Journal of Statistical Software

May 2012, Volume 48, Issue 5. http://www.jstatsoft.org/

New Developments in Mokken Scale Analysis in R

L. Andries van der Ark
Tilburg University

Abstract

Mokken (1971) developed a scaling procedure for both dichotomous and polytomous items that was later coined Mokken scale analysis (MSA). MSA has been developed ever since, and the developments until 2000 have been implemented in the software package **MSP** (Molenaar and Sijtsma 2000) and the R package **mokken** (Van der Ark 2007). This paper describes the new developments in MSA since 2000 that have been implemented in **mokken** since its first release in 2007. These new developments pertain to invariant item ordering, a new automated item selection procedure based on a genetic algorithm, inclusion of reliability coefficients, and the computation of standard errors for the scalability coefficients. We demonstrate all new applications using data obtained with a transitive reasoning test and a personality test.

Keywords: nonparametric item response theory, Mokken scale analysis, invariant item ordering, R.

1. Introduction

Mokken scale analysis (MSA; Mokken 1971; Sijtsma and Molenaar 2002) is a scaling technique for ordinal data, and mainly used for scaling test[1] and questionnaire data. Having approximately 3,810 hits on Google Scholar, MSA is a rather popular scaling method, though not as popular as Rasch analysis (13,900 hits) or factor analysis (1,380,000 hits). MSA is based on nonparametric item response theory (IRT) models and consists of two parts: (1) an automated item selection procedure (AISP), which partitions a set of ordinal variables (from here on called items) into scales (called Mokken scales) and possibly leaving some items unselected, and (2) methods to investigate the goodness-of-fit of nonparametric IRT models for each of the Mokken scales. For a thorough discussion of nonparametric IRT, MSA, and data analysis strategies we refer to the following literature: Nonparametric IRT and MSA for (ordinal)

[1] The word 'test' is used for both a psychological or educational test and a statistical test. When this leads to confusion, the adjective 'psychological' or 'statistical' is added.

dichotomous item scores were developed by Mokken (1971; also see Mokken and Lewis, 1982) and extended to polytomous items scores by Molenaar (1991, 1997). Sijtsma and Molenaar (2002) gave an overview of nonparametric IRT and MSA, and provided many references and examples. Meijer and Baneke (2004) demonstrated how MSA can be used preliminary to parametric IRT.

The most popular software for MSA is the commercial package **MSP** Version 5.0 (Molenaar and Sijtsma 2000). It is user-friendly software that allows researchers and test developers to apply most methods and procedures that have been developed within MSA between 1971 and 2000. **MSP** faces two potential problems because no new versions have appeared since 2000 and no new versions are expected in the near future. First, there is a risk that the program may no longer run on future operating systems and, second, new research results on MSA can no longer be implemented in **MSP**.

The aim of the R package (R Development Core Team 2012) **mokken** (Van der Ark 2007, 2010b) was to offer all methods and procedures available in **MSP** as free and open source software (licensed under GPL) that would run irrespective of the operating system, and to provide a platform for new developments in MSA. The first version of **mokken** was a free version of **MSP** but over the years procedures and methods resulting from new results in MSA were embedded. Also the availability of the package generated new research. Because MSA is now embedded in R, it is much easier to compare MSA with other scaling procedures (e.g., Brusco, Koehn, and Steinley 2011; Smits, Timmermans, and Meijer 2011; Straat, Van der Ark, and Sijtsma 2011). This paper discusses new developments in Mokken scale analysis (MSA) and their implementation in **mokken**. The new developments are classified into four categories: Investigating invariant item ordering, a new algorithm for the AISP, computation of test-score reliability, and computation of standard errors of scalability coefficients.

The remainder of the paper is organized as follows: Section 2 briefly discusses nonparametric IRT models and Mokken scale analysis; Section 3 discusses the new developments and the corresponding code for **mokken**; Section 4 demonstrates MSA by applying the functions in **mokken** to data obtained using a transitive reasoning test and a personality test; and Section 5 is the discussion.

2. A summary of nonparametric IRT and MSA

2.1. Nonparametric IRT models

Suppose a test or a questionnaire contains a set of items which are numbered $1, \ldots, J$ and indexed by j. For convenience, but without loss of generality, suppose that each item has $m+1$ ordered answer categories. Let X_j denote the score on item j with realization $x_j = 0, 1, \ldots, m$; Scores $X_1, \ldots, X_J$ are referred to as *item scores.* If $m = 1$ the item is called dichotomous; if $m > 1$ the item is called polytomous. The *test score* is defined as $X_+ = \sum_{j=1}^{J} X_j$. In IRT it is assumed that a latent trait θ triggers the item responses. It is also assumed that the ordering of the scores of each item reflects the hypothesized ordering on θ. Expression $X_j \geq x$ is called an *item step* (Sijtsma and Molenaar 2002, p. 122) and $\mathsf{P}(X_j \geq x|\theta)$ is called the *item step response function.* Because $\mathsf{P}(X_j \geq 0|\theta) = 1$ for all θ, the relation between X_j and θ is characterized by m item step response functions: $\mathsf{P}(X_j \geq 1|\theta), \ldots, \mathsf{P}(X_j \geq m|\theta)$. For dichotomous items the item step response function reduces to $\mathsf{P}(X = 1|\theta)$.

Four assumptions are sufficient to define the two most popular nonparametric IRT models.

Unidimensionality : Only one latent variable θ is required to explain the association between item scores.

Local independence : $\mathsf{P}(X_1 = x_1, \ldots, X_J = x_J|\theta) = \prod_{j=1}^{J} \mathsf{P}(X_j = x_j|\theta)$.

Latent monotonicity : $\mathsf{P}(X_j \geq x|\theta_a) \leq \mathsf{P}(X_j \geq x|\theta_b)$, for all $\theta_a < \theta_b$ for $j = 1, \ldots, J; x = 1, \ldots, m$.

Nonintersection : If for a fixed value θ_0 $\mathsf{P}(X_i \geq x|\theta_0) \geq \mathsf{P}(X_j \geq y|\theta_0)$ then $\mathsf{P}(X_i \geq x|\theta) \geq \mathsf{P}(X_j \geq y|\theta)$ for all θ. This is true for all item pairs i, j $(i \neq j)$ and for all pairs of item scores x, y.

Local independence implies that the response to any item is unrelated to any other item when latent variable level is controlled. Latent monotonicity means that the item step response functions are nondecreasing functions of θ. (Latent monotonicity is usually referred to as monotonicity, but we prefer the term latent monotonicity to distinguish it from manifest monotonicity that it introduced later on.) Nonintersection means that the item step response functions do not intersect.

The assumptions unidimensionality, local independence, and latent monotonicity define the most general nonparametric IRT model: the monotone homogeneity model (Mokken 1971) also known as the nonparametric graded response model (Hemker, Sijtsma, Molenaar, and Junker 1997). Assumptions unidimensionality, local independence, latent monotonicity, and nonintersection define the double monotonicity model (Mokken 1971). Several other nonparametric IRT models have been proposed (see Van der Ark 2001, for an overview). All popular unidimensional parametric IRT models, such as the Rasch model (Rasch 1960), the two- and three-parameter logistic model (Birnbaum 1968), the graded response model (Samejima 1969), also assume unidimensionality, local independence, and latent monotonicity. Therefore, investigation of the assumptions of nonparametric IRT models is also useful when parametric IRT models are used. In addition, parametric IRT models assume that the item step response functions have a parametric functional form. The next two paragraphs describe the measurement properties of nonparametric IRT models.

Ordinal person measurement. For dichotomous items, the monotone homogeneity model implies stochastic ordering of θ by X_+ (known under the acronym SOL), that is,

$$\mathsf{P}(\theta > a|X_+ = L) \geq \mathsf{P}(\theta > a|X_+ = K) \text{ for all } a \text{ and for all } K < L \tag{1}$$

(Hemker, Sijtsma, Molenaar, and Junker 1996; Grayson 1988; Huynh 1994). Because the monotone homogeneity model is the most general IRT model, SOL also holds for other popular IRT models for dichotomous item scores. SOL allows the ordering of persons on θ by their test score. Although, in practice, the test score is almost always used to order or classify persons, test users do not always investigate whether SOL is a reasonable assumption.

For polytomous items, the monotone homogeneity model implies weak SOL; that is,

$$\mathsf{P}(\theta > a|X_+ > K) \geq \mathsf{P}(\theta > a|X_+ \leq K) \text{ for all } a \text{ and for all } K$$

(Van der Ark and Bergsma 2010). Weak SOL allows dividing the sample into a group of respondents having high θ values and a group of respondents having low θ values using $X_+ = K$ as a criterion. The practical relevance of weak SOL is that it allows to select the n respondents having the highest or lowest θ values by means of X_+. Examples are the selection of the 10 most depressed respondents in the sample or the single most qualified respondent in the sample. In general, the stricter form of SOL (Equation 1) does not hold for polytomous items (Hemker *et al.* 1997) but violations are rare if the number of items exceeds five (Van der Ark 2005).

