulting from different runs varied widely. Possibly, this may be a consequence of the fact that, unlike in other HICLAS models, no rational initial configuration can be provided. Instead, one can only make use of random starts. Inspired by recent promising findings in the context of binary HICLAS models (Ceulemans, Van Mechelen, and Leenen 2007), we therefore developed a simulated annealing algorithm to tackle the NNRV-HICLAS optimization

problem. One may note that if both a simulated annealing (SA) and a deterministic local descent (LD) algorithm are available to tackle the same optimization problem, it makes sense to combine both in an algorithm consisting of two stages. The first stage is an SA step and the second one an LD that starts from the solution returned by the SA stage. Using such a two-stage procedure ensures that at least a locally optimal solution is obtained (Aarts and Lenstra 1997) since the LD stage may act as a (quick) local optimum check in case no improvement was reached, or it may improve the value of the loss function to a (local) minimum. In the study of a different model Brusco (2001) found in this regard that the combination of SA and LD showed good results. Every run of the two-stage algorithm proposed here starts from a randomly chosen initial solution. As mentioned, we do not make use of the possibility of providing the algorithm with a rational starting configuration. This is because it is not clear how such a rational initial solution could readily be obtained. Moreover, the simulated annealing technique initially explores a large region of the solution space by a sequence of *random* perturbations, starting from the initial solution (see below). As a result, the process quickly wanders off to some solution that is far from the initial solution. One may argue that, because of this, the benefit of using a rational start (i.e., being near to the global optimum) would not be exploited anyway. As discussed above, the algorithm may end up in a local optimum. Therefore, it is desirable to run it multiple times, each time starting from a different initial solution. The NNRV-HICLAS algorithm has been implemented in a MATLAB package which is freely available from http://ppw.kuleuven.be/okp/software/NNRVHICLAS.

3.1 First Stage: Simulated Annealing

For a general introduction, see Aarts and Lenstra (1997). The *basic architecture* of the SA stage is the following: Given a current solution for (5), a trial solution is generated and evaluated to see whether its loss function value is smaller than that of the current solution. If this is the case, the current solution is always replaced by the trial solution; if the trial solution does not improve the loss function value the current solution is replaced by it with probability

$$P = e^{\frac{f_c - f_t}{\tau}}, \tag{6}$$

with τ denoting the current temperature of the annealing process, and f_c and f_t the values of the loss function corresponding to the current and trial solutions, respectively. During the annealing process, the temperature τ is gradually decreased in a series of steps (chains), resulting in an expected decrease of the probability parameter (6) during the annealing process. Within each chain, τ is kept constant.

To estimate the NNRV-HICLAS model, the following specifications and metaparameters have been chosen for the SA stage of the algorithm. In a few pilot studies of the implemented algorithm, these choices were found to be adequate in terms of keeping an optimal balance between computation time and quality of the returned solutions. First, the *initial solution* is generated by drawing the entries of the bundle matrices $\mathbf{A}$ and $\mathbf{B}$ iid from a Bernoulli distribution with parameter $\pi = .5$, subject to the restriction that no empty bundles are present. The entries in the initial core matrix $\mathbf{G}$ are further drawn iid from a uniform distribution with lower and upper bound equal to d_{min} and d_{max}, respectively, as those can be easily shown to be the theoretical minimum and maximum for the core values. To generate *trial solutions* the following stochastic mechanism is used given a current solution $(\mathbf{A}^c, \mathbf{B}^c, \mathbf{G}^c)$. A trial solution $(\mathbf{A}^t, \mathbf{B}^t, \mathbf{G}^t)$ is generated by altering the value of one arbitrarily selected single entry of $\mathbf{G}^c$, or one arbitrarily selected row pattern in $\mathbf{A}^c$ or $\mathbf{B}^c$, making use of a discrete uniform distribution. For the binary object (resp. attribute) bundle matrices, altering the value of an object (resp. attribute) row pattern implies that it is replaced by a pattern drawn uniformly from the set of all 2^R (resp. 2^S) binary row patterns, whereas for the real-valued core matrix $\mathbf{G}^c$, altering one single entry implies that its current value is replaced by a value drawn uniformly from the real interval $[d_{min}, d_{max}]$. Third, an *initial temperature* is chosen such that the annealing process accepts a high number of trial solutions in the initial stage of the process. For this purpose, one chain is generated in which all trial solutions are accepted. For every pair of successive solutions, the difference $\triangle f = f_{n-1} - f_n$ between the corresponding loss function values is stored if this difference is smaller than zero (or, in other words, if the new solution is worse). After a complete pass through this initial chain, the estimated initial temperature τ_{init} is calculated as:

$$\tau_{init} = \frac{\text{average} \triangle f}{\ln(0.8)}, \tag{7}$$

resulting in an average acceptance probability for worse solutions of .8 at τ_{init} (see e.g., Murillo, Fernando Vera, and Heiser 2005). This value ensures that a large region of the solution space is explored during the initial stage of the algorithm. Next, the choice of chain length, cooling factor and final temperature were based on the results of Ceulemans et al. (2007) in their study of the performance of simulated annealing in the context of the original binary HICLAS model. In particular, the *chain length*, that is, the number of generated solutions within one chain, was set equal to $2^R I + 2^S J$ such that it depends on the complexity of the problem at hand. With regard to the cooling factor, each time the algorithm has passed through one complete chain, the current temperature of the annealing process (τ_c) is lowered

