DAKS: An R Package for Data Analysis Methods in Knowledge Space Theory

Ali Ünlü
University of Dortmund

Anatol Sargin
University of Dortmund

Abstract

This introduction to the R package **DAKS** is based on the paper by Ünlü and Sargin (2010b) published in the *Journal of Statistical Software*.

Knowledge space theory is part of psychometrics and provides a theoretical framework for the modeling, assessment, and training of knowledge. It utilizes the idea that some pieces of knowledge may imply others, and is based on order and set theory. We introduce the R package **DAKS** for performing basic and advanced operations in knowledge space theory. This package implements three inductive item tree analysis algorithms for deriving quasi orders from binary data, the original, corrected, and minimized corrected algorithms, in sample as well as population quantities. It provides functions for computing population and estimated asymptotic variances of and one and two sample Z-tests for the *diff* fit measures, and for switching between test item and knowledge state representations. Other features are a function for computing response pattern and knowledge state frequencies, a data (based on a finite mixture latent variable model) and quasi order simulation tool, and a Hasse diagram drawing device. We describe the functions of the package and demonstrate their usage by real and simulated data examples.

Keywords: knowledge space theory, psychometrics, exploratory data analysis, maximum likelihood asymptotic theory, R.

1. Introduction

More than 50 years ago, Louis Guttman introduced his scalogram technique (Guttman 1944). The deterministic scalogram technique allows for linear orderings of persons (e.g., regarding their abilities) and items (e.g., regarding their difficulties). Since then, the Guttman model has been generalized in at least two directions. On the one hand, in a probabilistic and statistical direction, based on the Rasch (1960) model and generalized by Mokken (1971)'s monotone homogeneity model, a family of linear probabilistic models (item response theory, IRT; e.g., Van der Linden and Hambleton 1997) has emerged, retaining the linearity of person and item orderings. On the other hand, in a deterministic and order-theoretic direction, starting with Airasian and Bart (1973) and Bart and Krus (1973), a family of nonlinear deterministic models (knowledge space theory, KST; e.g., Doignon and Falmagne 1985, see Section 2) has been developed, weakening the linearity of person and item orderings to allow for incomparabilities among persons and items. In KST, persons are represented by collections of items of a domain they are capable of mastering.[1] Persons can be incomparable with respect to set-inclusion.

[1]Throughout this paper, mastery of an item stands for a subject's true, unobservable knowledge of the

Items, in turn, are assumed to be ordered, for instance, with respect to a hierarchy of mastery dependencies. Items can be incomparable with respect to that hierarchy. In IRT, on the other hand, persons and items are, for instance, represented by single real numbers, ability and difficulty parameters, respectively. Persons and items are linearly ordered with respect to the natural ordering of the real numbers. Conceptually speaking, KST may be viewed as a more 'qualitative, behavioral' approach, unlike IRT, as a 'quantitative, statistical' approach.

KST and IRT are split directions of psychological test theories, and there is a major interest in trying to conflate these test theories. What one ideally would like to have is a unified framework keeping the strengths and at the same time avoiding the drawbacks of both theories. In Section 5, we describe what the KST models can do that the IRT models cannot do, and vice versa (cf. Ünlü 2007). KST and IRT have been partly compared at a theoretical level (Stefanutti 2006; Stefanutti and Robusto 2009; Ünlü 2006, 2007). Using the R (R Development Core Team 2010) language and environment for statistical computing and graphics as an interface between these theories may prove valuable in comparing them at a computational level. R gives users the possibility to include own software packages for handling specific tasks. There are a number of R packages available for IRT; for instance, **ltm** (Rizopoulos 2006) or **mokken** (Van der Ark 2007). But there aren't for KST, and the present R package **DAKS** aims at providing a basis for computational work in the so far combinatorial theory of knowledge spaces. Implementing KST procedures in R can help to bring together KST and IRT.

KST was introduced by Doignon and Falmagne (1985). Most of the theory is presented in a monograph by Doignon and Falmagne (1999); for applications see Albert and Lukas (1999), and for survey articles see Doignon and Falmagne (1987), Falmagne (1989b), and Falmagne, Koppen, Villano, Doignon, and Johannesen (1990). A comprehensive bibliography on KST, including many references on empirical applications of KST, by C. Hockemeyer (University of Graz, Austria) can be retrieved from `http://wundt.kfunigraz.ac.at/kst.php`. KST provides a theoretical framework for the modeling, assessment, and training of knowledge. This theory utilizes the idea that some pieces of knowledge may imply others. For instance, the mastery of a test question may imply the mastery of other test questions. Implications between pieces of knowledge are modeled in KST by order and set theoretic structures. Based on such a framework, KST has been successfully applied for computerized adaptive assessment and training; for example, see the ALEKS system (`http://www.aleks.com/`), a Web-based, artificially intelligent assessment and learning system.