Ordinal item measurement. In several situations it is important to know whether the ordering of the items in terms of difficulty or popularity is the same for all levels of the trait measured by the test (for an overview see Ligtvoet, Van der Ark, Te Marvelde, and Sijtsma 2010). This property is known as invariant item ordering (IIO; Sijtsma and Junker 1996; Sijtsma and Hemker 1998). Sijtsma and Hemker (1998) defined IIO as follows: A set of J items has an IIO if the items can be numbered $1, \ldots, J$ and ordered accordingly such that

$$\mathsf{E}(X_1|\theta) \geq \mathsf{E}(X_2|\theta) \geq \cdots \geq \mathsf{E}(X_J|\theta) \text{ for all } \theta. \tag{2}$$

For dichotomous items, IRT models that have non-intersecting item response functions (e.g., the Rasch model and the double monotonicity model) imply IIO. Other IRT models for dichotomous items do not imply IIO.

For polytomous IRT models an invariant ordering of items cannot be defined unequivocally. Stronger invariant ordering properties than IIO in Equation 2 were coined *latent scales* (Ligtvoet, Van der Ark, Bergsma, and Sijtsma 2011). In order of increasing strictness of ordering, they are the *latent scale for cumulative probability models* (LS-CPM), which means that for all x and all θ

$$\mathsf{P}(X_1 \geq x|\theta) \geq \mathsf{P}(X_2 \geq x|\theta) \geq \cdots \geq \mathsf{P}(X_J \geq x|\theta); \tag{3}$$

the *latent scale for continuation ratio models* (LS-CRM), which means that for all x and all θ

$$\mathsf{P}(X_1 \geq x|X_1 \geq x-1;\theta) \geq \mathsf{P}(X_2 \geq x|X_2 \geq x-1;\theta) \geq \cdots \geq \mathsf{P}(X_J \geq x|X_J \geq x-1;\theta); \tag{4}$$

and the *latent scale for adjacent category models* (LS-ACM), which means that for all x and all θ

$$\begin{aligned}\mathsf{P}(X_1 = x|X_1 = x-1 \vee X_1 = x;\theta) \geq & \mathsf{P}(X_2 = x|X_2 = x-1 \vee X_2 = x;\theta) \geq \\ \cdots \geq & \mathsf{P}(X_J = x|X_J = x-1 \vee X_J = x;\theta).\end{aligned} \tag{5}$$

The hierarchical structure of IIO and the latent scales can be summarized as follows (Ligtvoet *et al.* 2011):

$$\text{IIO} \quad \Leftarrow \quad \text{LS-CPM} \quad \Leftarrow \quad \text{LS-CRM} \quad \Leftarrow \quad \text{LS-ACM} \tag{6}$$

Sijtsma and Hemker (1998) showed that only a few very restrictive parametric and nonparametric polytomous IRT models imply IIO (Equation 2); the most well-known being the rating scale model (Andrich 1978), the sequential rating scale model (Tutz 1990), the strong double monotonicity model (Sijtsma and Hemker 1998). Most commonly used polytomous IRT models, such as the graded response model (Samejima 1969), the partial credit model (Masters

1982), the generalized partial credit model (Muraki 1992), and the monotone homogeneity model for polytomous items (Molenaar 1997) do neither imply IIO nor a latent scale. These models simply do not provide any information on the ordering of items, and IIO must be investigated separately.

2.2. MSA

MSA pertains to investigating the fit of the nonparametric IRT models. If a researcher wants to use a measurement property implied by a particular nonparametric IRT model, then he or she must demonstrate that the model fits the data sufficiently well. The rationale of nonparametric IRT is to investigate whether observable properties implied by the nonparametric IRT model hold in the data. An example of an observable property is nonnegative inter-item correlations, which are implied by the monotone homogeneity model. If the observable properties do not hold, the researcher must conclude that the nonparametric IRT model does not describe the data sufficiently well, and he or she must refrain from using the ordinal measurement properties implied by the model. For example, negative inter-item correlations in the data indicate that the monotone homogeneity model does not hold, and that the test score cannot be used for ordinal person measurement. Ideally, one should select a set of observable properties that detect serious misfit of the IRT model but ignore insignificant violations of the IRT model due to sample fluctuations. We briefly discuss the methods that were included in **MSP** and the first version of **mokken**. For detailed information on these methods see Molenaar and Sijtsma (2000); Sijtsma and Molenaar (2002); Van der Ark (2007, 2010b).

Scalability coefficients

Three types of scalability coefficients play an important role in MSA.

For each pair of items, there is an *item-pair scalability coefficient* H_{ij}; $i, j = 1, \ldots, J; i \neq j$. Let $\mathsf{COV}(X_i, X_j)$ be the covariance between X_i and X_j, and let $\mathsf{COV}(X_i, X_j)^{\max}$ be the maximum covariance between X_i and X_j given the marginal distributions of X_i and X_j. If the variances of X_i and X_j are both positive, then H_{ij} is the normed covariance between the item scores:

$$H_{ij} = \frac{\mathsf{COV}(X_i, X_j)}{\mathsf{COV}(X_i, X_j)^{\max}} \tag{7}$$

(Molenaar 1991). The range of H_{ij} is $\langle -\infty, 1]$. The monotone homogeneity model implies that $0 \leq H_{ij} \leq 1$, for all $i \neq j$; so negative values of H_{ij} indicate that at least one of the items does not fit the monotone homogeneity model and that item may be removed. In general, higher H_{ij} values result in stronger scales but in MSA the magnitude of positive H_{ij} coefficients is seldom interpreted (Van der Ark, Croon, and Sijtsma 2008).

For each item, there is an *item scalability coefficient* H_j; $j = 1, \ldots, J$. Let $R_{(j)} = X_+ - X_j$; $R_{(j)}$ is called the *rest score*. Let $\mathsf{COV}(X_j, R_{(j)})$ be the covariance between X_j and $R_{(j)}$, and let $\mathsf{COV}(X_j, R_{(j)})^{\max}$ be the maximum covariance between X_j and $R_{(j)}$ given the marginal distributions of X_j and $R_{(j)}$. If X_j and $R_{(j)}$ both have positive variance, then H_j is the normed covariance between the item score and the rest score:

$$H_j = \frac{\mathsf{COV}(X_j, R_{(j)})}{\mathsf{COV}(X_j, R_{(j)})^{\max}}. \tag{8}$$

> The monotone homogeneity model implies that $0 \leq H_j \leq 1$, for all j, so a negative item-scalability coefficient indicates that the corresponding item violates the monotone homogeneity model. Van Abswoude, Van der Ark, and Sijtsma (2004) argued that H_j can be interpreted in a similar way as the discrimination parameters in parametric IRT. $H_j < 0$ indicates that the corresponding item response function is generally decreasing; if $H_j = 0$, the item does not give information on θ; and $H_j > 0$ indicates that the item response function is generally increasing. If $H_j = 1$, the item is a so-called 'Guttman item' with perfect discrimination (Guttman 1950; Sijtsma and Molenaar 2002). To avoid items with very little discriminatory power, Mokken advocated to retain only those items for which H_j is greater than some positive lower bound c. The default lower bound in software is $c = 0.3$ is most often used.

For the entire set of items, there is a *test-scalability coefficient* H:

$$H = \frac{\sum_{j=1}^{J} \mathsf{COV}(X_j, R_{(j)})}{\sum_{j=1}^{J} \mathsf{COV}(X_j, R_{(j)})^{\max}}.$$

> The monotone homogeneity model implies that $0 \leq H \leq 1$. If $H = 1$ the test data follow a perfect Guttman scalogram. Mokken (1971) proposed the following rules of thumb for H: A scale is considered weak if $0.3 \leq H < 0.4$, moderate if $0.4 \leq H < 0.5$, and strong if $H > 0.5$. The predicates 'weak', 'moderate', and 'strong' refer to the degree to which the ordering of persons by test score X_+ accurately reflects an ordering on θ.

Function `coefH` in **mokken** (see Van der Ark 2007, 2010b) yields sample estimates of the scalability coefficients, which will be denoted by $\widehat{H}_{ij}$, $\widehat{H}_j$, and $\widehat{H}$, respectively. Recent changes in `coefH` are discussed in Section 3.3.

Automated item selection procedure

The AISP partitions a set of items into so-called *Mokken scales* possibly leaving some items unscalable. A Mokken scale is a set of items for which, for a suitably chosen positive lower bound c, all inter-item covariances are strictly positive and $\widehat{H}_j \geq c > 0$ (cf. Mokken, 1971 p. 184). Partitioning a set of items into Mokken scales using the AISP is an exploratory method for obtaining sets of items that satisfy some basic observable properties implied by the monotone homogeneity model and reasonable discriminatory power. Mokken (1971) devised a hierarchical clustering algorithm for the AISP; for a detailed description we refer to Sijtsma and Molenaar (2002, Chap. 5). The AISP can be run in **mokken** using the function `aisp` (see Van der Ark 2010b). Because sample estimates of the scalability coefficients are used to check whether the criteria of a Mokken scale are met, sampling fluctuations may affect the partitioning of an item set into Mokken scales. Recently, a new algorithm for the AISP has been included, which is discussed in Section 3.2.