to $\tau_{new} = .975\tau_c$. Finally, the final temperature τ_{final} was set equal to 10^{-6}. This choice implies that

$$TD = \frac{\log(10^{-6}) - \log(\tau_{init})}{\log(.975)}, \tag{8}$$

chains will be passed through during the whole annealing process (Brusco 2001), after which the algorithm stops. As can be seen, τ_{init} appears in expression (8). This implies that, through the initial temperature, the data range affects the number of chains that will be passed through, which is not desirable. For that reason, the data entries are first multiplied by a constant c such that the norm of the data matrix equals 1. Multiplying all data entries by a constant does not affect the optimal solution, except for the scale of the entries in **G**. After the annealing process has finished, the entries of the estimated **G** are then multiplied by $1/c$ in order to set the estimated solution back to the original scale of the observed data. The best overall solution encountered during the whole process is finally retained and returned as the estimated solution after completion of the SA algorithm.

3.2 Second Stage: Alternating Local Descent

In the second stage of the algorithm, trial solutions worse than the current solution are no longer accepted. In particular, starting with initial values ($\mathbf{A}^0$,$\mathbf{B}^0$,$\mathbf{G}^0$), which are provided by the simulated annealing stage, alternatingly, the best fitting $\mathbf{G}^w$, given $\mathbf{A}^{w-1}$ and $\mathbf{B}^{w-1}$, the best fitting $\mathbf{A}^w$, given $\mathbf{B}^{w-1}$ and $\mathbf{G}^w$, and the best fitting $\mathbf{B}^w$, given $\mathbf{A}^w$ and $\mathbf{G}^w$ are looked for ($w = 1, 2, ...$), until no more improvement in the value of the loss function is observed.

The conditional estimation of, for example, the object bundle matrix **A**, given **B** and the core **G**, comes down to the following optimization problem:

$$\min_{a_{ir} \in \{0,1\}} \Big(\sum_{i=1}^{I}\sum_{j=1}^{J} |d_{ij} - \max_{r=1}^{R}\max_{s=1}^{S} a_{ir}b_{js}g_{rs}|^p\Big)^{1/p}. \tag{9}$$

Since (9) satisfies a separability property (Chaturvedi and Carroll 1994), conditionally optimal estimates for **A** can be obtained by optimizing the individual rows of **A** separately. To this end, for each row separately, an enumerative evaluation of all 2^R binary patterns is performed; in case of ties, the solution with the largest number of ones is retained. For instance, in case it turns out that for some object i two bundle patterns (0 0 1) and (0 1 1) are both optimal with respect to (9), the latter estimate is chosen. This ensures that the set-theoretical relations in **M** are necessarily represented correctly by the bundle matrices (for a more detailed discussion on how

HICLAS models reduce nonidentifiability of the associated decomposition by means of such a closure operation, see De Boeck and Rosenberg 1988). A conditionally optimal estimate of the attribute bundle matrix **B** is obtained similarly.

The conditional estimation of the real-valued core matrix **G** comes down to the following continuous optimization problem:

$$\min_{g_{rs}\in\mathbb{R}^+}\Big(\sum_{i=1}^{I}\sum_{j=1}^{J}|d_{ij}-\max_{r=1}^{R}\max_{s=1}^{S}a_{ir}b_{js}g_{rs}|^p\Big)^{1/p}. \tag{10}$$

Locally optimal estimates of (10) can be obtained by applying multivariate continuous optimization procedures such as Quasi-Newton, Levenberg-Marquardt or Sequential Quadratic Programming (SQP) (for detailed information on these optimization procedures, see Nocedal and Wright 1999). We propose to use a quasi-Newton line-search SQP approach, which also allows to provide the algorithm with the theoretical lower and upper bounds for the core entries, i.e., d_{min} and d_{max}, respectively. Note that as an initial estimate the previous estimate $\mathbf{G}^{w-1}$ is used. This implies that the sequence of loss function values, associated to the alternating conditional estimations of **A**, **B** and **G**, converges monotonically to a (possibly local) minimum.

4. Simulation Study

In this section, we present a simulation study to examine the performance of the proposed NNRV-HICLAS algorithm with respect to the minimization of the loss function (4.2.1), CPU time (4.2.2) and the extent to which the estimated solutions recover the truth underlying the data (4.2.3).

4.1 Design

In our simulation study three different types of non-negative real-valued $I \times J$ matrices must be distinguished: a true matrix **T**, which can be represented by an NNRV-HICLAS model of a prespecified rank; a data matrix **D**, which is **T** perturbed with error; and the model matrix **M** yielded by the algorithm, which can be represented by a NNRV-HICLAS model of the same rank as the true matrix **T**.

The design of the simulation study was fully crossed, the factors being as follows:

(1) the *Size*, $I \times J$, of **T**, **D**, and **M**, at 4 levels: 20×20, 40×40, 80×40, 150×20;

(2) the *Rank*, (R, S) of the two-mode NNRV-HICLAS model for **T** and

M, at 4 levels: $(2,2)$, $(4,2)$, $(6,2)$, $(4,4)$, $(6,4)$, $(6,6)$;

(3) the *Error*, ε, which is the expected proportion of variance in **D** due to error, at 4 levels: .00, .05, .15, .30;

For each combination of these three independent variables, 15 replicates were studied, yielding 15 $\times$ 4 (*Size*) $\times$ 6 (*Rank*) $\times$ 4(*Error*) = 1440 simulated data sets.

