Journal of Classification 30:56-74 (2013)
DOI: 10.1007/s00357-013-9120-0

Additive Biclustering: A Comparison of One New and Two Existing ALS Algorithms

Tom F. Wilderjans
KU Leuven, Belgium

Dirk Depril
suAzio Consulting, Belgium

Iven Van Mechelen
KU Leuven, Belgium

Abstract: The additive biclustering model for two-way two-mode object by variable data implies overlapping clusterings of both the objects and the variables together with a weight for each bicluster (i.e., a pair of an object and a variable cluster). In the data analysis, an additive biclustering model is fitted to given data by means of minimizing a least squares loss function. To this end, two alternating least squares algorithms (ALS) may be used: (1) $PENCLUS$, and (2) Baier's ALS approach. However, both algorithms suffer from some inherent limitations, which may hamper their performance. As a way out, based on theoretical results regarding optimally designing ALS algorithms, in this paper a new ALS algorithm will be presented. In a simulation study this algorithm will be shown to outperform the existing ALS approaches.

Keywords: Biclustering; Additive clustering; PENCLUS; ALS algorithms; Simulation study; Simultaneous overlapping clusterings; Co-clustering; Two-mode clustering; Two-mode data.

The research in this paper was partially supported by the Research Fund of KU Leuven (PDM-kort project 3H100377, Dr. Tom F. Wilderjans; GOA $2005/04$, Prof. dr. Iven Van Mechelen), by the Belgian Science Policy (IAP P$6/03$, Prof. dr. Iven Van Mechelen), and by the Fund of Scientific Research (FWO)-Flanders (project G.0546.09, Prof. dr. I. Van Mechelen). The simulation study was conducted using high performance computational resources provided by KU Leuven (http://ludit.kuleuven.be/hpc). The authors are obliged to Prof. dr. W. Gaul for kindly providing information on the PENCLUS algorithm. We also would like to thank the Editor and three anonymous reviewers for their interesting and helpful comments which substantially improved the paper.

Corresponding Author's Address: Tom F. Wilderjans, Faculty of Psychology and Educational Sciences, KU Leuven, Andreas Vesaliusstraat 2, Box 3762, B-3000 Leuven, Belgium, Tel: +32.16.32.61.23, Fax: +32.16.32.62.00, e-mail: tom.wilderjans@ppw.kuleuven.be

Published online 11 January 2013

1. Introduction

Two-way two-mode object by variable data sets are often encountered in statistical practice. Examples of such data include patient by symptom data on symptom severity (see, e.g., Gara, Rosenberg, and Goldberg 1992; Mezzich and Solomon 1980; Van Mechelen and De Boeck 1989, 1990), and gene by tissue data on gene expression levels (see, e.g., Faith et al. 2007; Gasch et al. 2000; Segal et al. 2003; Spellman et al. 1998). Substantive questions regarding such data include the identification of both patterns in the data and underlying structures or mechanisms. For patient by symptom data, such structures may, for example, correspond to certain diseases or syndromes, with each disease pertaining to a set of patients who suffer from it and a set of associated symptoms; for gene by condition data, a relevant structure may be a module consisting of a number of coregulated genes that display a similar expression pattern in a particular set of conditions. Note that both example structures are discrete in nature.

To reveal discrete structures in a given object by variable data set, one may appeal to a simultaneous clustering of both the objects (patients, genes) and the variables (symptoms, conditions) involved in the data, resulting in a set of *biclusters* (i.e., pairs of object and variable clusters)[1]. A large number of biclustering methods has already been proposed in the literature (for an overview, see Madeira and Oliveira 2004; Van Mechelen, Bock and De Boeck 2004). Quite a few applications can be accounted for by structural mechanisms that can be formalized in terms of overlapping biclusters, which further also implies overlapping clusterings of the objects and the variables. Indeed, in the patient by symptom case, patients may suffer in general from more than a single syndrome (a phenomenon called syndrome comorbidity) and a symptom may be characteristic for more than a single disease. In the gene by condition case, a gene may belong to several modules, whereas a single condition may trigger several modules as well. In this paper we will further focus on constant (homogeneous) biclusters[2] and on additive approaches. The former implies that every bicluster is assigned a characteristic value or *weight*, with this weight being the reconstructed (constant) data value for the entries in that bicluster (see Madeira and Oliveira 2004). The latter means that, in case of bicluster overlap, the reconstructed data value in the overlap equals the sum of the values associated with the intersecting biclusters. This leads to the additive biclustering model (Both and Gaul 1987, 1985; Gaul and Schader 1996), which was introduced first by De Sarbo (1982) as a 'dual domain' version of GENNCLUS.

1. In some research domains, biclusters are being denoted by the terms two-mode clusters and coclusters. In the present paper we consider these different terms exchangeable.
2. For approaches in which biclusters may be heterogeneous (i.e., non-constant), see, e.g., Lazzeroni and Owen (2002), and Turner, Bailey, and Krzanowski (2005).

In the additive biclustering model, which is presented schematically in Figure 1, the $I \times J$ object by variable real-valued data matrix $\mathbf{X}$ (with entries x_{ij}) is approximated by an $I \times J$ real-valued model matrix $\mathbf{M}$ (with entries m_{ij}) that can be decomposed into binary (0/1) matrices $\mathbf{A}$ and $\mathbf{B}$ of sizes $I \times K$ and $J \times L$, respectively, and a real-valued $K \times L$ matrix $\mathbf{W}$:

$$\mathbf{M} = \mathbf{AWB}', \tag{1}$$

which means that

$$m_{ij} = \sum_{k,l=1}^{K,L} a_{ik} w_{kl} b_{jl}. \tag{2}$$

The columns of $\mathbf{A}$ define K, possibly overlapping, object clusters and therefore $\mathbf{A}$ is called the *object cluster membership matrix*; the entries a_{ik} of this matrix denote whether object i belongs to cluster k ($a_{ik} = 1$) or not ($a_{ik} = 0$). Similarly, the L columns of the *variable cluster membership matrix* $\mathbf{B}$ constitute a, possibly overlapping, clustering of the variables, with the entries b_{jl} denoting whether variable j belongs to cluster l ($b_{jl} = 1$) or not ($b_{jl} = 0$). The non-zero entries w_{kl} of the *weights matrix* $\mathbf{W}$ denote the reconstructed (constant) data value of the *bicluster* that is obtained by the Cartesian product of object cluster k and variable cluster l. Equation (2) then states that the value of object i on variable j is the sum of the weights w_{kl} of the biclusters (i, j) belongs to. It should be noted that the additive biclustering model has two existing models as special subcases: (1) the 'single domain' GENNCLUS model of De Sarbo (1982), which comes down to model (1) with the matrix of weights $\mathbf{W}$ being constrained to be square and symmetric, and (2) the additive box clustering model of Mirkin, Arabie, and Hubert (1995) that can be obtained by constraining $\mathbf{W}$ to be diagonal (and, hence, also square and symmetric). Note that both subcases imply an equal number of object and variable clusters (i.e., $K = L$).