Methods for investigating latent monotonicity

Manifest monotonicity is an observable property of the test data. Let r and s be realizations of R_{-j} $(r, s = 0, \ldots, [J-1] \times m)$, then manifest monotonicity is defined as

$$\mathsf{P}(X_j \geq x | R_{(j)} = s) \geq \mathsf{P}(X_j \geq x | R_{(j)} = r) \text{ for all } j, x, s > r.$$

For dichotomous items, latent monotonicity implies manifest monotonicity, but for polytomous items, some counterexamples have been found (Junker and Sijtsma 2000). However, Molenaar and Sijtsma (2000, p. 128) assumed that in practice, also for polytomous items, manifest monotonicity can be used to assess latent monotonicity. Manifest monotonicity can be investigated using `check.monotonicity` in **mokken** (see Van der Ark 2007, 2010b).

Methods for investigating intersection of item step response functions

Molenaar and Sijtsma (2000, pp. 74–88) describe three methods to investigate nonintersection: method "pmatrix", method "restscore", and method "restsplit". Method "pmatrix" investigates for all item pairs i, j $(i \neq j; \overline{X}_i > \overline{X}_j)$, and for all combinations of realizations x, y, whether

$$\mathsf{P}(X_i \geq x, X_k \geq z) \geq \mathsf{P}(X_j \geq y, X_k \geq z) \text{ for all } k \neq i, j \text{ and all } z \tag{9}$$

and

$$\mathsf{P}(X_i < x, X_k < z) < \mathsf{P}(X_j < y, X_k < z) \text{ for all } k \neq i, j \text{ and all } z. \tag{10}$$

Let $R_{(ij)}$ be the test score minus the scores on X_i and X_j with realization r $(r = 0, \ldots, [J - 2] \times m)$. Method "restscore" investigates for all item pairs i, j $(i \neq j; \overline{X}_i > \overline{X}_j)$, and for all combinations of realizations x, y, whether

$$\mathsf{P}(X_i \geq x | R_{(ij)} = r) \geq \mathsf{P}(X_j \geq y | R_{(ij)} = r) \text{ for all } r. \tag{11}$$

Method "restsplit" is a variant of method "restscore"; it investigates for all item pairs i, j $(i \neq j, \overline{X}_i > \overline{X}_j)$, and for all combinations of realizations x, y whether

$$\mathsf{P}(X_i \geq x | R_{(ij)} \geq r) \geq \mathsf{P}(X_j \geq y | R_{(ij)} \geq r) \text{ for all } r. \tag{12}$$

Equations 9, 10, 11, and 12 are observable properties of the double monotonicity model. Methods "pmatrix" and "restscore" can be investigated in **mokken** using functions `check.pmatrix` and `check.restscore`; method "restsplit" is not yet included in **mokken**. For dichotomous items these methods are useful because the double monotonicity model allows ordinal item measurement. For polytomous items, the double monotonicity model does not imply ordinal item measurement (e.g., Sijtsma, Meijer, and Van der Ark 2011). As a result, these methods have little use for polytomous items. New methods for investigating IIO, which can be applied to both dichotomous and polytomous items, have been developed (discussed hereunder); we advocate using these instead.

3. New developments in MSA

3.1. Investigating IIO

Ligtvoet *et al.* (2010) derived an observable property of IIO (Equation 2), which they coined *manifest IIO* (MIIO). MIIO as implemented in **mokken** means that items can be numbered and ordered accordingly such that

$$\mathsf{E}(X_i | R_{(ij)} = r) \geq \mathsf{E}(X_j | R_{(ij)} = r) \text{ for all } i < j \text{ and all } r. \tag{13}$$

Furthermore they developed a method to investigate MIIO and investigated the sensitivity and specificity of the new method. First, the items are ordered by descending mean item score, and numbered accordingly such that $\overline{X}_1 \geq \overline{X}_2 \geq \cdots \geq \overline{X}_J$. Second, for each item pair (i, j), $i < j$, it is investigated whether Equation 13 is violated, yielding $\binom{J}{2}$ Boolean outcomes on violation of MIIO. For a particular item pair, the following statistical testing procedure is applied to determine whether or not the item pair violates Equation 13: If the sample means $\overline{X}_i|R_{(ij)} = r$ and $\overline{X}_j|R_{(ij)} = r$ are reversely ordered (i.e., $\overline{X}_i|R_{(ij)} = r$ is less than $\overline{X}_j|R_{(ij)} = r$), a one-sided t-test is conducted for deciding whether the violation is significant. Violations less than *minvi* (default $minvi = m \times 0.03$) are ignored to avoid testing very small violations on a scale from 0 to m. Following other methods in MSA, significance is tested at level α without a correction for multiple testing, because each significant violation in itself provides evidence against MIIO (cf. Molenaar and Sijtsma, 2000, p. 72). If one or more significant t values are found, the item pair is said to violate MIIO. Adjacent rest-score groups $r, r+1, \ldots$ containing few observations, may be joined to increase statistical power. The conventions for joining rest-score groups used for all methods in **MSP** and **mokken** are applied (Molenaar and Sijtsma 2000, p. 67). If MIIO does not hold for all item pairs, items are removed one-by-one using a backwards selection algorithm until no violations remain. For the remaining set of items for which MIIO holds coefficient H^T is computed (H^T means coefficient H computed on the transposed data matrix). Ligtvoet *et al.* (2010) advocated that if MIIO holds, $0.3 < H^T \leq 0.4$ should be interpreted as a weak ordering, $0.4 < H^T \leq 0.5$ as a moderate ordering, and $H^T > 0.5$ as a strong ordering.

Ligtvoet *et al.* (2011) extended the method from IIO to latent scales (Equations 3, 4, and 5). They proved that MIIO (Equation 13) is also an observable property for all latent scales. Moreover, they found two other observable properties for the latent scales, and devised a method for investigating these observable properties similar to the method for investigating MIIO. The observable properties are called *manifest scale of the cumulative probability model* (MS-CPM), which is implied by all latent scales but not by IIO, and *increasingness in transposition* (IT Rosenbaum 1987), which is implied only by the latent scale for adjacent category models (Equation 5). Hence, Equation 6 can be extended to

$$\begin{tikzcd}
\text{IIO} \arrow[d, Rightarrow] & \text{LS-CPM} \arrow[l, Rightarrow] \arrow[d, Rightarrow] & \text{LS-CRM} \arrow[l, Rightarrow] & \text{LS-ACM} \arrow[l, Rightarrow] \arrow[d, Rightarrow] \\
\text{MIIO} & \text{MSCPM} & & \text{IT}
\end{tikzcd}$$

For a description of these manifest properties see Ligtvoet *et al.* (2011).

A pilot study showed that the methods based on MIIO, MSCPM, and IT may suggest different items as candidates for removal. Under violations of IIO, the method based on MSCPM was the most sensitive and, in general, suggested the removal of more items than the methods based on IT and MIIO. At first glance, this may seem a strange result because IT relates to a stricter ordering property than MSCPM. It may be explained as follows: Suppose that IIO does not hold. By implication, none of the latent scales hold, and it is expected that all methods indicate that IIO should be rejected. Unlike the latent scales, the three methods are not (necessarily) hierarchically related; in fact, the three methods investigate different properties in the data. Therefore, the three methods may suggest different items for deletion.

The function `check.iio` in **mokken** consists of methods to investigate IIO, and is used as follows:

```
check.iio(X, method = "MIIO", minvi = default.minvi,
  minsize = default.minsize, alpha = 0.05, item.selection = TRUE,
  verbose = FALSE)
```

The argument `method` indicates the method used to investigate IIO. Besides default option `"MIIO"`, options `"MSCPM"` and `"IT"` are also allowed. Changing the default values of arguments `minvi`, `minsize`, and `alpha` change *minvi*, the minimum number of respondents in each rest-score group, and the nominal type I error rate of the statistical tests, respectively. If `item.selection` is `FALSE`, the backward selection algorithm is omitted, and if `verbose` is `TRUE`, additional output is sent to the screen.