However, KST models can only be successfully applied if the latent implications underlying the items are sufficiently known. Therefore a crucial problem in KST is the empirical derivation of the implications between items using the data. Three inductive item tree analysis (IITA) algorithms have been proposed for deriving implications from dichotomous data: the original IITA algorithm (Schrepp 2003), and the corrected and minimized corrected IITA algorithms (Sargin and Ünlü 2009; Ünlü and Sargin 2010a). These methods constitute the main part of the package **DAKS** and are implemented in sample and population quantities. Besides the three IITA algorithms, the package **DAKS** also provides functions for computing population and estimated asymptotic variances of and one and two sample Z-tests for the fit measures, and for switching between test item and knowledge state representations. Other features are a function for computing response pattern and knowledge state frequencies, a data and quasi

solution to the item (latent level); solving an item stands for the observed response of a subject to the item (manifest level).

order simulation tool, and a Hasse diagram (see Footnote 6) drawing device.

Currently available software implementing the original IITA algorithm is **ITA 2.0** by Schrepp (2006). Compared to this stand-alone software that runs only on Windows, the package **DAKS** is embedded in the comprehensive R computing environment and provides much more functionalities such as more flexible input/output features. In particular, the corrected and minimized corrected IITA algorithms are only implemented in the package **DAKS**.

In Section 2, the basic deterministic and probabilistic concepts of KST and the three IITA algorithms are reviewed. In Section 3, the package **DAKS** is presented and its functions are explained. In Section 4, the package **DAKS** is demonstrated using real and simulated data. In Section 5, we conclude with a summary, some suggestions for future implementations of the package, and general remarks about the interplay between KST and IRT.

2. Knowledge space theory and data analysis methods

We briefly recapitulate the basic concepts of KST relevant for this work and the three IITA algorithms. Details can be found in the respective references afore mentioned.

2.1. Basic concepts of knowledge space theory

Assume a set Q of m dichotomous items. Mastering an item $j \in Q$ may imply mastering another item $i \in Q$. If no response errors are made, these implications, $j \to i$, entail that only certain response patterns (represented by subsets of Q) are possible. Those response patterns are called knowledge states, and the set of all knowledge states (including $\emptyset$ and Q) is called a knowledge structure, and denoted by $\mathcal{K}$. The knowledge structure $\mathcal{K}$ is a subset of 2^Q, the power set of Q. Implications are assumed to form a quasi order, that is, a reflexive, transitive binary relation, $\sqsubseteq$ on the item set Q. In other words, an implication $j \to i$ stands for the pair $(i, j) \in \sqsubseteq$, also denoted by $i \sqsubseteq j$. Quasi orders are referred to as surmise relations in KST. Theoretically, a surmise relation can even consist of only the reflexive item pairs. No item then implies another, and any response pattern is consistent with the surmise relation. The IITA algorithms do cover this case as well, because this is a special type of surmise relation. Practically, however, having no implications between, for a specific purpose, reasonably constructed items is virtually impossible. In general, items measure some common latent traits and so they do correlate to some degree. Moreover, the special case of a chain hierarchy (see Footnote 3) among the test items is covered by the IITA procedures as well.

A possible application is an aptitude test, where participants can solve (coded 1) or fail to solve (coded 0) a question. In this paper, the latter interpretation is used to illustrate the IITA algorithms.

Implications are latent and not directly observable, due to random response errors. A person who is actually unable to solve an item, but does so, makes a lucky guess. On the other hand, a person makes a careless error, if he fails to solve an item which he masters. A probabilistic extension of the knowledge structure model covering random response errors is the basic local independence model.[2]

[2]The basic local independence model is assumed to hold throughout this paper. In Section 4.2, we use that probability model with prespecified/known parameter values for simulating the data. Based on the simulated data, we can check and compare the IITA algorithms. In particular, we do not estimate the latent parameters

Basic local independence model

A quadruple $(Q, \mathcal{K}, p, r)$ is called a basic local independence model (BLIM) if and only if

1. $(Q, \mathcal{K})$ is a knowledge structure,
2. p is a probability distribution on $\mathcal{K}$, that is, $p : \mathcal{K} \to]0, 1[\,, K \mapsto p(K)$, with $p(K) > 0$ for any $K \in \mathcal{K}$, and $\sum_{K \in \mathcal{K}} p(K) = 1$,
3. r is a response function for $(Q, \mathcal{K}, p)$, that is, $r : 2^Q \times \mathcal{K} \to [0, 1]$, $(R, K) \mapsto r(R, K)$, with $r(R, K) \geq 0$ for any $R \in 2^Q$ and $K \in \mathcal{K}$, and $\sum_{R \in 2^Q} r(R, K) = 1$ for any $K \in \mathcal{K}$,
4. r satisfies local independence, that is,

$$r(R, K) = \prod_{q \in K \setminus R} \beta_q \cdot \prod_{q \in K \cap R} (1 - \beta_q) \cdot \prod_{q \in R \setminus K} \eta_q \cdot \prod_{q \in Q \setminus (R \cup K)} (1 - \eta_q),$$

with two constants $\beta_q, \eta_q \in [0, 1[$ for each $q \in Q$, respectively called careless error and lucky guess probabilities at q.