To fit the additive biclustering model in a least squares sense to a data matrix at hand, two alternating least squares (ALS) approaches have been proposed: (1) $PENCLUS$ (Both and Gaul 1987, 1985; Schader and Gaul 1996), and (2) the ALS approach of Baier, Gaul, and Schader (1997), which further will be denoted as *Clusterwise ALS*.[3] In ALS algorithms, which be-

3. Note that other methods for identifying overlapping object and variable clusterings have been proposed by Hartigan (1976), Mirkin et al. (1995), Greenacre (1988), and Eckes and Orlik (1993). These methods, however, do not imply the optimization of an overall (least squares) loss function or do not fit the explicit model structure in (1) to the data; therefore, they will not be further considered in this manuscript (for more information regarding different classes of modeling and optimization techniques for cluster analysis, see Van Mechelen, Bock, and De Boeck 2004). Also the algorithms of De Sarbo (1982) and Mirkin et al. (1995) will not be further considered, because they have been developed for fitting constrained versions of (1) to the data, implying a symmetric and/or diagonal $\mathbf{W}$.

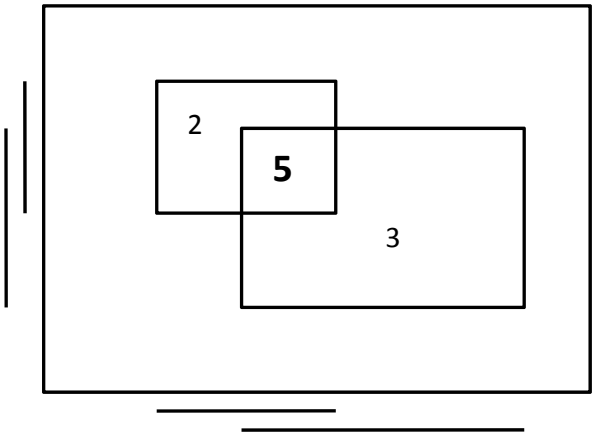

Figure 1. An example of a biclustering model. The outer rectangle is a schematic representation of an object by variable data matrix. The two inner rectangles represent two intersecting biclusters, with their corresponding (overlapping) object and variable clusters indicated with lines next to the data matrix. The biclusters have weights 2 and 3 and the model value of the intersection is the sum of the two bicluster weights.

long to the broader class of block-relaxation algorithms (de Leeuw 1984), the model parameters are grouped (e.g., partitioned) into a number of subsets; during the algorithmic process each subset is conditionally re-estimated in turn while keeping the parameters not belonging to the set in question fixed; this updating procedure is continued until there is no further improvement in the loss function value. Unfortunately, both for $PENCLUS$ and *Clusterwise ALS* some inherent limitations may be identified (see further below), which may interfere with algorithmic performance (in terms of a slower convergence rate and worse optimization and recovery performance). As a way out, in the present paper we will propose a new ALS algorithm to overcome the inherent limitations of $PENCLUS$ and *Clusterwise ALS*. Subsequently, we will conduct an extensive simulation study in which the three algorithms will be compared with regard to optimization performance and recovery of the underlying truth.

The remainder of this paper is organized as follows. In Section 2, we will deal with the associated data analysis, including a discussion of both existing algorithms (and their limitations) and the new ALS algorithm to estimate the additive biclustering model. Subsequently, in Section 3, we will report on a simulation to evaluate the algorithmic performance of the three algorithms. In Section 4, some concluding remarks will be given.

2. Data Analysis

Given an $I \times J$ data matrix $\mathbf{X}$, and a number of underlying object clusters K and variable clusters L, the aim of an additive biclustering analysis is to estimate binary matrices $\mathbf{A}$ and $\mathbf{B}$ and a real-valued matrix $\mathbf{W}$ such that the least squares loss function:

$$F(\mathbf{A}, \mathbf{B}, \mathbf{W}) = \|\mathbf{X} - \mathbf{AWB}'\|_F^2, \tag{3}$$

is minimized, with $\|.\|_F$ denoting the Frobenius norm of a matrix (i.e., the square root of the sum of the squared entries).

In practice, the number of clusters K and L is usually unknown. A common strategy is then to fit a series of additive biclustering models with different numbers of clusters K and L and to subsequently select the final model by means of some model selection heuristic (see, Ceulemans and Kiers 2006; Schepers, Ceulemans, and Van Mechelen 2008; Wilderjans, Ceulemans, and Van Mechelen 2012; Wilderjans, Ceulemans, Van Mechelen, and Depril 2001). Also, one could decide on K and L on the basis of substantive considerations.

For given membership matrices $\mathbf{A}$ and $\mathbf{B}$, the conditionally optimal matrix of weights $\mathbf{W}$ can be found by ordinary least squares regression yielding

$$\mathbf{W} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{XB}(\mathbf{BB}')^{-1}. \tag{4}$$

As a consequence, the solution space of the minimization problem of loss function (3) can be reduced to the finite set of all 2^{IJKL} possible pairs of possible matrices $\mathbf{A}$ and $\mathbf{B}$, together with their corresponding optimal profiles $\mathbf{W}$. This implies that the global optimum of (3) can, in principle, be determined enumeratively. However, when the number of clusters K and/or L increases, this becomes quickly computationally infeasible, and suitable heuristics need to be adopted, like $PENCLUS$ (Section 2.1) and $Clusterwise\ ALS$ (Section 2.2). In Section 2.3, we will present a new ALS algorithm that overcomes the limitations of the former two algorithms.