Investigating IIO is demonstrated using the item scores on first ten items in the data set `acl`; Data set `acl` contains the item scores of 433 first-year Psychology students on 218 items from a Dutch version of the Adjective Checklist (Gough and Heilbrun 1980). Each item has five ordered answer categories (0 = completely disagree, 1 = disagree, 2 = agree nor disagree, 3 = agree, 4 = completely agree). The respondents were instructed to consider whether an adjective described their personality, and mark the answer category that fits best to this description. The first ten items are extremely popular or unpopular self-descriptive adjectives. The unpopular items (indicated by an asterisk) were reversely coded. For example, the item "cruel" is extremely unpopular as a self-descriptive adjective. They are indicators for a response style called *communality*. Respondents that have a high test score are particularly good at giving responses that are commonly accepted. Investigating IIO of these 10 items provides information whether these adjectives are invariantly ordered in popularity.

```
R> library("mokken")
R> data("acl")
R> X <- acl[, 1:10]
R> summary(check.iio(X))

$method
[1] "MIIO"

$item.summary
               ItemH #ac #vi #vi/#ac maxvi  sum sum/#ac tmax #tsig crit
cruel*          0.25  27   0    0.00  0.00 0.00  0.0000 0.00     0    0
unintelligent*  0.12  26   2    0.08  0.15 0.29  0.0112 2.17     1   74
unscrupulous*   0.24  26   1    0.04  0.14 0.14  0.0054 1.21     0   37
unfriendly*     0.31  27   1    0.04  0.15 0.15  0.0057 2.17     1   53
thankless*      0.24  27   1    0.04  0.12 0.12  0.0045 1.50     0   35
dependable      0.30  27   0    0.00  0.00 0.00  0.0000 0.00     0    0
obnoxious*      0.29  27   1    0.04  0.12 0.12  0.0045 1.50     0   32
reliable        0.30  27   0    0.00  0.00 0.00  0.0000 0.00     0    0
honest          0.26  27   2    0.07  0.18 0.31  0.0114 2.08     1   70
deceitful*      0.32  25   2    0.08  0.18 0.31  0.0124 2.08     1   69

$backward.selection
               step 1 step 2 step 3
cruel*              0      0      0
unintelligent*      1     NA     NA
unscrupulous*       0      0      0
unfriendly*         1      0      0
thankless*          0      0      0
```

```
dependable          0       0       0
obnoxious*          0       0       0
reliable            0       0       0
honest              1       1      NA
deceitful*          1       1       0

$HT
[1] 0.07338353
```

Function `check.iio` returns an object of class `iio.class` containing all test results. Typically `check.iio` is used in combination with `summary`, returning a list. The first element of the output (`$method`) echoes the method used. The second element (`$item.summary`) shows a summary of the results for each item; it includes `itemH` = item-scalability coefficient H_j (Equation 8); `#ac` = the number of possible violations in which the item can be involved; `#vi` = number of actual violations in which the item is involved; `maxvi` = maximum violation; `sum` = sum of all violations in which the item is involved; `tmax` = maximum t statistic; `tsig` = number of times the item appears in a significant violation of MIIO; and `crit` = the *crit value* (Molenaar and Sijtsma 2000, pp. 47); which is a weighted sum of the other components (i.e., `itemH`, `#ac`, etc.), and by some researchers used as a diagnostic statistic. High crit values indicate bad items (for more details and examples see, e.g., Van Schuur 2011, p. 54).

The third element (`$backward.selection`) shows the backward selection procedure: Step 1 shows, for each item, the number of conflicting items. Item j is a conflicting item to item i (and vice versa) if their estimated item-response functions intersect; resulting in a violation of Equation 13. In this example, the estimated item response functions of items 2 (unintelligent*) and 4 (unfriendly*), and items 9 (honest) and 10 (deceitful*) intersect. To check this, pairs of estimated item response functions can be plotted using `plot(check.iio(X))`. The item or items having the highest number of conflicting items are candidates for removal. If there is more than one candidate for removal, then the item having the lowest $\widehat{H}_j$ value is removed. In this example, items 2, 4, 9, and 10 all have one conflicting item. Item 2 had the lowest $\widehat{H}_j$ value and was removed. Step 2 shows for each of the remaining items the number of conflicting items. The procedure is repeated until no more violations occur. Finally, the fourth element (`$HT`) the value of coefficient $\widehat{H}^T$ computed on the remaining items. The output for methods MS-CPM and IT is similar to the output for method MIIO.

3.2. New algorithms for the automated item selection procedure

Mokken's AISP aims at partitioning a set of items into Mokken scales, possibly leaving some items unscalable. The hierarchical clustering algorithm starts with the two items having the highest $\widehat{H}_{ij}$ value that is significantly greater than 0; hence, a scale consists of at least two items. The algorithm keeps adding items that meet the criteria of a Mokken scale, until there are no more items left that meet the criteria. Then the procedure is repeated for the remaining unselected items resulting in a second Mokken scale; then the procedure is repeated a second time, until no more scales can be formed. Hence, the idea is to have as many good items in the first scale, then have as many good items - that were not selected in the first scale - in the second scale, and so on. Straat, Van der Ark, and Sijtsma (2013) noticed two problems in the hierarchical clustering algorithm. First, although the objective of the AISP is clear, the algorithm lacked a clear objective function. Thus, it is difficult to assess whether an optimal partitioning was obtained. Second, in some cases the algorithm does not yield Mokken scales because some item-scalability coefficients $\widehat{H}_j$ are slightly less than the required lower bound c. They proposed an objective function (for details, see Straat *et al.* 2013) that was completely in line with Mokken's original idea, and devised a genetic algorithm for the AISP.

The two algorithms are included in the function `aisp`:

```
aisp(X, search = "normal", lowerbound = 0.3, alpha = 0.05, popsize = 20,
  maxgens = default.maxgens, pxover = 0.5, pmutation = 0.1, verbose = FALSE)
```

The argument `search` (name taken from **MSP**) gives the algorithm: `"normal"` is the hierarchical clustering algorithm, `"ga"` is the genetic algorithm. The arguments `lowerbound`, `alpha`, and `verbose` affect the hierarchical clustering algorithm: `lowerbound` specifies lower bound c for scalability coefficient $\widehat{H}_j$; `alpha` specifies the nominal type I error rate for testing $H_{ij} = 0$; and `verbose` specifies whether the hierarchical steps in the scaling procedure should be send to the screen. The arguments `popsize`, `maxgens`, `pxover`, and `pmutation` affect the genetic algorithm: Argument `popsize` is the size of the population of partitionings. Argument `maxgens` is the number of generations considered before the genetic algorithm stops; `maxgens` depends on the number of items and its default value is $1000 \cdot 10^{\log_2 \frac{J}{5}}$. Argument `pxover` is the probability of cross-over, and `pmutation` is the probability of mutation.

The following code gives the partitioning of the items into Mokken scales according to the hierarchical cluster algorithm and the new genetic algorithm.

```
R> scale.normal <- aisp(X)
R> scale.ga <- aisp(X, search = "ga")
R> cbind(scale.normal, scale.ga)
```

```
               Scale Scale
reliable           1     1
honest             1     0
unscrupulous*      0     0
deceitful*         1     1
unintelligent*     0     0
obnoxious*         2     1
thankless*         2     1
unfriendly*        2     1
dependable         1     1
cruel*             2     1
```

Function `aisp` returns a $J \times 1$ matrix containing the partitioning of the items. The numbers $1, 2, \ldots$ indicate the scale that the items are assigned to. A value 0 means that the item is unscalable. Using default option `search = "normal"` resulted in two scales: Scale 1 consists of four items and Scale 2 consists of four items; two items are unscalable. Using `search = "ga"` resulted in one scale consisting of seven items, leaving three items unscalable. The partitioning provided by the two algorithms is rather different. The genetic algorithm yielded a better partitioning in terms of Mokken's desiderata because the longest scale consists more items than in the partitioning obtained using the hierarchical clustering algorithm. The hierarchical clustering algorithm started with item pair ("reliable", "dependable"), and then

	Number of items						
	10	15	20	25	30	35	40
`maxgens`	10,000	38,456	100,000	209,859	384,559	641,734	1,000,000
CPU `"ga"`	0'08	0'21	1'18	3'03	8'45	21'38	49'17
CPU `"normal"`	0'00	0'00	0'01	0'01	0'02	0'03	0'04

Table 1: Default maximum number of generations `maxgens` in the genetic algorithm of the AISP, and computation time (CPU) in minutes and seconds for the two algorithms in the AISP for an increasing number of items. Default settings were used on an ASUS U6Sg notebook under Windows 7.

added item "honest" to the first scale. The inclusion of "honest" prevented the inclusion of any other item except "deceitful*" because due to the presence of "honest" the item-scalability coefficients of the remaining items (except "deceitful*") did not exceed the lower bound. Due to the hierarchical structure of the algorithm "honest" stayed in the first scale. In the genetic algorithm, also partitionings without "honest" were considered, which resulted in a longer first scale.

The two algorithms were compared in a simulation study (Straat *et al.* 2013). On the one hand, results showed that the genetic algorithm yielded better partitionings in terms of the objective function, and the genetic algorithm did not yield scales violating the criteria of a Mokken scale. On the other hand, results showed that the genetic algorithm can be slow if the number of items is large. Table 1 (first row) shows for $10, 15, \ldots, 40$ items, the maximum number of generations in the genetic algorithm and the computation time of the two algorithms in minutes and seconds. Hence, for large numbers of items the genetic algorithm may not be suitable.

3.3. Computing standard errors for scalability coefficients

Several heuristic rules in MSA involve interpreting absolute values of the scalability coefficients. For example, Mokken used the label *strong scale* if $H > 0.50$; and an item is selected in a Mokken scale only if, $H_j > c$, and all $H_{ij} > 0$. For a sound application of these guidelines, the standard errors of the scalability coefficients should be taken into account. For example, if $\widehat{H} = 0.51$ but its standard error equals 0.09, then the qualification 'strong scale' is not appropriate because the standard error indicates that the population value of H may well be less than 0.50.