Here, $K \setminus R := \{q \in Q : q \in K \text{ and } q \notin R\}$, $K \cap R := \{q \in Q : q \in K \text{ and } q \in R\}$, $R \setminus K := \{q \in Q : q \in R \text{ and } q \notin K\}$, and $Q \setminus (R \cup K) := \{q \in Q : q \notin R \text{ and } q \notin K\}$. The items in $K \setminus R$, $K \cap R$, $R \setminus K$, and $Q \setminus (R \cup K)$ are mastered but not solved (careless error), mastered and solved (no careless error), solved but not mastered (lucky guess), and not solved and not mastered (no lucky guess), respectively.

Let n be the sample size. The data are the observed absolute counts of response patterns $R \subset Q$. Let D denote the corresponding $n \times m$ data matrix of 0/1 item scores. The data are assumed to be multinomially distributed over 2^Q. Let $\rho(R)$ denote the (unknown) true probability of occurrence of a response pattern R. The BLIM is based on the following assumptions. To each knowledge state $K \in \mathcal{K}$ is attached a probability $p(K)$ measuring the likelihood that a respondent is in state K. For a manifest response pattern $R \subset Q$ and a latent knowledge state $K \in \mathcal{K}$, $r(R, K)$ specifies the conditional probability of response pattern R for a respondent in state K. The item responses of a respondent are assumed to be independent given the knowledge state of the respondent (local independence). The response error, that is, careless error and lucky guess, probabilities β_q and η_q are attached to the items and do not vary with the knowledge states.

The BLIM allows expressing the occurrence probabilities $\rho(R)$ of response patterns R by means of the model parameters $p(K)$ and β_q, η_q:

$$\rho(R) = \sum_{K \in \mathcal{K}} \left\{ \left[\prod_{q \in K \setminus R} \beta_q \right] \cdot \left[\prod_{q \in K \cap R} (1 - \beta_q) \right] \cdot \left[\prod_{q \in R \setminus K} \eta_q \right] \cdot \left[\prod_{q \in Q \setminus (R \cup K)} (1 - \eta_q) \right] \right\} p(K).$$

Remarks regarding the basic local independence model

The BLIM is fundamental in KST, in the sense that most of the KST probabilistic models are special cases of this model (Doignon and Falmagne 1999). Viewing the knowledge states as the

(probabilities) of and assess model fit for the basic local independence model. The focus and advantage of the exploratory IITA methods lies in the data-analytic derivation of a knowledge structure solely based on the manifest parameters (probabilities) of the multinomial sampling distribution for the data. The multinomial distribution is the true saturated model, and its parameters can easily be estimated using the corresponding sample analogs.

latent classes, the BLIM can be seen as a constrained latent class model with $|\mathcal{K}|$ latent classes ($|\mathcal{K}|$, the size of $\mathcal{K}$) and two conditional response probabilities per test item. It is important to note that the latent classes $K \in \mathcal{K}$ possess an inner structure composed of the indicators, which determines the constraints imposed on the conditional class probabilities. The idea expressed in the definition of the BLIM is not a new one and goes back to traditional latent class measurement or scaling models such as the Proctor (1970) model, the Dayton and Macready (1976) intrusion–omission model, and more generally, the Lazarsfeld and Henry (1968) latent distance model. They originated as probabilistic generalizations of the deterministic, linear Guttman (1944) model. The BLIM is a de-linearized latent distance model. De-linearized here means that the knowledge structure $\mathcal{K}$ is not necessarily linearly ordered with respect to set-inclusion (cf. Footnote 3), as is the case for the traditional latent class scaling models. For a description of the relationships of these models, and more generally, for a description of the connection between KST and latent class analysis (including inference methodologies), see Ünlü (2010) (cf. also Schrepp 2005).

The BLIM is a restricted latent class model and the most general model used in KST. The dynamic, based on stochastic processes, KST stochastic learning paths systems are special cases of the BLIM (Doignon and Falmagne 1999; Falmagne 1989a; Falmagne *et al.* 1990). They are obtained by further restricting, besides combinatorial constraints on the knowledge structure, the state/class probabilities of the BLIM based on postulated learning mechanisms describing successive transitions of subjects, over time, from the state $\emptyset$ to the state Q. The models considered in Schrepp (2005) and Stefanutti and Robusto (2009) are the BLIM with the error parameters being a priori restricted to sub-intervals of the unit interval. In the current version of the package **DAKS**, these special cases of the BLIM are not supported and represent useful features that may be added in future improvements of the package.

The number of independent parameters of the BLIM is $2|Q| + (|\mathcal{K}| - 1)$. Since $|\mathcal{K}|$ generally tends to be prohibitively large in practice, parameter estimation and model testing based on classical maximum likelihood methodology are not feasible in general (for details, see Ünlü 2006, 2007). For instance, in an experiment by Kambouri (1991) (see also Kambouri, Koppen, Villano, and Falmagne 1994), reviewed in Doignon and Falmagne (1999), the number of knowledge states ranges from several hundreds to several thousands (for 50 items). In such cases, without any restrictions, it may be infeasible to obtain reliable estimates of the several hundreds to several thousands of model parameters, given a specific knowledge structure. This is why exploratory methods such as the IITA algorithms are important in KST. Exploratory methods can be applied without having to estimate the latent parameters of and assess model fit for the BLIM (cf. also Footnote 2).