2.1 $PENCLUS$

In $PENCLUS$ (Both and Gaul 1987, 1985; Gaul and Schader 1996), which is based on a penalty approach, the loss function

$$F^* = F + \rho(BC_A + BC_B) \tag{5}$$

is minimized, without any constraints on $\mathbf{A}$ and $\mathbf{B}$ (i.e., not being restricted to be binary); $BC_A = \sum_{i=1}^{I}\sum_{k=1}^{K}(a_{ik}(1-a_{ik}))^2$ and $BC_B = \sum_{j=1}^{J}\sum_{l=1}^{L}(b_{jl}(1-b_{jl}))^2$ denote the *penalty terms*, and ρ the *penalty parameter*. To

enforce $\mathbf{A}$ and $\mathbf{B}$ to be (close to) binary, resulting in $BC_A = 0$ and $BC_B = 0$, the penalty parameter ρ is increased during the algorithm. In particular, the $PENCLUS$ algorithm consists of the following steps:

1. Put $\rho = 1$ and generate initial estimates for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{W}$ by sampling independently the entries a_{ik} of $\mathbf{A}$ and b_{jl} of $\mathbf{B}$ from $U(0, 1)$, and by computing the corresponding w_{kl} of $\mathbf{W}$ by means of (4).
2. While not all entries of $\mathbf{A}$ and $\mathbf{B}$ are in the intervals $(-.1, .1)$ or $(.9, 1.1)$ do:
   (a) Minimize $F_A^* = F + \rho BC_A$ over $\mathbf{A}$ by means of the method of steepest descent.
   (b) Calculate $\mathbf{W}$ by means of (4).
   (c) Minimize $F_B^* = F + \rho BC_B$ over $\mathbf{B}$ by means of the method of steepest descent.
   (d) Calculate $\mathbf{W}$ by means of (4).
   (e) Put $\rho = 1.05\,\rho$.
3. Round the matrices $\mathbf{A}$ and $\mathbf{B}$ (to binary values) and calculate the corresponding $\mathbf{W}$ by means of (4).

To minimize F_A^* over $\mathbf{A}$ in Step 2a of the $PENCLUS$ algorithm (in Step 2c, F_B^* is minimized over $\mathbf{B}$ analogously), $\mathbf{A}$ is updated until the difference in loss F_A^* between two consecutive matrices $\mathbf{A}$ is smaller than .1. This is done in a stepwise manner by repeatedly executing the expression

$$\mathbf{A} = \mathbf{A} + \frac{\nabla L_A^2}{\|\nabla L_A^2\|_F}, \tag{6}$$

with

$$\nabla L_A^2 = \left[\frac{\partial L^2}{\partial a_{ik}} + \rho \frac{\partial BC_A}{\partial a_{ik}}\right]_{i,k=1}^{I,K}, \tag{7}$$

$$\frac{\partial L^2}{\partial a_{ik}} = -2 \sum_j \left[\left(x_{ij} - \sum_{k'} \sum_l a_{ik'} w_{k'l} b_{jl}\right)\left(\sum_l w_{kl} b_{jl}\right)\right], \tag{8}$$

$$\frac{\partial BC_A}{\partial a_{ik}} = 2(a_{ik} - 3a_{ik}^2 + 2a_{ik}^3). \tag{9}$$

2.2 *Clusterwise ALS*

The *Clusterwise ALS* algorithm (Baier et al. 1997) works as follows:

1. Obtain initial parameter estimates by sampling the entries a_{ik} of $\mathbf{A}$ and b_{jl} of $\mathbf{B}$ independently from a Bernoulli distribution with parameter $\pi = .5$, and calculating the corresponding $\mathbf{W}$ by means of (4).

2. Update $\mathbf{A}$ column by column by doing for $k = 1, \ldots, K$ the following:
   (a) Calculate the residual data
   $$x_{ij}^{(k)} = x_{ij} - \sum_{k' \neq k, l} a_{ik'} w_{k'l} b_{jl} \tag{10}$$
   (b) For $i = 1, \ldots, I$, put
   $$a_{ik} = \begin{cases} 1 & \text{if } \sum_j \left(x_{ij}^{(k)} - \sum_l w_{kl} b_{jl} \right)^2 < \sum_j \left(x_{ij}^{(k)} \right)^2 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$
   (c) Calculate the k-th row $(w_{k1}, \ldots, w_{kL})$ of $\mathbf{W}$ by minimizing the residual loss $\sum_{i,j} \left(x_{ij}^{(k)} - \sum_l a_{ik} b_{jl} w_{kl} \right)^2$ by means of ordinary least squares regression.
3. Update the columns of $\mathbf{B}$ and $\mathbf{W}$ analogously as in Step 2.
4. Repeat steps 2 and 3 until there is no more decrease in the loss function.

2.3 *Full clustering ALS*

Both $PENCLUS$ and $Clusterwise\ ALS$ have some inherent limitations. In particular, the $PENCLUS$ algorithm may not yield conditionally optimal matrices $\mathbf{A}$ ($\mathbf{B}$) given some $\mathbf{W}$ and $\mathbf{B}$ ($\mathbf{A}$), because the conditional re-estimations of $\mathbf{A}$ ($\mathbf{B}$) are based on a single steepest descent move starting from one initial solution only. Regarding $Clusterwise\ ALS$, it may, in general for ALS algorithms, be recommended to group together these parameters that are highly dependent (de Leeuw 1994; Wilderjans, Depril, and Van Mechelen 2012). Yet, $Clusterwise\ ALS$ has in this regard two drawbacks. First, the strong dependencies among the continuous bicluster weights are neglected to a large extent, because only weights that belong to the same row or column of $\mathbf{W}$ are simultaneously updated. Second, by re-estimating the binary membership parameters in $\mathbf{A}$ and $\mathbf{B}$ cluster per cluster (i.e., column-wise), the within-row dependencies in these matrices are largely ignored.