For each combination of the levels of *Size*, *Rank* and *Error*, the 15 true matrices **T** corresponding to the 15 replicates were constructed as follows: Bundle matrices **A** and **B** were randomly generated with entries that were independent realizations of a Bernoulli variable with probability parameter equal to .5, subject to the restriction that no empty columns were present. The entries in the core matrix **G** were independent realizations of a uniformly distributed variable on the real interval [0,1]. The true matrix **T** resulted from combining **A**, **B** and **G** by (1). Subsequently, a data matrix **D** was constructed by adding error to each true matrix **T** using the following expression:

$$d_{ij} = \max_{r=1}^{R} \max_{s=1}^{S} a_{ir} b_{js} g_{rs} + e_{ij} \quad \forall i,j, \tag{11}$$

where e_{ij} was sampled from $N(0,\sigma_\epsilon^2)$ with $\sigma_\epsilon^2 = \frac{\epsilon}{1-\epsilon}\sigma_{\mathrm{T}}^2$, ($\sigma_{\mathrm{T}}^2$ denoting the variance across all matrix entries of **T**).

Finally, all data matrices **D** were analyzed in the true rank (R,S) with a loss function based on the L_2-norm by the NNRV-HICLAS algorithm. Each analysis of a data matrix **D** yielded one model matrix **M**; therefore, in total 15 (*replicates*) $\times$ 4 (*Size*) $\times$ 6 (*Rank*) $\times$ 4 (*Error*) different triplets (**T**,**D**,**M**) were obtained.

4.2 Results

4.2.1 Local Optima

In absence of knowledge about the global optimum, for each solution **M** we evaluated whether the loss value of **M** was less than or equal to the loss value of the truth. Note that the loss value of the truth equals

$$f(\mathbf{T}) = (\sum_{i=1}^{I}\sum_{j=1}^{J} |d_{ij} - t_{ij}|^2)^{1/2}. \tag{12}$$

When the data are analyzed in the true rank, a loss value of **M** greater than that of the truth implies certainty that the obtained solution is a local optimum.

For only 2 data sets (out of the 1440 data sets in the simulation study) no solution was returned (out of 50 runs) with a loss value less than or equal

to that of the truth (12). Since we also kept record of the loss value returned after the first stage (SA) of the algorithm, it is possible to assess to what extent the second stage (LD) improves the solutions returned by the first stage. It turns out that if only the SA stage would be adopted, then for 99 data sets local optimality of the obtained solution is certain. In other words, including the second stage (LD) in the algorithm clearly improves its performance.

Next, the Attraction rate (Vera, Macias and Heiser 2009), defined as the number of times (out of 50 runs) the minimal loss value is returned by the algorithm, was calculated for each data set. We analyzed the effect of the three design factors on Attraction rate by means of an analysis of variance. Table 3 shows effect sizes (η^2) for all main and interaction effects of the three factors. We define those effects for which it holds that $\eta^2 > .10$ as sizable. The effects of *Rank* ($\eta^2 = .25$) and *Error* ($\eta^2 = .29$) clearly stand out. Figure 3 shows that Attraction rate decreases as the number of entries of the core matrix (i.e., RS) increases and as the amount of error in the data increases. Furthermore, it appears that in case one is interested in estimating an NNRV-HICLAS solution of rank $(6, 6)$ or larger, one may wish to increase the number of restarts.

4.2.2 CPU Time

The algorithm as described here was implemented in Matlab, working on an Intel Xeon X5560 (2.80GHz) processor. In order to investigate computational cost, CPU times (i.e., the mean over 50 runs) for each data set were calculated. We analyzed the effect of the three design factors on CPU time by means of an analysis of variance. Table 4 shows effect sizes (η^2) for all main and interaction effects of the three factors. It appears that, next to their main effects, a sizable interaction between *Size* and *Rank* is present. Figure 4 shows that mean CPU time increases with larger sizes, and that this increase is more pronounced in case the rank of the estimated solution implies at least one mode with a number of bundles equal to 6.

As an aside, one may note that the amount of CPU time required by the second stage (LD) of the algorithm is on average only 3.7% of the total CPU time required by the algorithm (with a standard deviation of 2.5%). Taking into account the increased performance of the algorithm in terms of minimizing the loss function, the benefit of using a second stage therefore seems to outweigh its small cost.

4.2.3 Recovery of Underlying Truth

In order to assess how well the estimated solutions recover the underlying truth we calculated the relative badness-of-recovery (RBOR) of the best solution out of 50 runs for each simulated data set:

Table 3. Effect sizes (η^2) calculated for all main and interaction effects of Size, Rank and Error on Attraction rate.