Birkhoff's theorem

A knowledge structure closed under union and intersection is called a quasi ordinal knowledge space. Quasi ordinal knowledge spaces and surmise relations are equivalent formulations. According to Birkhoff (1937)'s theorem, there exists a one-to-one correspondence between the collection of all quasi ordinal knowledge spaces $\mathcal{K}$ on a domain Q, and the collection of all surmise relations $\sqsubseteq$ on Q. Such a correspondence is defined through the two equivalences:

$$\begin{aligned} p \sqsubseteq q \quad &:\Longleftrightarrow \quad [\forall K \in \mathcal{K} : \{q \in K \Longrightarrow p \in K\}]\,, \\ K \in \mathcal{K} \quad &:\Longleftrightarrow \quad [\forall (p \sqsubseteq q) : \{q \in K \Longrightarrow p \in K\}]\,. \end{aligned}$$

This theorem is important from a practical point of view. Though the quasi ordinal knowledge

space and surmise relation models are empirically interpreted at the different levels of persons and items, they are connected with each other mathematically, through Birkhoff's theorem. This theorem is realized in the package **DAKS** using two functions for switching between test item and knowledge state representations (see Section 3.2).

2.2. Inductive item tree analysis algorithms

The functions of the package **DAKS** realizing the IITA algorithms in sample and population quantities are described in Section 3.2. Their usage by real and simulated data examples is demonstrated in Section 4.

Inductive item tree analysis algorithms in sample values

The three IITA algorithms are exploratory methods for extracting surmise relations from data. In each algorithm, competing binary relations are generated, and a fit measure is computed for every relation in order to find the quasi order that fits the data best. In the following, the methods are briefly reviewed.

Algorithms. For the original IITA version (Schrepp 2003) the algorithm is:

1. For two items i, j, the value $b_{ij} := |\{R \in D | i \notin R \wedge j \in R\}|$ is the number of counterexamples, that is, the number of observed response patterns in the data matrix D contradicting $j \to i$. Based on these values, binary relations $\sqsubseteq_L$ for $L = 0, \ldots, n$ are defined as follows.

   1a. Let $i \sqsubseteq_0 j :\Leftrightarrow b_{ij} = 0$. The relation $\sqsubseteq_0$ is a quasi order.

   1b. Construct inductively: Assume $\sqsubseteq_L$ is transitive. Define the set $S^{(0)}_{L+1} := \{(i,j) | b_{ij} \leq L+1 \wedge i \not\sqsubseteq_L j\}$. This set consists of all item pairs that are not already contained in the relation $\sqsubseteq_L$ and have at most $L+1$ counterexamples. From $S^{(0)}_{L+1}$, exclude those item pairs that cause an intransitivity in $\sqsubseteq_L \cup S^{(0)}_{L+1}$; the remaining item pairs (of $S^{(0)}_{L+1}$) are referred to as $S^{(1)}_{L+1}$. Then, from the item pairs in $S^{(1)}_{L+1}$, those are excluded that cause an intransitivity in $\sqsubseteq_L \cup S^{(1)}_{L+1}$, and the remaining item pairs (of $S^{(1)}_{L+1}$) are referred to as $S^{(2)}_{L+1}$. This process continues iteratively, say k times, until no intransitivity is caused.

   1c. The generated relation $\sqsubseteq_{L+1} := \sqsubseteq_L \cup S^{(k)}_{L+1}$ is a quasi order by construction. Hence $\sqsubseteq_L$ for $L = 0, \ldots, n$ are quasi orders. They constitute the selection set of the IITA procedure.

2. The coefficient $\mathit{diff}_o(\sqsubseteq_L, D)$ is used to assess the fit of each quasi order $\sqsubseteq_L$ to the binary data matrix D (see below).

3. Choose the quasi order with minimum $\mathit{diff}_o(\sqsubseteq_L, D)$ value.

For the corrected and minimized corrected IITA versions (Sargin and Ünlü 2009) the algorithms are:

1. The generation of the selection set of quasi orders is the same as in the original IITA version.

2. The coefficients $diff_c(\sqsubseteq_L, D)$ and $diff_{mc}(\sqsubseteq_L, D)$ are used to assess the fit of each quasi order $\sqsubseteq_L$ to the binary data matrix D (see below), respectively.
3. Choose the quasi orders with minimum $diff_c(\sqsubseteq_L, D)$ and $diff_{mc}(\sqsubseteq_L, D)$ values, respectively.

Fit measures. The *diff* fit measures $diff_o$, $diff_c$, and $diff_{mc}$ are defined by

$$diff\,(\sqsubseteq, D) = \frac{1}{m(m-1)} \sum_{i \neq j} (b_{ij} - b^*_{ij})^2,$$

where corresponding estimates b^*_{ij} are used, varying from algorithm to algorithm. We describe the computation of these estimates. The estimates b^*_{ij} are obtained based on a single error probability.