To overcome the drawbacks mentioned above, the following two improvements to $Clusterwise\ ALS$ may be proposed: (1) Update the bicluster weights in $\mathbf{W}$ as a whole instead of row-wise or column-wise, and (2) re-estimate the object (variable) cluster membership parameters in $\mathbf{A}$ ($\mathbf{B}$) as a whole instead of cluster-wise (i.e., column-wise). Otherwise, for the latter,

a separability property of the loss function (3) may be invoked (Chaturvedi and Carroll 1994). Indeed, the loss function can be rewritten as

$$\begin{aligned} F &= \sum_j \left(x_{1j} - \sum_{k,l} a_{1k} w_{kl} b_{jl} \right)^2 \\ &+ \cdots \\ &+ \sum_j \left(x_{Ij} - \sum_{k,l} a_{Ik} w_{kl} b_{jl} \right)^2, \end{aligned} \tag{12}$$

which implies that the contribution of the i-th row of $\mathbf{A}$ to the value of F can be separated from the contribution of the other rows of $\mathbf{A}$. As a consequence, $\mathbf{A}$ can be updated row-wise (i.e., row by row). A similar separability property holds for the rows of $\mathbf{B}$. Note that for the columns of $\mathbf{A}$ and $\mathbf{B}$ no such separability property holds (i.e., cluster contributions to F cannot be separated from one another).

Based on these modifications, a new $Full\ clustering\ ALS$ algorithm is obtained, which proceeds with the following steps:

1. Construct initial parameter estimates by generating the entries a_{ik} of $\mathbf{A}$ and b_{jl} of $\mathbf{B}$ independently from a Bernoulli distribution with parameter $\pi = .5$, and calculating the corresponding weights matrix $\mathbf{W}$ by means of (4).
2. Calculate, conditional upon the current estimates of $\mathbf{B}$ and $\mathbf{W}$, the optimal membership matrix $\mathbf{A}$, which can be done row-wise by, for each row, evaluating all 2^K possible row patterns enumeratively.
3. Re-estimate $\mathbf{W}$, conditional on $\mathbf{A}$ and $\mathbf{B}$, by means of (4).
4. Compute, conditional upon the current estimates of $\mathbf{A}$ and $\mathbf{W}$, the optimal membership matrix $\mathbf{B}$, which again can be performed row-wise.
5. Update $\mathbf{W}$, conditional on $\mathbf{A}$ and $\mathbf{B}$, by means of (4).
6. Repeat steps 2, 3, 4, and 5 until there is no more decrease in the loss function.

It must be noted that a $Full\ clustering\ ALS$ run (similar to a $PENCLUS$ and a $Clusterwise\ ALS$ run) may strongly depend on the starting values that have been used (i.e., the initial $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{W}$ obtained in the first step of each algorithm). As a consequence, the algorithms may end up in a local rather than the global solution for loss function (3). To resolve this issue, a multi-start strategy may be advised, which consists of running the algorithm multiple times, each time with different initial values for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{W}$ (i.e., first step of each algorithm), and retaining the solution yielding

the lowest loss function value (3) as the final result. Note that each multistart involves an iterative procedure (i.e., repeating different times step 2 of $PENCLUS$, steps 2 and 3 of $Clusterwise\ ALS$, and steps 2 to 5 of $Full\ clustering\ ALS$) that is continued until convergence.

3. Simulation Study

3.1 Introduction

In the previous sections, three algorithms were presented to estimate the best fitting additive biclustering model for a given data set. In this section, we will present a simulation study to evaluate the performance of these algorithms. In this regard, we are interested in two aspects of algorithmic performance: *goodness-of-fit* and *goodness-of-recovery*. With regard to *goodness-of-fit*, we will examine whether an algorithm finds the global optimum of the loss function. Concerning *goodness-of-recovery*, we will investigate to which extent each algorithm succeeds in recovering the true structure underlying a given data set. Algorithmic performance will be evaluated on a global level and as a function of data characteristics. Furthermore, it will be examined both from the viewpoint of the algorithms as a whole, and from the viewpoint of algorithmic differences.

In the next subsections we will outline the design of the simulation study (3.2) and the specific evaluation criteria (3.3). In Subsection 3.4 the results will be presented and discussed. A summary of the results will be presented in Subsection 3.5.

3.2 Design

To generate data sets $\mathbf{X}$ of size $I \times J$, we will independently generate true matrices $\mathbf{A}^{true}$, $\mathbf{B}^{true}$, $\mathbf{W}^{true}$ and noise $\mathbf{E}$. The rows of $\mathbf{A}^{true}$ ($\mathbf{B}^{true}$) will be independently drawn from a multinomial distribution on all possible 2^K (2^L) binary row patterns (with a probability of .05 for the zero pattern and with the probabilities for row patterns with a single 1 put equal to one another). The entries of $\mathbf{W}^{true}$ will be independently drawn from a normal distribution $N(0, \sigma^2_{\mathbf{W}})$ with $\sigma^2_{\mathbf{W}}$ being set equal to 36.[4] The noise entries of $\mathbf{E}$ will be independently drawn from a normal distribution $N(0, \sigma^2_{noise})$, with σ^2_{noise} depending on the desired amount of noise ε in the data (see further). A data set $\mathbf{X}$ will then be obtained as $\mathbf{X} = \mathbf{T} + \mathbf{E}$, with $\mathbf{T} = \mathbf{A}^{true}\ \mathbf{W}^{true}\ (\mathbf{B}^{true})'$ and $\mathbf{T}$ being the true underlying clustering model.

4. In a pilot study it appeared that this value for $\sigma^2_{\mathbf{W}}$, which determines the dispersion of the $\mathbf{W}^{true}$ values, ensures the generated biclusters being (separated far enough from each other to be) distinguishable. To understand this, note that in the special case of a bipartitioning (i.e., the row and column clusterings are both a partitioning instead of an overlapping clustering), setting $\sigma^2_{\mathbf{W}}$ to zero (or to a too small value) would yield (almost) identical biclusters that are not distinguishable from each other.

On the level of the data generation, the following factors were manipulated.