Effect	η^2
Size	.01
Rank	.25
Error	.29
Size × Rank	.01
Size × Error	.02
Rank × Error	.01
Size × Rank × Error	.02

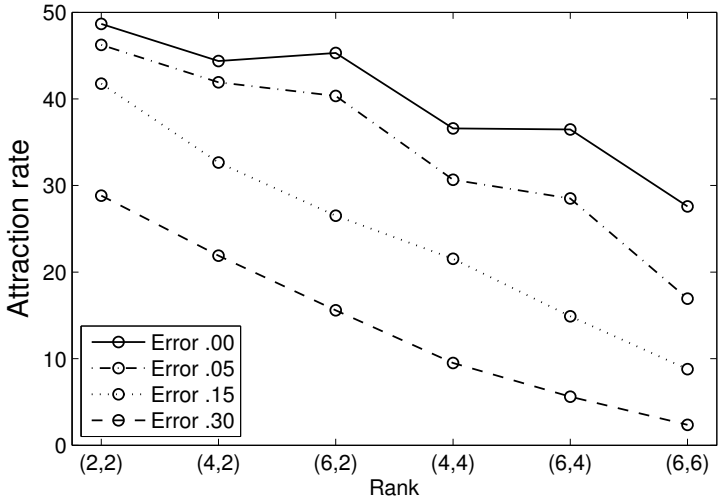


Figure 3. Mean Attraction rate as a function of *Rank* and *Error*.

$$\text{RBOR} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(t_{ij} - m_{ij})^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}(t_{ij} - \bar{t})^2}, \tag{13}$$

where $\bar{t}$ denotes the overall average of all entries in $\mathbf{T}$. RBOR is 0 in case of perfect recovery and 1 in case of poor recovery because in that case the fitted model entries m_{ij} would on average approximate the true values t_{ij} no better than the single constant value $\bar{t}$. We analyzed the effect of the independent variables on RBOR by means of an analysis of variance. Table 5 shows effect sizes (η^2) for all main and interaction effects of the three independent variables. Sizable effects are observed for *Error* and its interaction with *Size*. Figure 5 shows that mean RBOR increases with increasing amounts of error in the data, implying that the estimates get further away

Table 4. Effect sizes (η^2) calculated for all main and interaction effects of Size, Rank and Error on CPU time.

Effect	η^2
Size	.32
Rank	.52
Error	.00
Size × Rank	.16
Size × Error	.00
Rank × Error	.00
Size × Rank × Error	.00

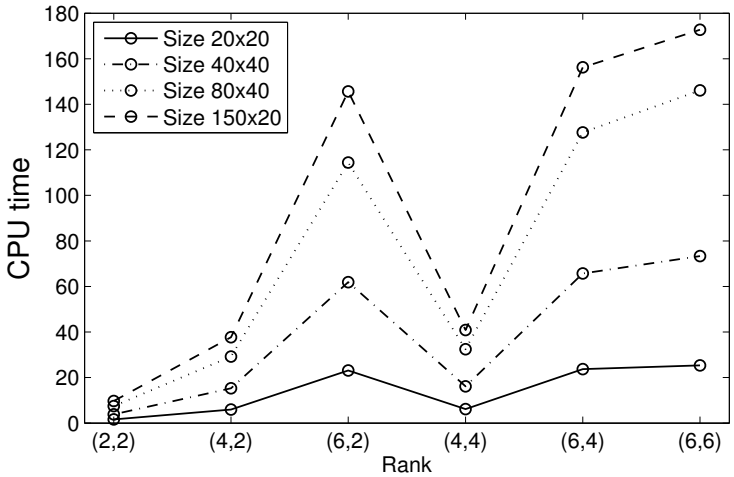


Figure 4. Mean CPU time (in seconds) for a single run, as a function of *Size* and *Rank*.

from the underlying truth when the proportion of noise in the data increases. Furthermore, performance appears to be better when both modes of the data include more than 20 elements. All in all, these results suggest that the algorithm recovers the underlying truth quite well.

5. Illustrative Application

In this section, we apply the NNRV-HICLAS model to a problem in the domain of componential emotion theory. In this domain, emotions are not treated as monolithic unities but rather as sets of components, including situational appraisals (e.g., *I feel attacked by someone else*) and emotional meta-experiences (e.g., *I feel angry*). A challenge for researchers in this area is to understand the interrelations (including implicational if-then relations) between these emotional components (see e.g., Kuppens, Van Mechelen, Smits, and De Boeck 2003).

Table 5. Effect sizes (η^2) calculated for all main and interaction effects of Size, Rank and Error on RBOR.

Effect	η^2
Size	.10
Rank	.07
Error	.44
Size × Rank	.03
Size × Error	.12
Rank × Error	.09
Size × Rank × Error	.04

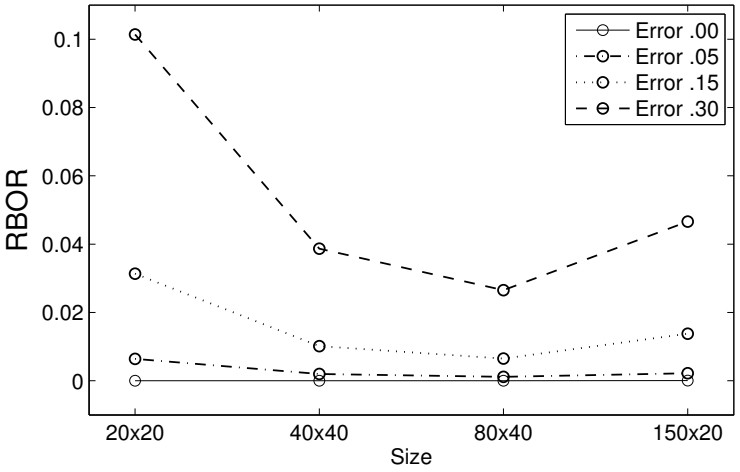


Figure 5. Mean RBOR as a function of *Size* and *Error*.