In the original IITA version this single error rate is given by

$$\gamma_\sqsubseteq = \frac{1}{|\sqsubseteq| - m} \sum_{i \sqsubseteq j, i \neq j} \frac{b_{ij}}{p_j n}.$$

If $(i,j) \in \sqsubseteq$, the expected number of counterexamples is estimated by $b^*_{ij} = \gamma_\sqsubseteq p_j n$. If $(i,j) \notin \sqsubseteq$, no dependency between the two items is assumed, and the estimate $b^*_{ij} = (1-p_i)p_j n(1-\gamma_\sqsubseteq)$ is used. In this formula, $(1-p_i)p_j n$ is the usual probability for two independent items, and the factor $1-\gamma_\sqsubseteq$ is assumed to state that no random error occurred. As discussed in Sargin and Ünlü (2009), the main criticism on the original algorithm is on the used estimates b^*_{ij}.

In Sargin and Ünlü (2009), it is shown that this estimation scheme leads to methodological inconsistencies, and corrected estimators avoiding the inconsistencies of the original algorithm are proposed. Two problems arise in the calculation of the estimates of the original algorithm. For $(i,j) \notin \sqsubseteq$, the estimate used in the original algorithm is $b^*_{ij} = (1-p_i)p_j n(1-\gamma_\sqsubseteq)$. But the original algorithm does not take two different cases into account, namely $(j,i) \notin \sqsubseteq$ and $(j,i) \in \sqsubseteq$. In the first case, independence holds, and a corrected estimator is $b^*_{ij} = (1-p_i)p_j n$. In the second case, independence cannot be assumed, as $j \sqsubseteq i$. A corrected estimator b^*_{ij} in this case is $(p_j - p_i + \gamma_\sqsubseteq p_i)n$ (see Sargin and Ünlü 2009), instead of $(1-p_i)p_j n(1-\gamma_\sqsubseteq)$.

In the corrected IITA version the same $\gamma_\sqsubseteq$ and $b^*_{ij} = \gamma_\sqsubseteq p_j n$ for $(i,j) \in \sqsubseteq$ are used. The choice for b^*_{ij} in the case of $(i,j) \notin \sqsubseteq$ now depends on whether $(j,i) \notin \sqsubseteq$ or $(j,i) \in \sqsubseteq$. If $(i,j) \notin \sqsubseteq$ and $(j,i) \notin \sqsubseteq$, set $b^*_{ij} = (1-p_i)p_j n$. If $(i,j) \notin \sqsubseteq$ and $(j,i) \in \sqsubseteq$, set $b^*_{ij} = (p_j - p_i + \gamma_\sqsubseteq p_i)n$.

In the minimized corrected IITA version the corrected estimators b^*_{ij} as in the $diff_c$ coefficient are used. Minimizing the *diff* expression as a function of the error probability $\gamma_\sqsubseteq$ gives $\gamma_\sqsubseteq = -\frac{x_1+x_2}{x_3+x_4}$, where

$$\begin{aligned}
x_1 &= \sum_{i \not\sqsubseteq j \,\wedge\, j \sqsubseteq i} -2b_{ij}p_i n + 2p_i p_j n^2 - 2p_i^2 n^2, \\
x_2 &= \sum_{i \sqsubseteq j} -2b_{ij}p_j n, \\
x_3 &= \sum_{i \not\sqsubseteq j \,\wedge\, j \sqsubseteq i} 2p_i^2 n^2, \\
x_4 &= \sum_{i \sqsubseteq j} 2p_j^2 n^2
\end{aligned}$$

(for details, see Sargin and Ünlü 2009). This error probability can now be used for an alternative IITA procedure, in which a minimized *diff* value is computed for every quasi order. The idea underlying the minimized corrected IITA version is to use the corrected estimators and to optimize the fit criterion. The fit measure then favors quasi orders that lead to smallest minimum discrepancies, or equivalently, largest maximum matches, between the observed and expected numbers of counterexamples.

General remarks. Mathematical considerations and comparisons based on simulated and real data examples reported by Sargin and Ünlü (2009) and Ünlü and Sargin (2010a) based on sample and population values, respectively, suggest using the minimized corrected IITA version as the prior choice. For instance, in the extensive simulation studies in Sargin and Ünlü (2009) and Ünlü and Sargin (2010a), overall the minimized corrected IITA algorithm performs best, second comes the corrected IITA algorithm, and worst is the original IITA algorithm, with respect to all of the considered summary statistics. The summary statistics according to which the IITA algorithms have been compared are, for example, the sample and population symmetric differences at the levels of items and knowledge states, and the ranks of the underlying quasi orders in the ordered lists of population *diff* values. Moreover, similar results are obtained for the corrected and minimized corrected algorithms, with a slight advantage for the latter. For each of the considered summary statistics, the original IITA algorithm shows considerably bad results for larger error probabilities. The original IITA algorithm should only be used specifically for datasets with very few underlying knowledge states and when the error rates are very low. See also the "General remarks" in "IITA analyses of the PISA data" of Section 4.1.

Inductive item tree analysis algorithms in population values

In Ünlü and Sargin (2010a), we introduce the population analogs of the *diff* fit measures, interpret the coefficients as maximum likelihood estimators (MLEs) for the corresponding population values, and show for the estimators the quality properties of asymptotic efficiency, asymptotic normality, asymptotic unbiasedness, and consistency. This is briefly reviewed next.

Cautionary notes. Why do we need population variants of the *diff* coefficients? What is the BLIM for? What is the connection between the fit of a quasi order to the data assessed in terms of the *diff* coefficients on the one hand and the BLIM and parameters $\rho(R)$ on the other?