Van der Ark, Croon, and Sijtsma (2008) derived standard errors for the scalability coefficients for dichotomous item scores using marginal models. This approach required the evaluation of all possible response patterns. For a typical psychological test having $J = 20$ items, this results in $2^{20} > 1,000,000$ response patterns, which is infeasible for practical computation. For polytomous items, the curse of dimensionality is even greater because 20 Likert items (5 ordered answer categories) result in almost 100 trillion response patterns. Kuijpers, Van der Ark, and Croon (2011) generalized the approach to scalability coefficients for polytomous items, and solved the dimensionality problem. The standard errors are estimated under the assumption that the observed frequencies of the item-score patterns follow a multinomial distribution. Using normal approximation, the standard errors can be used to compute confidence intervals. Because the theoretical maximum of the scalability coefficients equals 1, these confidence intervals may not be range preserving; alternatively, bootstrap confidence intervals of the scalability coefficients may be computed (Van Onna 2004).

As of version 2.6 of **mokken**, function `coefH`, which provides the scalability coefficients, also gives the theoretical standard errors:

```
coefH(X, se = TRUE, nice.output = TRUE)
```

By default `coefH` returns an object of class `noquote`, which presents the scalability coefficients and standard errors in a pretty format. Argument `se` is a Boolean variable indicating whether or not standard errors should be computed. For huge sample sizes in combination with large numbers of items, the computation of standard errors may take rather long or memory problems may occur. In this case, it is advised to set `se = FALSE`. Argument `nice.output` is a Boolean variable indicating whether the resulting output should be of class `noquote`: `nice.output = TRUE` produces pretty output of class `noquote`, whereas `nice.output = FALSE` produces a `matrix`, which may be convenient if the scalability coefficients are being used as input later on, for example, in simulation studies.

Because the scalability coefficients are ratios, standard errors can be very large even for large sample sizes when the item scores have a skewed distribution. This is illustrated by the following code: The scores on two items are simulated for a large sample ($N = 10,000$); coefficient H is computed; and a cross-classification of the item scores is presented.

```
R> set.seed(1)
R> library("MASS")
R> X <- sign(mvrnorm(10000, c(0, -3),
+    matrix(c(1, 0.1, 0.1, 1), 2, 2)))/2 + 0.5
R> coefH(X)$H

 Scale H se
   0.337 (0.313)

R> table(X[, 1], X[, 2])

        0    1
  0 5021    3
  1 4970    6
```

The table of bivariate frequencies shows that `X[, 2]` has a very skewed distribution. The standard error of $\widehat{H}$ (0.313) is large despite the large sample size. It is, therefore, important that a test developer always knows the stability of the estimated coefficients.

3.4. Computation of test-score reliability

Test-score reliability (denoted $\rho_{XX'}$) is the degree of stability of a respondent's test score across independent replications of a test administration. It is one of the most important concepts in psychometrics. Most often, psychological tests are only administered once, so there are no replications of the test administration. Even if a test is administered multiple times to the same sample, it is very unlikely that these would be independent replications because at the second administration, the respondents are very likely to remember the test items from the first administration. As a result, the test-score reliability cannot be computed exactly and has to be estimated. The most common estimator is Cronbach's alpha (Cronbach 1951), which estimates the test-score reliability using the data obtained in a single test administration. Cronbach's alpha is in fact one of the worst estimators because it underestimates the test-score reliability to a larger degree than readily-available alternative statistics (e.g., lambda-2) do (Schmitt 1996; Sijtsma 2009).

Mokken (1971), (also see Sijtsma and Molenaar 1987; Molenaar and Sijtsma 1988) derived a statistic that is an unbiased estimator of the test-score reliability given that the double monotonicity model holds. This is a rather strong assumption. This statistic, which is known under several names including "rho" and the MS statistics, is included in **MSP** but suffers from a programming error (Van der Ark 2010a). As a result **MSP** may not produce the correct value.

Alternatively, Van der Ark, Van der Palm, and Sijtsma (2011) proposed a statistic coined the latent class reliability coefficient (LCRC) that is an unbiased estimator of the test-score reliability given that an unconstrained latent class model holds. First, the joint density of the item scores is estimated using a latent class model (e.g., Linzer 2011; Vermunt, Van Ginkel, Van der Ark, and Sijtsma 2008), and, second, the reliability coefficient is derived from the latent class model. When latent class models are used for density estimation, the number latent classes should be large to allow precise estimation of the density; issues such as interpretation of the latent classes, identifiability, and local maxima, which are important in the traditional use of latent class models, are not so important. Compared to the assumptions of the double monotonicity model (unidimensionality, local independence given θ, monotonicity, and non-intersection), the assumptions of the unconstrained latent class model (local independence given class membership) are rather weak; especially when the number of latent classes is large. This gives LCRC an advantage over the MS statistic. A drawback of LCRC is that the number of latent classes should be specified beforehand, which means that the fit of several latent class models must be investigated. Simulation studies showed that MS and LCRC are superior to Cronbach's alpha, and if the data are not unidimensional LCRC is the statistic to be preferred (Van der Ark *et al.* 2011).

The MS statistic, Cronbach's alpha, lambda-2 and LCRC have been included in **mokken** in the function `check.reliability`:

```
check.reliability(X, MS = TRUE, alpha = TRUE, lambda.2 = TRUE, LCRC = FALSE,
  nclass = nclass.default)
```

Arguments `MS`, `alpha`, `lambda.2`, and `LCRC` are Boolean variables indicating whether or not MS, Cronbach's alpha, lambda-2, and LCRC should be computed. Argument `nclass` denotes the number of classes used for computing LCRC; the default value equals $J-1$. The output is a list containing the values of the required test-score reliability estimators.

4. Real-data examples

This section is a demonstration of the new methods in **mokken**. For the larger part of the demonstration, we use the scores of 425 children on 12 items measuring transitive reasoning (Verweij, Sijtsma, and Koops 1996). Transitive reasoning is the ability to deduce a relationship

Item	Property	Format	Objects	Measures	p value
9	length	$A = B < C = D$	sticks	12.5, 12.5, 13, 13 (cm)	0.30
12	pseudo				0.48
10	weight	$A = B < C = D$	balls	60, 60, 100, 100 (g)	0.52
11	pseudo				0.64
4	weight	$A = B = C = D$	cubes	65 (g)	0.78
5	weight	$A < B < C$	balls	40, 50, 70 (cm)	0.80
2	length	$A = B = C = D$	tubes	12 (cm)	0.81
7	length	$A > B = C$	sticks	28.5, 27.5, 27.5 (cm)	0.84
3	weight	$A > B > C$	tubes	45, 25, 18 (g)	0.88
1	length	$A > B > C$	sticks	12, 11.5, 11 (cm)	0.94
8	weight	$A > B = C$	balls	65, 40, 40 (g)	0.97
6	area	$A > B > C$	discs	7.5, 7, 6.5 (diameter; cm)	0.97

Table 2: Characteristics of the 12 items measuring transitive reasoning.

from two or more other relationships of equality or inequality. For example, Item 1 (Table 2) contains three sticks of different length, here labeled A (12cm), B (11.5cm), and C (11cm). First, the test administrator shows the child two pairs of sticks, consecutively; for example, first A and B and then A and C. Second, the test administrator asks the child whether B is longer, equal, or smaller than C. If the item consists of four objects rather than three, the test administrator shows three pairs of objects that, together, are sufficient to deduce the relationships among all objects, and asks for the relationship between the objects in each of the three remaining pairs. Table 2 shows the characteristics of the 12 items. The items are ordered by the proportion of correct answers. For items 11 and 12, transitive reasoning was not a necessary ability to solve the items; they are referred to as pseudo items. For more detailed information, see Verweij *et al.* (1996). The data set `transreas` is available from the package **mokken**. The first variable is the students' grade, which is not used for scaling and omitted.

```
R> data("transreas")
R> X <- transreas[, -1]
```

First, it should be investigated whether all items measure transitive reasoning. If this is true, it is expected that the 12 items form a Mokken scale, and all $\widehat{H}_{ij}$s are positive and all $\widehat{H}_j$s are greater than or equal to an appropriately chosen lower bound c; here $c = 0.3$.

```
R> coefH(X)
```

The results (output not included) show that the 12 items do not form a Mokken scale because several negative $\widehat{H}_{ij}$ values were found, and several values of $\widehat{H}_j$ were less than 0.3.