To study these relations with regard to the emotion of anger, we will make use of a data set in which 357 participants were asked to what degree they experienced a collection of 14 different emotion components in a set of 24 possibly anger-eliciting situations (Kuppens and Van Mechelen 2007) (see Tables 6 and 7 for the full lists of emotion components and situations, respectively). In the present reanalysis of these data we will only focus on the general psychological question: How does the average person react in different situations? We therefore will proceed with the 24×14 situation by emotion component matrix $\mathbf{D}$ as obtained after averaging the raw data across participants.

NNRV-HICLAS models were fitted to $\mathbf{D}$, in ranks ranging from $(1, 1)$ to $(5, 5)$, yielding a set of 25 solutions. Based on a convex hull procedure (Ceulemans and Kiers 2006; Schepers, Ceulemans, and Van Mechelen 2008) and on the interpretability of the different solutions, the solution in

Table 6. Description of all fourteen emotion components.

(1)	I am capable of undertaking action against it. (ACTION)
(2)	I think someone else does not take enough consideration of me. (NO CONSIDERATION)
(3)	I feel attacked by someone else. (ATTACKED)
(4)	I think someone else is responsible. (SOMEONE ELSE RESPONSIBLE)
(5)	I find the situation frustrating. (FRUSTRATION)
(6)	I feel angry. (ANGRY)
(7)	I am being treated unrespectfully. (NO RESPECT)
(8)	I think someone has bad intentions towards me. (BAD INTENTIONS)
(9)	My self-esteem is being threatened. (SELF-ESTEEM)
(10)	My plans/goals are being obstructed (OBSTRUCTION)
(11)	It really affects me as a person. (AFFECT PERSONALLY)
(12)	It is important to me. (IMPORTANT)
(13)	I think someone else is to blame. (SOMEONE ELSE TO BLAME)
(14)	I feel desperate. (DESPERATE)

rank $(3, 3)$ was selected; this solution accounted for 65.89% of the variance in **D**. Figure 6 shows the corresponding graphic representation.

In order to gain insight in the anger-eliciting process, we first focus on the *hierarchy of the situation classes*. To build an interpretation we consecutively examine the bottom classes along with all hierarchically higher situated classes, and look for distinctive characteristics for the corresponding sets of classes. For example, the empty leftmost bottom class along with its two hierarchically higher classes, contains situations such as CD-PLAYER and LET DOWN that all seem to indicate the presence of a deliberate infliction of harm (*INTENTIONALLY*). The middle bottom class (including situations such as SWIM) along with its hierarchically higher classes (including CD-PLAYER, LET DOWN and COMA) contains situations that all involve the presence of someone else rather than the participant under study (*OTHER INVOLVED*). Finally, the rightmost bottom class (including situations such as FAIL EXAM) along with its two hierarchically higher classes (including COMA and LET DOWN) involves situations with a large negative impact (*LARGE DAMAGE*). Note that the empty bottom class in the *INTENTIONALLY* set reflects the logical relation that intentionality requires the presence of someone else. To illustrate the hierarchical relations in this application, it appears that *if* a situation involves the presence of *OTHER INVOLVED* and *LARGE DAMAGE* (e.g., CAR ACCIDENT), *then* that situation elicits at least as much anger as a situation that only involves *LARGE DAMAGE* (e.g., ALARM CLOCK). A similar interpretation holds with regard to the hierarchical relations between the other situation(s) (classes).

Table 7. Description of all twenty-four situations.

(1)	Your friend is in a coma after an accident. (COMA)
(2)	A friend lets you down on a date, and calls you the following day to let you know that he/she didn't feel like meeting with you and went out with other people instead. (LET DOWN)
(3)	A friend returns your CD player, claiming that everything is OK, but it turns out to be broken afterwards. (CD-PLAYER)
(4)	A swimming appointment is cancelled because one of your friends falls ill. (SWIM)
(5)	The waiter in a restaurant informs you that it may take a while before you can eat because it is a busy evenening. Finally, you are served after 50 minutes of waiting. (RESTAURANT)
(6)	Upon leaving class, you notice that a police officer is removing your bike because it was illegally parked. (POLICE REMOVES BIKE)
(7)	You are hit by a car on your way to an important appointment, causing you to miss the appointment. (CAR ACCIDENT)
(8)	You arrange with a good friend to go out together, and he/she will contact you to meet each other. You don't hear from him/her. (NOT CALL)
(9)	On holiday with friends, you arrange that each, in turn, has to carry the heavy tent gear. One day, the tent gear is missing. (TENT)
(10)	You are hit on your bike by another biker. He/she apologizes, and proposes to pay back the damage to your bike. (BIKES HIT)
(11)	You have arranged for a hotel room with sea-view. Upon arrival, you are given a room without a sea-view. (SEA VIEW)
(12)	You are in love with someone but he/she is more interested in someone else. (LOVE SOMEONE ELSE)
(13)	You didn't study hard enough for an exam, and you fail the exam. (FAIL EXAM)
(14)	Your clock failed to wake you up in the morning and you miss the final class of a course. (ALARM CLOCK)
(15)	You arrange with your roommates that each in turn has to put out the garbage. When it is someone else's turn you noticed that he/she didn't clean up. (GARBAGE)
(16)	A floppy disk holding an important school assignment is destroyed by your computer. (FLOPPY)
(17)	You hear that a friend is spreading gossip about you. (GOSSIP)
(18)	You miss a popular party because you fall asleep at home. (MISS PARTY)
(19)	You are fired from your holiday job. (FIRED FROM HOLIDAY JOB)
(20)	A fellow student fails to return your notes when you need them for studying. (COURSE NOTES)
(21)	You bump into someone on the street. (BUMP)
(22)	You have a group assignment with some fellow students. They don't work hard, and you all get a bad grade. (GROUP ASSIGNMENT)
(23)	You're out for a drink after a hard day's work, and you have to wait 30 minutes before you are served. (DRINK)
(24)	Your roommates went to the movies without informing you. (MOVIES)