The original, corrected, and minimized corrected IITA algorithms with their respective *diff* fit measures have been proposed for building quasi orders from dichotomous data. So far, they have been treated descriptively, without examining a theory that may underlie these procedures. A statistical theory, however, requires a population based approach. Theoretical considerations in population quantities are important. Supposing the population completely to be known is the way to begin with in constructing sound fit measures for quasi orders. After having provided justification for a measure in a known population, one has to consider sampling problems concerning estimation and testing (Goodman and Kruskal 1979). For instance, based on the package **DAKS** (see Section 3.2) we now can perform an approximate significance test to test whether the population *diff* value for one quasi order is greater than

the population value obtained for another, which is the crucial hypothesis to be tested when choosing among competing quasi orders. Literature on the IITA algorithms has dealt with samples rather than a population. For a purely descriptive approach, however, statistical estimation and testing do not make sense.

The BLIM is a fairly general and realistic probability model, which explains the responses to test items of individuals in certain knowledge states. It is *the* probabilistic generalization of the knowledge structure model that is used in KST. A knowledge structure, more precisely a quasi ordinal knowledge space, corresponds to a surmise relation (cf. Birkhoff's theorem). In this sense, the BLIM is a fairly realistic probabilistic generalization of the surmise relation model, which takes into account deviations from the latent true implications between the items according to random response errors (careless errors and lucky guesses). Therefore the BLIM is the probability model assumed to hold throughout this paper.

As mentioned in Section 2.1, the advantage of the exploratory IITA methods and the *diff* coefficients lies in their computation, which is solely based on the manifest probabilities of the multinomial sampling distribution for the data. The population *diff* coefficients are functions of the multinomial cell probabilities $\rho(R)$ $(R \subset Q)$ (see below). These probabilities can easily be estimated using the corresponding sample analogs. In this way, one avoids having to estimate the latent parameters of the BLIM, which is not feasible in general (Section 2.1). This is the reason why exploratory methods such as the IITA algorithms are important in KST. Although they are only indirectly related to the latent parameters of the BLIM—they are exploratory procedures operating on the manifest probabilities $\rho(R)$, and theoretically at least, they can be computed under any model for $\rho(R)$—the IITA methods provide good results when applied to response data arising from such a realistic response model as the BLIM and hence represent a different approach to solving the problem of deriving a knowledge structure data-analytically (for details, see Sargin and Ünlü 2009; Ünlü and Sargin 2010a).

The occurrence probabilities $\rho(R)$ of response patterns $R \subset Q$ provide the connection between the BLIM and the *diff* coefficients. The BLIM expresses the occurrence probabilities $\rho(R)$ by means of the model parameters $p(K)$ $(K \in \mathcal{K})$ and β_q, η_q $(q \in Q)$. Having specified the parameters of the BLIM as the data generating model in simulations, as a consequence the probabilities $\rho(R)$ $(R \subset Q)$ are determined as well. By calculating the *diff* coefficients based on these true values we obtain the population values of the coefficients.

Population coefficients. Consider the transformed sample *diff* coefficients $\mathit{diff} := \mathit{diff}/n^2$. The division is necessary to cancel out sample size n in replacements of sample quantities with population quantities. Given the multinomial probability distribution on the set of all response patterns (see Section 2.1), make the following replacements in the arguments, b_{ij} and p_i, of the sample *diff* coefficients:

$$\begin{aligned} \frac{b_{ij}}{n} &\rightarrow P(i=0, j=1) = \sum_{R \in 2^Q, i \notin R \wedge j \in R} \rho(R), \\ p_i &\rightarrow P(i=1) = \sum_{R \in 2^Q, i \in R} \rho(R). \end{aligned}$$

This gives three population *diff* coefficients corresponding to the sample *diff* coefficients. The population *diff* coefficients are functions of the cell probabilities $\rho(R)$ $(R \subset Q)$ of the multinomial distribution.

The formulations of the population *diff* coefficients are straightforward. We have

$$\mathit{diff}\,(\sqsubseteq, \{\rho(R)\}_{R\in 2^Q}) = \frac{1}{m(m-1)} \sum_{i\neq j} (P(i=0, j=1) - P^*(i=0, j=1))^2,$$

where corresponding theoretical probabilities $P^*(i=0, j=1)$ are used, varying from algorithm to algorithm. In the population corrected IITA version, for instance, the population error rate is given by

$$\gamma_{\sqsubseteq} = \frac{1}{|\sqsubseteq| - m} \sum_{i\sqsubseteq j, i\neq j} \frac{P(i=0, j=1)}{P(j=1)},$$

and if $(i, j) \in \sqsubseteq$ for example, the theoretical probability is $P^*(i=0, j=1) = \gamma_{\sqsubseteq} P(j=1)$. As defined above, $P(i=0, j=1) = \sum_{R\in 2^Q, i\notin R \wedge j\in R} \rho(R)$ and $P(j=1) = \sum_{R\in 2^Q, j\in R} \rho(R)$. Since the BLIM expresses the cell probabilities $\rho(R)$ by means of the model parameters, having specified the parameters of the BLIM, these probabilities are also determined. The population values of the *diff* coefficients are based on these true values.