Second, because the entire set of items does not form a Mokken scale, one should find the largest set of items that forms a Mokken scale. Both algorithms in the AISP were used.

```
R> scale.normal <- aisp(X)
R> scale.ga <- aisp(X, search = "ga")
R> cbind(scale.normal, scale.ga)
```

```
     Scale Scale
T09L     1     1
T12P     0     0
T10W     1     1
T11P     0     0
T04W     2     2
T05W     0     0
T02L     2     2
T07L     1     1
T03W     1     1
T01L     1     1
T08W     1     1
T06A     1     1
```

Unlike the example in Section 3.2, the two algorithms yield the same item partitioning. The two pseudo items (11 and 12) are unscalable (`Scale == 0`). This seems logical because the items do not measure transitive reasoning. Also, item 5 is unscalable, which is difficult to explain. If the lower bound is set lower than 0.30 (e.g., 0.25), item 5 is included in the scale. Items 2 and 4 form a separate second scale, which may be explained by the fact that both items pertain to equalities rather than inequalities. Items 1, 3, 6, 7, 8, 9, and 10 are included in the final scale, denoted `Y`.

```
R> Y <- X[, scale.normal == 1]
```

Third, one would be interested to know whether the 7 selected items form a unidimensional scale.

```
R> coefH(Y)

$Hij
     T09L    se      T10W    se      T07L    se      T03W    se
T09L                  0.463  (0.072)  0.497  (0.140)  0.526  (0.161)
T10W  0.463  (0.072)                  0.534  (0.096)  0.568  (0.111)
T07L  0.497  (0.140)  0.534  (0.096)                  0.444  (0.080)
T03W  0.526  (0.161)  0.568  (0.111)  0.444  (0.080)
T01L  0.336  (0.260)  0.538  (0.161)  0.479  (0.114)  0.548  (0.108)
T08W  1.000  (0.000)  0.588  (0.209)  0.493  (0.154)  0.677  (0.135)
T06A  0.698  (0.286)  0.650  (0.222)  0.785  (0.137)  0.589  (0.162)

     T01L    se      T08W    se      T06A    se
T09L  0.336  (0.260)  1.000  (0.000)  0.698  (0.286)
T10W  0.538  (0.161)  0.588  (0.209)  0.650  (0.222)
T07L  0.479  (0.114)  0.493  (0.154)  0.785  (0.137)
T03W  0.548  (0.108)  0.677  (0.135)  0.589  (0.162)
T01L                  0.317  (0.133)  0.420  (0.157)
T08W  0.317  (0.133)                  0.530  (0.153)
T06A  0.420  (0.157)  0.530  (0.153)
```

```
$Hi
     Item H  se
T09L   0.496 (0.070)
T10W   0.518 (0.057)
T07L   0.507 (0.060)
T03W   0.531 (0.064)
T01L   0.462 (0.085)
T08W   0.546 (0.103)
T06A   0.592 (0.098)

$H
 Scale H se
   0.515 (0.052)
```

If one does not take into account the standard errors of the scalability coefficients, the results look promising. All $\widehat{H}_{ij}$ coefficients are positive, all $\widehat{H}_j$ coefficients are greater than 0.3, and the items seem strongly scalable because $\widehat{H} > 0.5$. Items 8 (a very easy item, p value = 0.97) and 9 (difficult item, p value = 0.30) have a perfect scalability coefficient, which means that none of the children made a Guttman error; that is, none of the children failed the easy item and passed the difficult item. However, if the standard errors are taken into account, the results seem less promising. Notice that the standard errors of coefficients $\widehat{H}_{ij}$ are generally very large. For example, when computing the bounds of the asymptotic 95% confidence intervals by $\widehat{H}_{ij} \pm 1.96 \times \mathrm{se}(\widehat{H}_{ij})$, it becomes clear that for some coefficients the population value may be negative (e.g., $\widehat{H}_{1,9} = 0.336$ but $\mathsf{P}(-0.174 < H_{1,9} < 0.846) = 0.95$). The standard errors of the $\widehat{H}_j$ coefficients are also rather large; although it is reasonable to assume that the H_j values in the population exceed the lower-bound value 0.3. Given the standard error of coefficient $\widehat{H}$, it is plausible that the population value of H does not exceed 0.5; when computing the 95% confidence interval, we find $\mathsf{P}(0.413 < H < 0.617) = 0.95$. Therefore, the label 'moderate scale' is more appropriate than 'strong scale'. Several other properties may be investigated, such as monotonicity using the function `check.monotonicity`, and nonintersection of item step response functions using `check.pmatrix` and `check.restscore`. These functions are described in Van der Ark (2007).

Fourth, one may be interested in the reliability of the selected items.

```
R> check.reliability(Y, LCRC = TRUE)

$MS
[1] 0.6837627

$alpha
[1] 0.6058617

$lambda.2
[1] 0.6258501

$LCRC
[1] 0.7131621
```

As can be expected for sets of item scores that are not completely unidimensional, coefficients MS and LCRC have higher values than alpha and lambda-2 which underestimate the true test-score reliability. The value of LCRC may show small deviations across runs due to local maximums in the estimation of the latent class models. Also, the value of LCRC may change for different numbers of latent classes. Whether, the test-score reliability is high enough depends on the purpose of the test. If the test scores are used to investigate the difference in transitive reasoning between two groups (e.g., boys and girls), the test-score reliability suffices. If the test scores are used in a correlational study, then test-score reliability is more important. The correlation between the test scores and any other variable cannot exceed the square root of the product of reliability of both test-scores (Lord and Novick 1968, pp. 69-74). The test-score reliability is most important for individual diagnosis. Following Lord and Novick (1968, p. 59), lower and upper bounds of the 95% confidence interval of the respondents *true score* equal

$$X_+ \pm 1.96 \times \sigma_X \sqrt{1 - \rho_{XX'}}.$$

Using LCRC as a plug-in estimate of $\rho_{XX'}$, and the sample standard deviation as a plug-in estimate for σ_X, the following code shows the confidence interval bounds for test score $X_+ = 4$. Note that `set.seed(1)` eliminates differences in `rXX` over replications due to local maxima in the latent class model.

```
R> set.seed(1)
R> rXX <- check.reliability(Y, LCRC = TRUE)$LCRC
R> sigmaX <- sd(apply(Y, 1, sum))
R> cat(" Lower bound:", round(4 - 1.96 * sigmaX * sqrt(1 - rXX), 4),
+    "\n", "Upper bound:",
+    round(4 + 1.96 * sigmaX * sqrt(1 - rXX), 4), fill = TRUE)

 Lower bound: 2.7066
 Upper bound: 5.2934
```

The researcher must decide whether this interval is precise enough for the purpose of the psychological test. As the test-score reliability increase, the interval becomes smaller.

Fifth, one may be interested to find out whether the items are invariantly ordered. Here, the items are dichotomous. As a result, only the default option (`method = "MIIO"`) in `check.iio` can be used.

```
R> summary(check.iio(Y))

$method
[1] "MIIO"

$item.summary
     mean #ac #vi #vi/#ac maxvi  sum sum/#ac tmax #tsig
T06A 0.97  17   0    0.00  0.00 0.00       0 0.00     0
T08W 0.97  17   0    0.00  0.00 0.00       0 0.00     0
T01L 0.94  16   0    0.00  0.00 0.00       0 0.00     0
```

```
T03W 0.88  16   1    0.06  0.03 0.03        0 1.15     0
T07L 0.84  16   1    0.06  0.03 0.03        0 1.15     0
T10W 0.52  14   0    0.00  0.00 0.00        0 0.00     0
T09L 0.30  12   0    0.00  0.00 0.00        0 0.00     0

$backward.selection
     step 1
T06A      0
T08W      0
T01L      0
T03W      0
T07L      0
T10W      0
T09L      0

$HT
[1] 0.7540433
```

Following procedure MIIO, we found two items involved in violations of IIO. One violation means intersecting functions $E(X_i|R_{(i)})$ and $E(X_j|R_{(j)})$ (Equation 13), which pertain to two items. So if two items are involved in a violation of IIO, this means that the total number of violations is 1. In this case, the violation was not significant. Hence, no items have to be removed in the backward selection procedure. In combination with a high value for coefficient H^T ($H^T = 0.75$), this suggests a strong support for IIO. Methods check.restscore and check.pmatrix, described in Van der Ark (2007) are also suitable for investigating IIO for dichotomous items.

A demonstration of check.iio for polytomous items is taken from Adjective Checklist. Items 101 to 110 are adjectives that describe an aggressive personality. Negatively worded items are indicated by an asterisk.

```
R> X <- acl[, 101:110]
R> data.frame(Item = 101:110, Mean = round(apply(X, 2, mean), 2))

              Item Mean
patient*       101 1.82
calm*          102 1.77
argumentative  103 1.13
unkind         104 0.71
impatient      105 1.91
excitable      106 1.48
quiet*         107 1.91
quarrelsome    108 0.79
irritable      109 1.50
vindictive     110 0.88
```

Results from the AISP indicate that Item 102 (calm*) and Item 107 (quiet*) do not belong to the scale, whereas the remaining 8 items form a weak to moderate scale ($\widehat{H} = 4.02$, se =

0.022) The 8 selected items have rather different mean item scores, with Item 104 (unkind) being the least popular and Item 105 (impatient) being the most popular. It is investigated whether the ordering of the adjectives in terms is invariant across all levels of aggression.