- *Data shape:* The data shape of $\mathbf{X}$ is defined in terms of the ratio I/J between the number of objects and the number of variables and will take three different values: $1/4$, 1, and 4. The total number of entries in $\mathbf{X}$ is $4,096$, implying three different values for $I \times J$: 32×128, 64×64, 128×32.
- *The number of object clusters K:* 2, 3, 4.
- *The number of variable clusters L:* 2, 3, 4.
- *The amount of object cluster overlap:* This factor is defined as the probability of an object belonging to more than one object cluster and it is put equal to 25%, 50%, or 75%; for this purpose, the multinomial probabilities of all row patterns in $\mathbf{A}^{true}$ that contain more than a single 1 were put equal to one another with a total equal to the percentage in question.
- *The amount of variable cluster overlap:* This factor is defined as the probability of a variable belonging to more than one variable cluster and it is put equal to 25%, 50%, or 75%; for this purpose, the multinomial probabilities of all row patterns in $\mathbf{B}^{true}$ that contain more than a single 1 were put equal to one another with a total equal to the percentage in question.
- *The amount of noise ε:* This factor is defined as the proportion ε of the total variance in the data $\mathbf{X}$ accounted for by $\mathbf{E}$; ε will either be 0, .15, .30, .45, or .60. We set $\sigma^2_{noise} = \frac{S^2_{\mathbf{T}}}{S^2_{\mathbf{E}}}\frac{\varepsilon}{(1-\varepsilon)}$, with $S^2_{\mathbf{T}}$ and $S^2_{\mathbf{E}}$ denoting the variance of all the elements in $\mathbf{T}$ and $\mathbf{E}$, respectively.

All design factors were fully crossed, which yields 3 (*Data shape*) $\times$ 3 (*Number of object clusters*) $\times$ 3 (*Number of variable clusters*) $\times$ 3 (*Amount of object cluster overlap*) $\times$ 3 (*Amount of variable cluster overlap*) $\times$ 5 (*Amount of noise*) = $1,215$ combinations; for each combination, 10 replicates were generated, yielding $12,150$ simulated data sets in total.

Each simulated data set $\mathbf{X}$ was analyzed with $PENCLUS$, $Cluster-wise\ ALS$, and $Full\ clustering\ ALS$ (resulting in estimates $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{W}}$ for each algorithm), which yields an algorithmic factor with three levels in the design. A multi-start procedure was applied for each algorithm with each algorithm given a fixed amount of computation time of 5 minutes. Regarding computation time, one should know that the three algorithms were implemented in MATLAB-code (version 7.11.0.584, R2010b) and run on a supercomputer consisting of INTEL XEON L5420 processors with a clock frequency of 2.5 GHz and with 8 GB RAM. In a pilot study, it was demonstrated that running each algorithm for 5 minutes was long enough to arrive

at a high quality solution (in terms of minimizing the loss function). For the analysis with the $12,150$ data sets, during these 5 minutes, on average, $11,311$, $2,565$, and 178 (fully converged) multi-start runs have been performed for $Full\ clustering\ ALS$, $Clusterwise\ ALS$, and $PENCLUS$, respectively, implying $Full\ Clustering\ ALS$ runs being the fastest and $PENCLUS$ runs the slowest.

3.3 Evaluation Criteria

3.3.1 Minimization of the Loss Function

For each data set we want to determine for each algorithm whether it reached the global optimum of the loss function. However, when error has been added to $\mathbf{T}$, this global optimum is unknown. Therefore, we introduce the concept of *proxy* or *pseudo-optimum*. This proxy is determined for each data set separately and acts as an approximation of the global optimum. For each data set we will then determine whether or not (the best run for) each algorithm reached the proxy.

In particular, the proxy for each data set is determined as follows:

1. First of all, an *upper bound* (UB) on the loss value is determined. An obvious candidate upper bound is the loss of the true underlying clustering model $\mathbf{T} = \mathbf{A}^{true}\ \mathbf{W}^{true}\ (\mathbf{B}^{true})'$. However, we will use a sharper upper bound by running the $Clusterwise\ ALS$ algorithm and the $Full\ clustering\ ALS$ algorithm seeded with the true memberships $\mathbf{A}^{true}$ and $\mathbf{B}^{true}$; the best of the two resulting loss values will be taken as the upper bound UB.
2. All three algorithms are run on the data set which yields three loss values F_1, F_2, F_3.
3. The value of the proxy is then given by $\min(\text{UB}, F_1, F_2, F_3)$.

Note that either no, one, or several algorithms can reach the proxy for a given data set at hand. Further, we also computed the following *goodness-of-fit* (GOF) statistic:

$$1 - \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{ij} - m_{ij})^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{ij} - \bar{x})^2}, \tag{13}$$

with x_{ij} and m_{ij} denoting the values in the generated data $\mathbf{X}$ and the matrix $\mathbf{M} = \hat{\mathbf{A}}\hat{\mathbf{W}}\hat{\mathbf{B}}'$, respectively, and $\bar{x}$ the average value of $\mathbf{X}$. GOF takes values in the interval $(-\infty, 1]$, with a value of 1 implying a perfect reconstruction of the data (i.e., $\mathbf{M}$ equals $\mathbf{X}$).

3.3.2 Recovery of the Underlying Row and Column Clusters and Bicluster Weights

Cluster recovery. To evaluate the quality of the clustering $\hat{\mathbf{A}}$ found by each algorithm, we will calculate the Omega coefficient $\omega_{\mathbf{A}}$ (Collins and Dent 1988) between the true overlapping clustering (i.e., $\mathbf{A}^{true}$) and the overlapping clustering obtained by the algorithm in question (i.e., $\hat{\mathbf{A}}$). The $\omega_{\mathbf{A}}$ coefficient equals one when perfect recovery of the underlying overlapping clustering is encountered and zero when recovery is at chance level. $\omega_{\mathbf{B}}$, that is the *goodness-of-recovery* of clustering $\hat{\mathbf{B}}$, is defined analogously. Note that the ω coefficient is invariant under a permutation of the clusters (i.e., columns of $\hat{\mathbf{A}}$ or $\hat{\mathbf{B}}$).

Recovery of the (bicluster) weights. To evaluate the quality of the estimated profiles $\hat{\mathbf{W}}$, we will calculate the expression

$$1 - \frac{\sum_{k=1}^{K}\sum_{l=1}^{L}(\hat{w}_{kl} - w_{kl}^{true})^2}{\sum_{k=1}^{K}\sum_{l=1}^{L}(w_{kl}^{true} - \bar{w}^{true})^2}, \tag{14}$$

with $\bar{w}^{true}$ the average value of $\mathbf{W}^{true}$; this expression is a rescaled sum of the Euclidean distances between the corresponding profiles. To take the permutational freedom of the order of clusters into account, we will define the *goodness-of-recovery* of the weights matrix $\mathbf{W}$ (GOW) as the minimum value of (14) over all possible row and column permutations of $\hat{\mathbf{W}}$. GOW takes values in the interval $(-\infty, 1]$, with a value of 1 meaning perfect recovery.