Maximum likelihood estimators. The sample *diff* coefficients, as defined in sample values before,

$$\mathit{diff}\,(\sqsubseteq, D) = \frac{1}{m(m-1)} \sum_{i\neq j} (b_{ij} - b^*_{ij})^2$$

are the obvious sample analogs of these population fit measures. They are reobtained by replacing the arguments $\rho(R)$ of the population *diff* measures with the MLEs $n(R)/n$ of the multinomial distribution, where $n(R)$ are the absolute counts of response patterns $R \in 2^Q$. That is, the sample *diff* coefficients are equal to the respective population *diff* coefficients evaluated at the MLEs $n(R)/n$. Therefore, according to the invariance property of MLEs (e.g., Casella and Berger 2002), the sample *diff* coefficients (as defined in sample values before) are the MLEs for the corresponding population *diff* coefficients.

Asymptotic properties. The MLE for the multinomial distribution fulfills required regularity conditions and is asymptotically efficient (e.g., Casella and Berger 2002). The population *diff* coefficients are differentiable functions of the multinomial cell probabilities $\rho(R)$; therefore the sample *diff* coefficients are asymptotically efficient, asymptotically normal, asymptotically unbiased, and consistent estimators for the population values (Ünlü and Sargin 2010a).

3. Implementation in the package DAKS

In this section, we describe how surmise relations and knowledge structures are implemented, and discuss the functions of this package.

3.1. Surmise relations and knowledge structures in DAKS

A quasi order is a set of tuples, where each tuple is a pair (i, j) representing the implication $j \to i$. This is implemented in **DAKS** using the package **sets** (Meyer and Hornik 2009). The latter, in combination with the package **relations** (Hornik and Meyer 2010), are utilized in **DAKS**, because they provide useful functions for operating with surmise relations and knowledge structures. The following R output shows an example quasi order:

```
{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)}
```

or

```
{(1L, 2L), (1L, 3L), (1L, 4L), (2L, 3L), (2L, 4L), (3L, 4L)}
```

This code is to be read: item 1 is implied by items 2, 3, and 4, item 2 is implied by items 3 and 4, and item 3 is implied by item 4. This gives the chain $4 \to 3 \to 2 \to 1$.[3] Note that in the second code line an item i is represented by iL. This transformation takes place internally in the packages **sets** or **relations**, but it does not have any influence. Both representations are equal (see pp. 306–307 in R Development Core Team 2010, for how R parses numeric constants):

```
R> 1 == 1L
```

```
[1] TRUE
```

Note that reflexive pairs are not shown in order to reveal implications between different items only, and to save computing time. Surmise relations always contain all reflexive pairs, and these are included whenever required by the package **DAKS**.

A knowledge structure is implemented as a binary matrix, where rows and columns stand for knowledge states and items, respectively. Each entry of the matrix, 1 or 0, represents mastering or not mastering an item in a corresponding state. The following R output shows the knowledge structure corresponding to the above quasi order:

```
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    1    0    0    0
[3,]    1    1    0    0
[4,]    1    1    1    0
[5,]    1    1    1    1
```

3.2. Functions of the package DAKS

We introduce the functions of the package **DAKS**. The main functions are for performing the IITA algorithms, in sample and population values. We also present more minor auxiliary (used for implementing the IITA algorithms) and utility functions of the package. Examples of how to use the functions are given in Section 4, where we also illustrate the connections between the functions.

Main functions for performing the IITA algorithms in sample values

[3] A chain or linearly ordered set is any partially ordered set (reflexive, transitive, and antisymmetric binary relation) $(P, \mathcal{P})$ satisfying the property of "linearity," that is, for all $p_1, p_2 \in P$, $p_1 \mathcal{P} p_2$ or $p_2 \mathcal{P} p_1$.

Generating automatically the set of competing quasi orders. The main function of the package **DAKS** that can be used to perform one of the original, corrected, and minimized corrected IITA procedures selectively (Section 2.2) is:

```
iita(dataset, v)
```

Whereas for the three IITA functions `orig_iita`, `corr_iita`, and `mini_iita` described subsequently selection sets of competing quasi orders have to be passed via an argument manually, the function `iita` automatically generates a selection set from the `dataset` using the inductive generation procedure implemented in the auxiliary function `ind_gen` (see below). The parameter `v` specifies the IITA algorithm to be performed: `v = 1` (minimized corrected), `v = 2` (corrected), and `v = 3` (original). The function `iita` returns, besides the *diff* values corresponding to the inductively generated quasi orders, the derived solution quasi order (with minimum *diff* value) under the selected algorithm, the estimated error rate corresponding to the best fitting quasi order, the index of the solution quasi order in the selection set, and an index specifying the used algorithm. In case of ties in minimum *diff* value, a quasi order with smallest size is returned. In general, the minimized corrected version gives the best results, hence it is suggested to use this version.

The function `iita` automatically generates a selection set from the data using the inductive generation procedure implemented in `ind_gen` (see below), and calls one of the following three IITA functions for computing the *diff* values. The approach using `iita` is common so far in KST, where the inductive data analysis methods have been utilized for exploratory derivations of quasi orders from data. The functions `orig_iita`, `corr_iita`, and `mini_iita`, on the other hand, can be used to select among surmise relations for instance obtained from querying experts or from competing psychological theories.