Investigating the least restrictive manifest ordering property MIIO yielded the following results

```
R> X <- X[, -c(2, 7)]
R> summary(check.iio(X, method = "MIIO"))
R> plot(check.iio(X), item.pairs = 28)

$method
[1] "MIIO"

$item.summary
              ItemH #ac #vi #vi/#ac maxvi  sum sum/#ac tmax #tsig crit
impatient      0.43  21   1    0.05  0.13 0.13  0.0063 1.05     0   27
patient*       0.33  21   1    0.05  0.13 0.13  0.0063 1.05     0   33
irritable      0.50  21   1    0.05  0.14 0.14  0.0067 1.36     0   27
excitable      0.46  20   1    0.05  0.14 0.14  0.0070 1.36     0   29
argumentative  0.37  21   0    0.00  0.00 0.00  0.0000 0.00     0    0
vindictive     0.37  21   0    0.00  0.00 0.00  0.0000 0.00     0    0
quarrelsome    0.41  20   1    0.05  0.14 0.14  0.0070 1.79     1   49
unkind         0.31  21   1    0.05  0.14 0.14  0.0067 1.79     1   53

$backward.selection
              step 1 step 2
impatient          0      0
patient*           0      0
irritable          0      0
excitable          0      0
argumentative      0      0
vindictive         0      0
quarrelsome        1      0
unkind             1     NA

$HT
[1] 0.3514797
```

The output shows that six items were involved in one violation of MIIO. The first violation (not significant) pertains to Items 101 (impatient) and 103 (patient*), the second violation (not significant) pertains to Items 104 (irritable) and 105 (excitable), the third violation (significant) pertains to Items 109 (quarrelsome) and 110 (unkind). The last violation is visualized using `plot(check.iio(X), item.pairs = 28)` (Figure 1; 28 refers to the number of the item pair: $1 = (101, 103)$, $2 = (101, 104)$, $\ldots$, $28 = (109, 110)$. After removing item "unkind" no violations were left (`$backward.selection step 2`). For the 7 remaining items $H^T = 0.35$, suggesting weak support for IIO.

Alternatively, methods `"MSCPM"` can be used to check a stronger forms of item ordering.

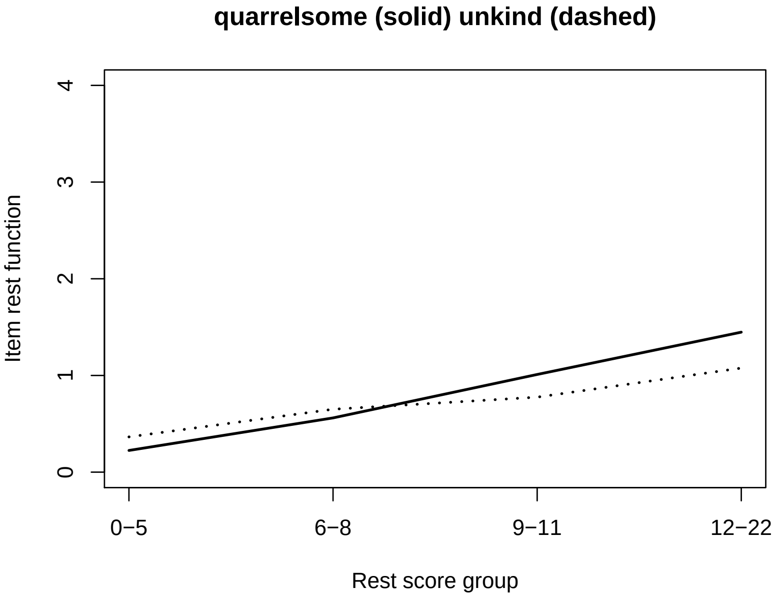

Figure 1: Estimated item response function of Item 109 (quarrelsome) and Item 110 (unkind).

```
R> summary(check.iio(X, method = "MSCPM"))

$method
[1] "MSCPM"

$item.summary
              ItemH #ac #vi #vi/#ac maxvi  sum sum/#ac  zmax #zsig crit
impatient      0.43  84   3    0.04  0.08 0.20  0.0024 10.17     3   84
patient*       0.33  84   4    0.05  0.08 0.28  0.0034 10.17     4   97
irritable      0.50  84   4    0.05  0.08 0.25  0.0030 10.10     4   88
excitable      0.46  80   3    0.04  0.08 0.17  0.0022  8.84     3   77
argumentative  0.37  84   4    0.05  0.07 0.22  0.0026  5.01     2   58
vindictive     0.37  84  10    0.12  0.13 0.61  0.0072  9.77     8  126
quarrelsome    0.41  80   6    0.08  0.15 0.42  0.0053  9.77     6  112
unkind         0.31  84   4    0.05  0.15 0.39  0.0047  8.76     4  100

$backward.selection
              step 1 step 2 step 3 step 4 step 5
impatient          1      1      0      0      0
patient*           2      2     NA     NA     NA
irritable          2      2      1      1      0
excitable          1      1      1      1     NA
argumentative      1      0      0      0      0
vindictive         3     NA     NA     NA     NA
```

```
quarrelsome        2      1      1      0      0
unkind             2      1      1     NA     NA

$HT
[1] 0.4791647
```

The eight items do not meet the MS-CPM. Several significant violations were found. Only if four items were removed, no significant violations of MS-CPM remain; leaving a scale consisting of four items. For the four remaining items $H^T = 0.48$, suggesting moderate support for IIO.

Finally, method "IT" can be applied, yielding the following results:

```
R> summary(check.iio(X, method = "IT"))

$method
[1] "IT"

$item.summary
              ItemH #ac #vi #vi/#ac maxvi  sum sum/#ac xmax #xsig crit
impatient      0.43 210   2    0.01  0.08 0.13  0.0006 2.88     0   22
patient*       0.33 210   5    0.02  0.08 0.24  0.0012 2.88     0   31
irritable      0.50 210   5    0.02  0.06 0.20  0.0010 3.00     0   20
excitable      0.46 200   4    0.02  0.06 0.17  0.0009 1.38     0   13
argumentative  0.37 210   3    0.01  0.07 0.13  0.0006 3.60     0   28
vindictive     0.37 210  12    0.06  0.11 0.76  0.0036 5.14     2   69
quarrelsome    0.41 200   7    0.04  0.15 0.56  0.0028 7.53     2   78
unkind         0.31 210   8    0.04  0.15 0.61  0.0029 7.53     2   84

$backward.selection
              step 1 step 2 step 3
impatient          0      0      0
patient*           0      0      0
irritable          0      0      0
excitable          0      0      0
argumentative      0      0      0
vindictive         2      1     NA
quarrelsome        2      1      0
unkind             2     NA     NA

$HT
[1] 0.3626995
```

All three methods have selected "unkind" for removal, and both method "IT" and method "MSCPM" have selected item "vindictive". Method "MSCPM" was the strictest method. A pilot study confirmed the result that if IIO does not hold, then method "MSCPM" will select the most items for removal and method "MIIO" will select the least.

5. Discussion

With approximately four new versions per year since 2007, **mokken** is frequently upgraded and new features are added. This is not likely to stop. The to-do list already contains the implementation of several more existing and new methods. The following methods are implemented in **MSP** (see Molenaar and Sijtsma 2000) but not yet in **mokken**: MSA for different groups in the sample, especially the check of equal item (step) ordering per group; the *search extended* option in the AISP; the frequency distribution of the number of Guttman errors; and method "restsplit" for the investigation of nonintersection of item step response functions. Several aspects of **MSP** have not been implemented in **mokken** because standard R code can be used to obtain the results; for example, computing the test scores and providing a histogram of the test scores can be done easily using standard R code

```
R> test.scores <- apply(Y, 1, sum)
R> hist(test.scores)
```

and the number of negative H_{ij} values per item can be obtained by

```
R> Hij <- coefH(Y, se = FALSE)$Hij
R> apply(Hij, 1, function(x) sum(x < 0))
```

The following new methods will hopefully be implemented in **mokken**: An investigation of local independence based on Straat *et al.* (2011), a branch-and-bound algorithm for item selection based on Brusco *et al.* (2011), and confidence regions that are graphically depicted in the plots of estimated item response functions and item step response functions.

References

Andrich D (1978). "A Rating Formulation for Ordered Response Categories." *Psychometrika*, **43**, 561–573.

Birnbaum A (1968). "Some Latent Trait Models." In FM Lord, MR Novick (eds.), *Statistical Theories of Mental Test Scores*, pp. 397–424. Addison-Wesley, Reading.

Brusco MJ, Koehn HF, Steinley D (2011). "A Branch-and-Bound Max-Cardinality Algorithm for Exploratory Mokken Scale Analysis." Paper presented at the International Meeting of the Psychometric Society, Hong Kong, July, 2011.

Cronbach L (1951). "Coefficient Alpha and the Internal Structure of Tests." *Psychometrika*, **16**, 297–334.

Gough HG, Heilbrun AB (1980). *The Adjective Check List, Manual 1980 Edition.* Consulting Psychologists Press, Palo Alto, CA.