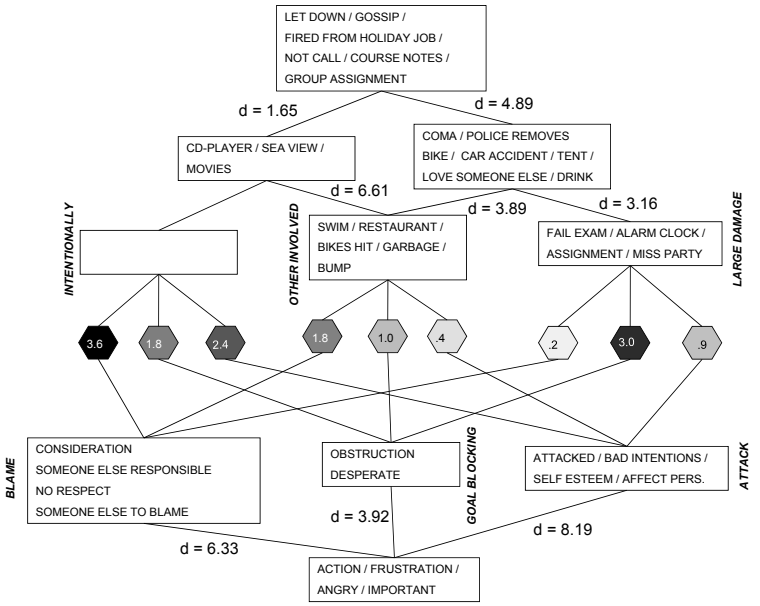


Figure 6. Graphic representation of the rank $(3, 3)$ NNRV-HICLAS solution for anger data.

Second, we consider the *hierarchy of the emotion component classes*. This hierarchy contains three non-empty bottom classes, and one class that spans these three. The top class mainly consists of emotional meta-experiences such as *I feel angry*. The three bottom classes in their turn represent channels through which the emotion of anger can be triggered. Specifically, a person may feel angry because he/she is being attacked (*ATTACK*), because his/her goal is being obstructed (*GOAL BLOCKING*), and/or because someone else is to blame (*BLAME*). To illustrate the hierarchical relations, it appears that *if* an emotion component involves the presence of *BLAME* and *GOAL BLOCKING* (e.g., FRUSTRATION), *then* that emotion component is associated to at least as much anger as an emotion component that only involves *BLAME* (e.g., SOMEONE ELSE TO BLAME). A similar interpretation holds with regard to the hierarchical relations between the other emotion component(s) (classes).

Third, we may examine the *association* between the situation and emotion component hierarchies. For this purpose, we look at the hexagons that link the two hierarchies, and that contain association strength values. In Figure 5, to facilitate the interpretation, the values in the hexagons are also visualized in terms of grayscale levels (with higher values corresponding to darker gray levels). It can be seen that *LARGE DAMAGE* is almost exclusively linked to *GOAL BLOCKING*, and *OTHER INVOLVED* almost exclu-

sively to *BLAME*, whereas *INTENTIONALLY* is linked to *GOAL BLOCKING*, *BLAME* as well as *ATTACK*. It can further be deduced that the highest levels of anger are elicited on the one hand by *LARGE DAMAGE* (which triggers anger through the *GOAL BLOCKING* channel) and on the other hand by *INTENTIONALLY* (which triggers anger primarily through *BLAME* but also to some lesser extent through the other two channels).

6. Discussion

Three relevant topics will be considered in this final section. We will first position the NNRV-HICLAS model within the context of the HICLAS family. Next, we will discuss the possibility and relevance of constrained NNRV-HICLAS solutions, and finally we will suggest some further extensions of the model.