3.4 Results

3.4.1 Minimization of the Loss Function

In 78.27% of the data sets the best run for one or more of the algorithms reached the proxy, and the mean GOF value equals .6904. Further, in Table 1, the percentage of data sets that hit the proxy and the mean GOF value is displayed for (the best run of) each algorithm separately; from this table, one can see that the $Full\ clustering\ ALS$ algorithm clearly outperforms the other two algorithmic strategies with respect to minimizing the loss function.

Furthermore, considerable differences were found between the different cells of the design of the study. To capture these differences, we calculated for each cell and for each individual algorithm the percentage of data sets on which the proxy had been reached. We then analyzed both the percentage of data sets that hit the proxy and GOF by means of a repeated measures analysis of variance with all data characteristics and the algorith-

Table 1. Average result, across all data sets, for each algorithm on each evaluation criterion.

			goodness-of-recovery		
Algorithm	Proxy reached (% data sets)	*goodness-of-fit* (GOF)	Row clusters ($\omega_{\mathbf{A}}$)	Column clusters ($\omega_{\mathbf{B}}$)	Weight matrix (GOW)
$PENCLUS$	14.10	.6805	.6911	.6866	.7885
$Clusterwise\ ALS$	14.93	.6882	.6874	.6862	.8091
$Full\ clustering\ ALS$	76.89	.7027	.8522	.8491	.9350

mic factor as independent variables. Regarding the percentage of data sets that hit the proxy, for which we only have one observation per algorithm in each cell, we fitted a reduced model by omitting the highest order interaction term. Below, we will only focus on effects with an effect size (η^2_G) of .04 or higher. For the minimization performance, these are given in the upper two panels of Table 2.

As can be seen in this table, both for the percentage of data sets that reach the proxy and GOF, most differences can be explained by the main effect of the algorithmic factor (see Table 1) and the amount of noise on the data: The more noise in the data, the more difficult it becomes for an algorithm to optimize the loss function. The latter effect, however, is less pronounced for $Full\ clustering\ ALS$ than for $PENCLUS$ and $Clusterwise\ ALS$. Further, an interaction between the number of row clusters K and the number of column clusters L was observed. In Table 3, in which this interaction is presented, it can be seen that minimization performance decreases when the maximum of the number of row and column clusters increases.

3.4.2 Recovery of the Underlying Clusters and (Bicluster) Weights

The overall average values for the recovery measures $\omega_{\mathbf{A}}$, $\omega_{\mathbf{B}}$, and GOW are .7436, .7406, and .8442, respectively. The averages for each algorithm separately can be found in Table 1. In Figure 2, a boxplot of the row cluster recovery $\omega_{\mathbf{A}}$ is presented for each algorithm (the boxplots for $\omega_{\mathbf{B}}$ and GOW look very similar). From these tables, it can be seen that the introduced $Full\ clustering\ ALS$ algorithm outperforms the other two algorithmic strategies. Furthermore, considerable differences between the different cells of the simulation design were observed. To examine those in more detail, a repeated measures analysis of variance was conducted on each recovery criterion separately with the data factors of the simulation design and the algorithmic factor acting as the independent variables. Significant effects with $\eta^2_G \geq .04$ are tabulated in the lower three panels of Table 2. For $\omega_{\mathbf{A}}$, the ANOVA table is displayed in Table 4, with only including the most important main effects and interactions.

Table 2. Most important effects and their effect size ($\eta^2_G \geq .04$) in repeated measures analyses of variance for each evaluation criterion. Also the largest effect with $\eta^2_G < .04$ is reported (between parentheses).

Criterion	Effect	η^2_G
Reaching proxy	Algorithm	.48
	Amount of noise ε	.06
	Number of row clusters K * Number of column clusters L	.06
	Number of row clusters K	.04
	Number of column clusters L	.04
	Algorithm * Amount of noise ε	.04
	(Algorithm * Number of row clusters K * Number of column clusters L * Amount of noise ε	.03)
goodness-of-fit (GOF)	Amount of noise ε	.98
	Algorithm	.10
	Algorithm * Amount of noise ε	.06
	(Algorithm * Number of row clusters K * Number of column clusters L * Amount of noise ε	.03)
Recovery of $\mathbf{A}$ ($\omega_{\mathbf{A}}$)	Number of row clusters K	.28
	Amount of noise ε	.17
	Algorithm	.12
	Number of column clusters L	.09
	Algorithm * Number of row clusters K	.05
	Data shape	.04
	(Algorithm * Amount of noise ε	.02)
Recovery of $\mathbf{B}$ ($\omega_{\mathbf{B}}$)	Number of column clusters L	.28
	Amount of noise ε	.16
	Algorithm	.12
	Number of row clusters K	.09
	Algorithm * Number of column clusters L	.05
	(Algorithm * Amount of noise ε	.02)
Recovery of weights $\mathbf{W}$ (GOW)	Algorithm	.05
	(Algorithm * Amount of noise ε	.02)

Table 3. Percentage of data sets that reached the proxy, aggregated over the three different algorithms, for each combination of a number of row clusters K and a number of column clusters L.

Row clusters	Column clusters		
	$L = 2$	$L = 3$	$L = 4$
$K = 2$	60.7%	40.8%	23.3%
$K = 3$	40.0%	44.2%	27.5%
$K = 4$	23.8%	28.5%	28.9%

Table 4. ANOVA table for recovery of $\mathbf{A}$ ($\omega_{\mathbf{A}}$) with most important main and interaction effects, all effects being significant at $\alpha = .05$ level (p<.0001).

Criterion	SS	df	MS	F	η^2_G
Number of row clusters K	614.04	2	307.02	3659.55	.28
Amount of noise ε	312.84	4	78.21	932.25	.17
Algorithm	215.28	2	107.64	3655.81	.12
Number of column clusters L	153.52	2	76.76	914.97	.09
Algorithm * Number of row clusters K	90.68	4	22.67	769.91	.05
Data shape	58.21	2	29.11	346.92	.04
Algorithm * Amount of noise ε	37.82	8	4.73	160.54	.02
Error(Algorithm)	643.93	21870	0.03		
Error	917.39	10935	0.08		

From these tables it can be seen that there is a main effect of the algorithmic factor on all three recovery measures, with $Full\ Clustering\ ALS$ clearly outperforming the other two algorithms. Furthermore, the amount of noise has an influence on both cluster recovery measures, with higher amounts of noise yielding worse recoveries. The number of row clusters K and the number of column clusters L have an influence on both row and column cluster recovery, with a higher number of clusters yielding worse recoveries. Moreover, this effect is more pronounced for $PENCLUS$ and $Clusterwise\ ALS$ than for $Full\ Clustering\ ALS$.