Grayson DA (1988). "Two-Group Classification in Latent Trait Theory: Scores With Monotone Likelihood Ratio." *Psychometrika*, **53**, 383–392.

Guttman L (1950). "The Basis for Scalogram Analysis." In SA Stoufer, L Guttman, EA Suchman, PF Lazersfeld, SA Star, JA Clausen (eds.), *Measurement and Prediction*, pp. 60–90. John Wiley & Sons, New York.

Hemker BT, Sijtsma K, Molenaar IW, Junker BW (1996). "Polytomous IRT Models and Monotone Likelihood Ratio of the Total Score." *Psychometrika*, **61**, 679–693.

Hemker BT, Sijtsma K, Molenaar IW, Junker BW (1997). "Stochastic Ordering Using the Latent Trait and the Sum Score in Polytomous IRT Models." *Psychometrika*, **62**, 331–347.

Huynh H (1994). "A New Proof for Monotone Likelihood Ratio for the Sum of Independent Bernoulli Random Variables." *Psychometrika*, **59**, 77–79.

Junker BW, Sijtsma K (2000). "Latent and Manifest Monotonicity." *Applied Psychological Measurement*, **24**, 65–81.

Kuijpers RE, Van der Ark LA, Croon MA (2011). "Computing Standard Errors and Confidence Intervals for Scalability Coefficients in Mokken Scale Analysis Using Marginal Models." Manuscript submitted for publication.

Ligtvoet R, Van der Ark LA, Bergsma WP, Sijtsma K (2011). "Polytomous Latent Scales for the Investigation of the Ordering of Items." *Psychometrika*, **76**, 200–216.

Ligtvoet R, Van der Ark LA, Te Marvelde JM, Sijtsma K (2010). "Investigating an Invariant Item Ordering for Polytomously Scored Items." *Educational and Psychological Measurement*, **70**, 578–595.

Linzer DA (2011). "Reliable Inference in Highly Stratiffed Contingency Tables: Using Latent Class Models as Density Estimators." *Political Analysis*, **19**, 173–187.

Lord FM, Novick MR (1968). *Statistical Theories of Mental Test Scores.* Addison-Wesley, Reading.

Masters G (1982). "A Rasch Model for Partial Credit Scoring." *Psychometrika*, **47**, 149–174.

Meijer RR, Baneke JJ (2004). "Analyzing Psychopathology Items: A Case for Nonparametric Item Response Theory Modeling." *Psychological Methods*, **9**, 354–368.

Mokken RJ (1971). *A Theory and Procedure of Scale Analysis.* De Gruyter, Berlin, Germany.

Mokken RJ, Lewis C (1982). "A Nonparametric Approach to the Analysis of Dichotomous Responses." *Applied Psychological Measurement*, **6**, 417–430.

Molenaar IW (1991). "A Weighted Loevinger H-Coefficient Extending Mokken Scaling to Multicategory Items." *Kwantitatieve Methoden*, **12**(37), 97–117.

Molenaar IW (1997). "Nonparametric Models for Polytomous Responses." In WJ Van der Linden, RK Hambleton (eds.), *Handbook of Modern Item Response Theory*, pp. 369–380. Springer-Verlag, New York.

Molenaar IW, Sijtsma K (1988). "Mokken's Approach to Reliability Estimation Extended to Multicategory Items." *Kwantitatieve Methoden*, **9**(28), 115–126.

Molenaar IW, Sijtsma K (2000). *User's Manual **MSP5** for Windows.* IEC ProGAMMA, Groningen.

Muraki E (1992). “A Generalized Partial Credit Model: Application of an EM Algorithm.” *Applied Psychological Measurement*, **16**, 159–177.

Rasch G (1960). *Probabilistic Models for Some Intelligence and Attainment Tests*. Nielsen and Lydiche, Copenhagen.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org/`.

Rosenbaum PR (1987). “Probability Inequalities for Latent Scales.” *British Journal of Mathematical and Statistical Psychology*, **40**, 157–168.

Samejima F (1969). “Estimation of Latent Ability Using a Response Pattern of Graded scores.” *Psychometrika Monograph*, **17**.

Schmitt N (1996). “Uses and Abuses of Coefficient Alpha.” *Psychological Assessment*, **8**, 350–353.

Sijtsma K (2009). “On the Use, the Misuse, and the Very Limited Usefulness of Cronbach’s Alpha.” *Psychometrika*, **74**, 107–120.

Sijtsma K, Hemker BT (1998). “Nonparametric Polytomous IRT Models for Invariant Item Ordering, With Results for Parametric Models.” *Psychometrika*, **63**, 183–200.

Sijtsma K, Junker BW (1996). “A Survey of Theory and Methods of Invariant Item Ordering.” *British Journal of Mathematical and Statistical Psychology*, **49**, 79–105.

Sijtsma K, Meijer RR, Van der Ark LA (2011). “Mokken Scale Analysis as Time Goes By: An Update for Scaling Practitioners.” *Personality and Individual Differences*, **50**, 31–37.

Sijtsma K, Molenaar IW (2002). *Introduction to Nonparametric Item Response Theory*. Sage, Thousand Oaks.

Sijtsma K, Molenaar WI (1987). “Reliability of Test Score in Nonparametric Item Response Theory.” *Psychometrika*, **52**, 79–97.

Smits IAM, Timmermans ME, Meijer RR (2011). “Assessing Dimensionality Through Mokken Scale Analysis: What Happens When Questionnaires Do Not Have Simple Structures?” Paper presented at the International Meeting of the Psychometric Society, Hong Kong, July, 2011.

Straat JH, Van der Ark LA, Sijtsma K (2011). “A New Scaling Procedure Based on Conditional Association for Assessing IRT Model Fit.” Paper presented at the International Meeting of the Psychometric Society, Hong Kong, July, 2011.

Straat JH, Van der Ark LA, Sijtsma K (2013). “Comparing Optimization Algorithms for Item Selection in Mokken Scale Analysis.” *Journal of Classification*. Forthcoming.

Tutz G (1990). “Sequential Item Response Models With an Ordered Response.” *British Journal of Mathematical and Statistical Psychology*, **43**, 39–55.

Van Abswoude AAH, Van der Ark LA, Sijtsma K (2004). "A Comparative Study of Test Data Dimensionality Assessment Procedures Under Nonparametric IRT Models." *Applied Psychological Measurement*, **28**, 3–24.

Van der Ark LA (2001). "Relationships and Properties of Polytomous Item Response Theory Models." *Applied Psychological Measurement*, **25**, 273–282.

Van der Ark LA (2005). "Stochastic Ordering of the Latent Trait by the Sum Score Under Various Polytomous IRT Models." *Psychometrika*, **70**, 283–304.

Van der Ark LA (2007). "Mokken Scale Analysis in R." *Journal of Statistical Software*, **20**(11), 1–19. URL http://www.jstatsoft.org/v20/i11/.

Van der Ark LA (2010a). "Computation of the Molenaar Sijtsma Statistic." In A Fink, B Lausen, W Seidel, A Ultsch (eds.), *Advances in Data Analysis, Data Handling and Business Intelligence*, pp. 775–784. Springer-Verlag, Berlin.

Van der Ark LA (2010b). "Getting Started with Mokken Scale Analysis in R." R package vignette, URL http://CRAN.R-project.org/package=mokken.

Van der Ark LA, Bergsma WP (2010). "A Note on Stochastic Ordering of the Latent Trait Using the Sum of Polytomous Item Scores." *Psychometrika*, **75**, 272–279.

Van der Ark LA, Croon MA, Sijtsma K (2008). "Mokken Scale Analysis for Dichotomous Items Using Marginal Models." *Psychometrika*, **73**, 183–208.

Van der Ark LA, Van der Palm DW, Sijtsma K (2011). "A Latent Class Approach to Estimating Test-Score Reliability." *Applied Psychological Measurement*, **35**, 380–392.

Van Onna MJH (2004). "Estimates of the Sampling Distribution of Scalability Coefficient H." *Applied Psychological Measurement*, **28**, 427–449.

Van Schuur WH (2011). *Ordinal Item Response Theory: Mokken Scale Analysis*. Sage, Thousand Oaks.

Vermunt JK, Van Ginkel JR, Van der Ark LA, Sijtsma K (2008). "Multiple Imputation of Categorical Data Using Latent Class Analysis." *Sociological Methodology*, **38**, 369–397.

Verweij AC, Sijtsma K, Koops W (1996). "A Mokken Scale for Transitive Reasoning Suited for Longitudinal Research." *International Journal of Behavioral Development*, **23**, 241–264.

Affiliation:

L. Andries van der Ark
Tilburg School of Social and Behavioral Sciences
Tilburg University

P. O. Box 90153,
5000 LE, Tilburg, The Netherlands
Email: A.vdArk@TilburgUniversity.edu
URL: http://www.tilburguniversity.edu/webwijs/show/?uid=a.vdark

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 48, Issue 5 *Submitted:* 2011-09-13
May 2012 *Accepted:* 2012-01-18