6.1 Positioning within HICLAS Family

In this paper, we proposed a novel model of the HICLAS family. This novel member has a number of features in common with the other models in the family. These include: A representation of the internal structure of all modes involved in the data, including a representation of (generalized) if-then relations, and a representation of their interrelations; a comprehensive graphical representation. The novelty of the NNRV-HICLAS model lies in the fact that it implies positively real-valued reconstructed data: As such, it considerably expands the scope of data sets that may be subjected to a hierarchical classes analysis. The novel NNRV-HICLAS model is closely related to the HICLASR model (Van Mechelen et al. 2007) with both models relying on decomposition rule (1), the difference being that in HICLASR the core entries are discrete-valued whereas in NNRV-HICLAS they are positively real-valued. This seemingly small difference has far reaching consequences. Firstly, on the model side it implies a transition from a purely discrete type of modeling to one that is mixed discrete-continuous. Mathematically speaking this implies a transition from a matrix decomposition over a Boolean algebra to one over the tropical semiring $(\mathbb{R}^+, \max, \cdot)$ (where $\cdot$ denotes scalar multiplication). The mathematical study of this semiring has gained increasing attention in recent years; the interested reader may consult Baccelli, Cohen, Olsder, and Quadrat (1993) to learn more on this subject. Secondly, the estimation of the NNRV-HICLAS model requires a fully new discrete-continuous algorithmic strategy.

6.2 Constrained NNRV-HICLAS Models

The model discussed in this paper is an unconstrained real-valued hierarchical classes model. It may, however, be sensible to develop con-

strained variants of it. In this regard, a particularly interesting constraint could be to enforce **G** to be diagonal. Linking up with a framework developed by Ceulemans, Van Mechelen, and Kuppens (2004), this may be considered a value constraint. Such a constraint can be included in a straightforward way in the estimation approach as proposed in this paper. Note that a diagonal NNRV-HICLAS core implies a one-to-one correspondence between object and attribute bundles, such as in the original disjunctive binary HICLAS model for binary data (De Boeck and Rosenberg 1988). An example in which such a constrained model may be particularly useful is the case of a student by item family data matrix containing conditional proportions for a given student i to solve items from item family j correctly. In that case, the bundles as yielded by a constrained diagonal core NNRV-HICLAS model could be interpreted as latent solution strategies. On the student side, the bundles then indicate, for each individual, whether he or she masters the corresponding strategies or not, whereas on the item side, they denote whether a particular strategy is appropriate to solve each item family or not. The diagonal elements of the core then represent the probability that mastery of a solution strategy leads to a correct solution. If a certain student then masters two or more solution strategies through each of which items of a particular item family can be solved, the reconstructed conditional proportion for that student to solve the items of that family then will equal the highest succes probability associated to the strategies in question, as students can be assumed to pursue only the solution strategy that maximizes their chance of solving the item correctly.

6.3 Model Extensions

Several further extensions of the NNRV-HICLAS model can be considered; two of these will be discussed here in somewhat more detail. First, real-valued HICLAS models for three-way three-mode data can be developed. Such models could be considered real-valued counterparts for Tucker3-HICLAS (Ceulemans et al. 2003), Tucker2-HICLAS (Ceulmans and Van Mechelen 2004) and INDCLAS (Leenen et al. 1999), in which the core is allowed to contain positively real-valued entries. One field of application in which such models could be useful is the contextualized study of personality (Mischel and Shoda 1998; Kuppens, Van Mechelen, Smits, De Boeck, and Ceulemans 2007), and in particular the study of individual differences structure in situation-behavior profiles (Vansteelandt and Van Mechelen 1998), where three-mode person by situation by behavior data are a key object of research. Such data could be used, for example, to investigate whether individuals behave differently (e.g., with different anger-related responses such as verbal aggression, physical aggression, self-aggression) and how this may

vary as a function of the situation (e.g., the target of the anger-related behavior is a person of lower status vs. of higher status).

Second, the model as described in the present paper is based on the maximum operator and, as such, belongs to the class of (generalized) disjunctive HICLAS models. It is possible, however, to formulate a dual, conjunctive variant of the model that would rely on the min- rather than on the max-operator. Such a variant could be useful in the case of a student by item data matrix containing response times for a given student i to solve item j correctly. In that case, the bundles as yielded by a conjunctive NNRV-HICLAS model could again be interpreted as solution strategies (as in the example above). The core entries, however, then represent the amount of time a solution strategy requires to solve the item correctly. If a certain student then masters two or more solution strategies through each of which an item can be solved, the reconstructed response time for that student to solve the item will equal the smallest response time associated to the strategies in question. This model therefore assumes that students pursue only the solution strategy that leads to a correct solution in the smallest amount of time.

References

AARTS, E.H.L., and LENSTRA, J.K. (1997), *Local Search in Combinatorial Optimization*, New York: Wiley.

BACCELLI, F., COHEN, G., OLSDER, G.-J., and QUADRAT, J.-P. (1993), *Synchronization and Linearity: An Algebra for Discrete Event Systems*, London: John Wiley & Sons.

BRUSCO, M.J. (2001), "A Simulated Annealing Heuristic for Unidimensional and Multidimensional (City-Block) Scaling of Symmetric Proximity Matrices", *Journal of Classification, 18*, 3–33.

CARROLL, J.D., and ARABIE, P. (1980), "Multidimensional Scaling", *Annual Review of Psychology, 31*, 607–649.

CEULEMANS, E., and KIERS, H.A.L. (2006), "Selecting Among Three-Mode Principal Component Models of Different Types and Complexities: A Numerical Convex Hull Based Method", *British Journal of Mathematical and Statistical Psychology, 59*, 133–150.

CEULEMANS, E., and VAN MECHELEN, I. (2003), "Uniqueness of N-Way N-Mode Hierarchical Classes Models", *Journal of Mathematical Psychology, 47*, 259–264.