3.4.3 Equal Number of Multi-Start Runs

The question could be raised whether our comparison of the algorithms is a fair one as the algorithms were given an equal amount of computation time. Since the amount of computational effort required to execute one iteration differs across algorithms, this approach has the benefit that each algorithm did more or less the same amount of algorithmic effort to obtain its final result. However, the amount of computation time depends on the actual implementation of the algorithm. Therefore, we also analyzed the best result of the first 40 starts of each algorithm, with 40 being the minimum number of starts taken by an algorithm (i.e., $PENCLUS$) over all $12,150$ data sets of the simulation study. For each evaluation criterion we found similar results as the ones reported in Section 3.4: Again, the $Full\ clustering\ ALS$ algorithm outperformed $PENCLUS$ and $Clusterwise\ ALS$ with respect to optimization and recovery performance, and, again, negative influences of an increasing amount of noise and an increasing number of clusters K and L were found.

3.4.4 Performance of $Full\ Clustering\ ALS$

In the simulation results, $Full\ clustering\ ALS$ clearly outperforms the other two ALS algorithms. However, the algorithmic performance of

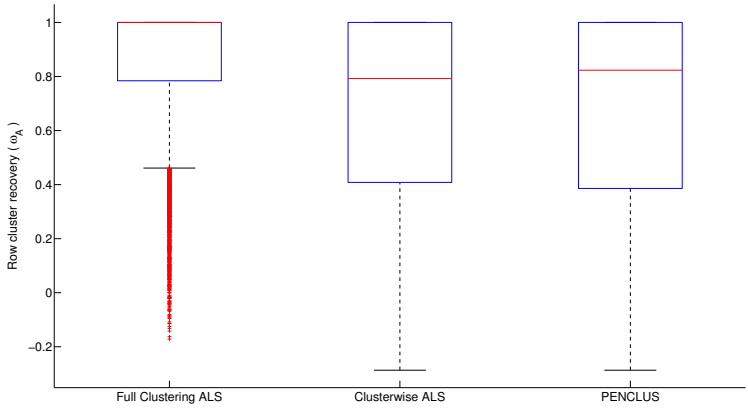

Figure 2. Boxplot of the row cluster recovery $\omega_{\mathbf{A}}$ for *Full clustering ALS*, *Clusterwise ALS*, and *PENCLUS*.

Full clustering ALS fluctuates as a function of the data characteristics. In particular, an increasing amount of noise and a larger number of row or column clusters have a negative influence on algorithmic performance. Both effects are fairly obvious, since the former implies less coherent clusters and the latter implies a larger solution space to the minimization of (3). When studying the performance of *Full clustering ALS* in conditions with problematic values of the noise and number of clusters (i.e., 60% of noise, 4 row clusters K and 4 column clusters L), and limiting ourselves to data sets of size 32×128 (which yields nine cells in total), we see that the proxy is found by the algorithm on 16% of the data sets only, whereas the average $\omega_{\mathbf{A}}$, $\omega_{\mathbf{B}}$, *GOW*, and *GOF* were .710, .472, .790, and .420, respectively. This result, first of all, indicates that especially the minimization aspect of algorithmic performance is troublesome in case of more problematic data characteristics, whereas under such circumstances recovery performance still remains rather satisfactory. It suggests the presence of many local optima in the additive biclustering optimization problem, which may remind one of the somewhat similar (and much simpler) discrete optimization problem in the K-means case, for which the problem of local optima has been well documented (Hand and Krzanowski 2005; Steinley and Brusco 2007). In the case of the additive biclustering model, with its considerably larger optimization space, obviously, the local optima problem is even much more challenging. The question then may be raised of how to deal with this problem. An obvious solution could be to increase the number of multi-start runs. Indeed, if we give the *Full clustering ALS* algorithm ten times more time (i.e., a total of 50 minutes), which increases the number of multi-start runs by a factor of ten, all performance measures increase substantially.

3.5 Summary of the Results

For each evaluation criterion, a sizeable main effect of the algorithmic factor was found, while no large (disordinal) interactions (see Schepers and Van Mechelen 2011) of this algorithmic factor with other design factors were observed. In particular, the novel $Full\ clustering\ ALS$ algorithm performs consistently better than the other strategies, with the algorithmic performance in general decreasing when the data contain a large amount of noise and the number of underlying object or variable clusters is large.

4. Conclusion

In this paper we presented a new $Full\ clustering\ ALS$ algorithm to fit additive biclustering models to a given data set. The new algorithm, which incorporates general recommendations regarding optimally designing ALS algorithms, calculates conditionally optimal estimates for the membership matrices and the core matrix of the model in their entirety. In an extensive simulation study, the new algorithm has been shown to outperform two previously proposed algorithms for the same model: $PENCLUS$ (Both and Gaul 1985) and $Clusterwise\ ALS$ (Baier et al. 1997). As a consequence, the use of the $Full\ clustering\ ALS$ algorithm can be recommended in statistical practice[5]. The presence of local optima, however, is still a bottleneck. As a way out, it may be advisable to use a large amount of multi-start runs, especially when fitting models involving a large number of row or column clusters or when a large amount of noise is present in the data.

References

BAIER, D., GAUL, W., and SCHADER, M. (1997), "Two-Mode Overlapping Clustering with Applications to Simultaneous Benefit Segmentation and Market Structuring", in *Classification and Knowledge Organization*, eds. R. Klar, and K. Opitz, Berlin, Germany: Springer, pp. 557–566.

BOTH, M., and GAUL, W. (1987), "Ein Vergleich Zweimodaler Clusteranalyseverfahren," *Methods of Operations Research, 57*, 593–605.

BOTH, M., and GAUL, W. (1985), "PENCLUS: Penalty Clustering for Marketing Applications," Discussion Paper No. 82, Institution of Decision Theory and Operations Research, University of Karlsruhe.