CEULEMANS, E., and VAN MECHELEN, I. (2004), "Tucker2 Hierarchical Classes Analysis", *Psychometrika, 69*, 375–399.

CEULEMANS, E., VAN MECHELEN, I., and KUPPENS, P. (2004), "Adapting the Formal to the Substantive: Constrained Tucker3-HICLAS", *Journal of Classification, 21*, 19–50.

CEULEMANS, E., VAN MECHELEN, I., and LEENEN, I. (2003), "Tucker3 Hierarchical Classes Analysis", *Psychometrika, 68*, 413–433.

CEULEMANS, E., VAN MECHELEN, I., and LEENEN, I. (2007), "The Local Minima Problem in Hierarchical Classes Analysis: An Evaluation of a Simulated Annealing Algorithm and Various Multistart Procedures", *Psychometrika, 72*, 377–391.

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.

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

DE BOECK, P., and VAN MECHELEN, I. (1990), "Traits and Taxonomies: A Hierarchical Classes Approach", *European Journal of Personality, 4*, 147–156.

GARA, M.A., SILVER, R.C., ESCOBAR, J.I., HOLMAN, A., and WAITZKIN, H. (1998), "A Hierarchical Classes Analysis (HICLAS) of Primary Care Patients With Medically Unexplained Somatic Symptoms", *Psychiatry Research, 81*, 77–86.

KUPPENS, P., and VAN MECHELEN, I. (2007), "Interactional Appraisal Models for the Anger Appraisals of Threatened Self-Esteem, Other-Blame, and Frustration", *Cognition & Emotion, 21*, 56–77.

KUPPENS, P., VAN MECHELEN, I., SMITS, D. J.M., and DE BOECK, P. (2003), "The Appraisal Basis of Anger: Specificity, Necessity and Sufficiency of Components", *Emotion, 3*, 254–269.

KUPPENS, P., VAN MECHELEN, I., SMITS, D. J.M., DE BOECK, P., and CEULEMANS, E. (2007), "Individual Differences in Patterns of Appraisal and Anger Experience", *Cognition & Emotion, 21*, 689–713.

LEENEN, I., VAN MECHELEN, I., DE BOECK, P., and ROSENBERG, S. (1999), "INDCLAS: A Three-Way Hierarchical Classes Model", *Psychometrika, 64*, 9–24.

MISCHEL, W., and SHODA, Y. (1998), "Reconciling Processing Dynamics and Personality Dispositions", *Annual Review of Psychology, 49*, 229–258.

MURILLO, A., VERA, J.F., and HEISER, W.J. (2005), "A Permutation-Translation Simulated Annealing Algorithm for L_1 and L_2 Unidimensional Scaling", *Journal of Classification, 22*, 119–138.

NOCEDAL, J., and WRIGHT, S.J. (1999), *Numerical Optimization*, New York: Springer.

REID, A., and DEAUX, K. (1996), "Relationship Between Social and Personal Identities: Segregation or Integration?", *Journal of Personality and Social Psychology, 71*, 1084–1091.

SCHEPERS, J., and HOFMANS, J. (2009), "TwoMP: a MATLAB Graphical User Interface for Two-Mode Partitioning", *Behavior Research Methods, 41*, 507–514.

SCHEPERS, J., CEULEMANS, E., and VAN MECHELEN, I. (2008), "Selecting Among Multi-Mode Partitioning Models of Different Complexities: A Comparison of Four Model Selection Criteria", *Journal of Classification, 25*, 67–85.

SCHEPERS, J., and VAN MECHELEN, I. (2010), "Uniqueness of Real-Valued Hierarchical Classes Models", *Journal of Mathematical Psychology, 54*, 215–221.

STIRRATT, M.J., MEYER, I.H., OUELLETTE, S.C., and GARA, M.A. (2008), "Measuring Identity Multiplicity and Intersectionality: Hierarchical Classes Analysis (HICLAS) of Sexual, Racial, and Gender Identities", *Self and Identity, 7*, 89–111.

VAN MECHELEN, I., and DE BOECK, P. (1989), "Implicit Taxonomy in Psychiatric Diagnosis: A Case Study", *Journal of Social and Clinical Psychology, 8*, 276–287.

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

VAN MECHELEN, I., LOMBARDI, L., and CEULEMANS, E. (2007), "Hierarchical Classes Modeling of Rating Data", *Psychometrika, 72*, 475–488.

VAN ROSMALEN, J., GROENEN, P.J.F., TREJOS, J. and CASTILLO W. (2009), "Optimization Strategies for Two-Mode Partitioning", *Journal of Classification, 26*, 155–181.

VANSTEELANDT, K., and VAN MECHELEN, I. (1998), "Individual Differences in Situation-Behavior Profiles: A Triple Typology Model", *Journal of Personality and Social Psychology, 75*, 751–765.

VANSTEELANDT, K., and VAN MECHELEN, I. (2006), "Individual Differences in Anger and Sadness: In Pursuit of Active Situational Features and Psychological Processes", *Journal of Personality, 74*, 871–910.

VERA, J.F., MACIAS, R., and HEISER, W.J. (2009), "A Dual Latent Class Unfolding Model for Two-Way Two-Mode Preference Rating Data", *Computational Statistics and Data Analysis, 53*, 3231–3244.