5. Also for large gene by tissue data sets containing gene expression levels, the $Full\ clustering\ ALS$ algorithm may be practical. In particular, applying 100 multi-start runs of the $Full\ clustering\ ALS$ algorithm with four row and four column clusters to two real data sets concerning colon cancer (i.e., $2,000$ genes by 62 tissues) and cancer (i.e., $17,334$ genes by 69 tissues) takes, on average, 6 minutes and 3 hours for the first and second data set, respectively. Note that, when many multi-start runs are needed, it is possible to use a parallel implementation of the algorithm (i.e., more than hundred multi-start runs may be performed simultaneously).

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.

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.

COLLINS, L.M., and DENT, C.W. (1988), "Omega: A General Formulation of the Rand Index of Cluster Recovery Suitable for Non-Disjoint Solutions," *Multivariate Behavioral Research, 23*, 231–242.

DE LEEUW, J. (1994), "Block-Relaxation Algorithms in Statistics", in *Information Systems and Data Analysis*, eds. H.-H. Bock, W. Lenski, and M.M. Richter, Berlin: Springer-Verlag, pp. 308–325.

DE SARBO, W.S. (1982), "Gennclus: New Models for General Nonhierarchical Clustering Analysis," *Psychometrika, 47*, 449–475.

ECKES, T., and ORLIK, P. (1993), "An Error Variance Approach to Two-Mode Hierarchical Clustering," *Journal of Classification, 10*, 51–74.

FAITH, J.J., HAYETE, B., JOSHUA, T., THADEN, J.T., MONGO, I.M., WIERZBOWSKI, J., COTTAREL, G., KASIF, S., COLLINS, J.J., and GARDNER, T.S. (2007), "Large-Scale Mapping and Validation of Escherichia Coli Transcriptional Regulation from a Compendium of Expression Profiles," *PLoS Biology, 5(1)*, 54–66.

GARA, M., ROSENBERG, S., and GOLDBERG, L. (1992), "DSM-IIIR as a Taxonomy: A Cluster Analysis of Diagnoses and Symptoms," *Journal of Nervous and Mental Disease, 180*, 11–19.

GASCH, A.P., SPELLMAN, P.T., KAO, C.M., CARMEL-HAREL, O., EISEN, M.B., STORZ, G., BOTSTEIN, D., and BROWN, P.O. (2000), "Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes," *Molecular Biology of the Cell, 11*, 4241–4257.

GAUL, W., and SCHADER, M. (1996), "A New Algorithm for Two-Mode Clustering", in *Data Analysis and Information Systems: Statistical and Computational Approaches*, eds. H.-H. Bock, and W. Polasek, Berlin, Germany: Springer, pp. 15–23.

GREENACRE, M.J. (1988), "Clustering the Rows and Columns of a Contingency Table," *Journal of Classification, 5*, 39–51.

HAND, D., and KRZANOWSKI, W. (2005), "Optimizing K-means Clustering Results with Standard Software Packages," *Computational Statistics and Data Analysis, 49*, 969–973.

HARTIGAN, J.A. (1976), "Modal Blocks in Dentition of West Coast Mammals," *Systematic Zoology, 25*, 149–160.

LAZZERONI, L., and OWEN, A. (2002), "Plaid Models for Gene Expression Data," *Statistica Sinica, 12*, 61–86.

MADEIRA, S.C., and OLIVEIRA, A.L. (2004), "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1*, 24–45.

MEZZICH, J.E., and SOLOMON, H. (1980), *Taxonomy and Behavioral Science: Comparative Performance of Grouping Methods*, London: Academic Press.

MIRKIN, B., ARABIE, P., and HUBERT, L.J. (1995), "Additive Two-Mode Clustering: The Error-Variance Approach Revisited?," *Journal of Classification, 12*, 243–263.

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. (2011), "A Two-Mode Clustering Method to Capture the Nature of the Dominant Interaction Pattern in Large Profile Data Matrices," *Psychological Methods, 16*, 361–371.

SEGAL, E., SHAPIRA, M., REGEV, A., PE'ER, D., BOTSTEIN, D., KOLLER, D., and FRIEDMAN, N. (2003), "Module Networks: Identifying Regulatory Modules and Their Condition-Specific Regulators from Gene Expression Data," *Nature Genetics, 34*, 166–176.

SPELLMAN, P.T., SHERLOCK, G., ZHANG, M.Q., IYER, V.R., ANDERS, K., EISEN, M.B., BROWN, P.O., BOTSTEIN, D., and FUTCHER, B. (1998), "Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization," *Molecular Biology of the Cell, 9*, 3273–3297.

STEINLEY, D., and BRUSCO, M.J. (2007), "Initializing K-means Batch Clustering: A Critical Evaluation of Several Techniques," *Journal of Classification, 24*, 99–121.

TURNER, H., BAILEY, T., and KRZANOWSKI, W. (2005), "Improved Biclustering of Microarray Data Demonstrated Through Systematic Performance Tests," *Computational Statistics and Data Analysis, 48*, 235–254.

VAN MECHELEN, I., BOCK, H.-H., and DE BOECK, P. (2004), "Two-Mode Clustering Methods: A Structured Overview," *Statistical Methods in Medical Research, 13*, 363–394.

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., and DE BOECK, P. (1990), "Projection of a Binary Criterion into a Model of Hierarchical Classes," *Psychometrika, 55*, 677–694.

WILDERJANS, T. F., CEULEMANS, E., and VAN MECHELEN, I. (2008), "The CHIC Model: A Global Model for Coupled Binary Data," *Psychometrika, 73*, 729–751.

WILDERJANS, T. F., CEULEMANS, E., and VAN MECHELEN, I. (2012), "The SIMCLAS Model: Simultaneous Analysis of Coupled Binary Data Matrices with Noise Heterogeneity Between and Within Data Blocks," *Psychometrika, 77,* 724–740.

WILDERJANS, T. F., CEULEMANS, E., VAN MECHELEN, I., and DEPRIL, D. (2011), "ADPROCLUS: A Graphical User Interface for Fitting Additive Profile Clustering Models to Object by Variable Data Matrices," *Behavior Research Methods, 43*, 56–65.

WILDERJANS, T. F., DEPRIL, D., and VAN MECHELEN, I. (2012), "Block-Relaxation Approaches for Fitting the INDCLUS Model," *Journal of Classification, 29,* 277–296.