Passing manually the set of competing quasi orders. Three functions of the package **DAKS** realizing the original, corrected, and minimized corrected IITA algorithms separately (Section 2.2) are, in respective order:

```
orig_iita(dataset, A)
corr_iita(dataset, A)
mini_iita(dataset, A)
```

These functions perform the respective IITA procedures using the `dataset` and the list `A` of prespecified competing quasi orders. The set of competing quasi orders must be passed via the argument `A` manually, so any selection set of surmise relations can be used. In all three functions, the number of estimated counterexamples (according to each algorithm) and the number of observed counterexamples using the auxiliary function `ob_counter` (see below) are computed, and the vectors of the *diff* values

$$\mathit{diff}(\sqsubseteq, D) = \frac{1}{m(m-1)} \sum_{i \neq j} (b_{ij} - b^*_{ij})^2$$

and of the error rates (computed within each algorithm) corresponding to the competing quasi orders in `A` are returned (cf. Section 2.2).

Main functions for performing the IITA algorithms in population values

The package **DAKS** also contains functions which provide the basis for statistical inference methodology (cf. Section 5).

Population IITA algorithms. The population analog of the previous function that can be used to perform one of the three IITA algorithms in population quantities (in a known population) selectively is:

```
pop_iita(imp, ce, lg, items, dataset = NULL, A = NULL, v)
```

Compared to `iita`, this function implements the three IITA algorithms in population, not sample, quantities: `v = 1` (minimized corrected), `v = 2` (corrected), and `v = 3` (original). See "Inductive item tree analysis algorithms in population values" in Section 2.2 for details. The argument `imp` specifies a surmise relation, and `items` gives the number of items of the domain taken as basis for `imp`. The knowledge structure corresponding to `imp` is equipped with the careless error `ce` and lucky guess `lg` probabilities and the uniform distribution on the knowledge states, and is the known BLIM underlying the population. From this BLIM the occurrence probabilities $\rho(R)$ of response patterns $R \subset Q$ can be computed, and the algorithms can be performed in population values. If `dataset = NULL` and `A = NULL`, a set of competing quasi orders is constructed based on a population analog of the inductive generation procedure implemented in sample quantities in `ind_gen` (see below). If the `dataset` is specified explicitly, that data are used to generate the set of competing quasi orders based on the sample version of the inductive generation procedure. If the selection set `A` of quasi orders is specified explicitly, this is used as the set of competing quasi orders. (Specifying both `dataset` and `A` gives an error.) This function returns the population *diff* values corresponding to the inductively generated quasi orders, all possible response patterns with their population probabilities of occurrence, the population $\gamma_{\sqsubseteq}$ rates corresponding to the inductively generated quasi orders, the selection set, and an index specifying the used algorithm.

Computing population asymptotic variances. The function for computing population asymptotic variances of the MLEs *diff* (Section 2.2; Ünlü and Sargin 2010a) is:

```
pop_variance(pop_matrix, imp, error_pop, v)
```

Subject to the selected version to be performed in population quantities, `v = 1` (minimized corrected) and `v = 2` (corrected), this function computes the population asymptotic variance of the MLE *diff*, which is formulated for the relation and error rate specified in `imp` and `error_pop`, respectively. This population variance, which is a function of the true multinomial probabilities $\rho(R)$, is obtained using the delta method (e.g., see Casella and Berger 2002), which requires calculating the Jacobian matrix of the *diff* coefficient and the inverse of the expected Fisher information matrix for the multinomial distribution.[4] Both matrices, functions of the true multinomial probabilities $\rho(R)$, are implemented analytically in closed

[4]The population asymptotic variance of the MLE *diff* is

$$Var(\mathit{diff}) = \partial\, \mathit{diff}/\partial\, \theta_{|\theta=\theta_t} \cdot \left\{ \left(\frac{1}{n} E_{\theta_t}(-I) \right)^{-1} \cdot \partial\, \mathit{diff}/\partial\, \theta^T_{|\theta=\theta_t} \right\},$$

form. The cell probabilities of that distribution are specified in `pop_matrix`, a matrix of all possible response patterns and their population occurrence probabilities. Note that the arguments `pop_matrix` and `error_pop` can be obtained from a call to the function `pop_iita` (see above), and that the current version of the package **DAKS** does not support computing population asymptotic variances for the original IITA algorithm. This function returns a single value, the population asymptotic variance of the MLE *diff*.

Computing estimated asymptotic variances. The function for computing estimated asymptotic variances of the MLEs *diff* (Section 2.2; Ünlü and Sargin 2010a) is:

```
variance(dataset, imp, v)
```

Subject to the selected version to be performed in sample quantities, `v = 1` (minimized corrected) and `v = 2` (corrected), this function computes a consistent estimator for the population asymptotic variance of the MLE *diff*, which is formulated for the relation and the data specified in `imp` and `dataset`, respectively. This estimated asymptotic variance is obtained using the delta method (cf. `pop_variance`; see above). In the expression for the population asymptotic variance (see Footnote 4), a function of the true probabilities $\rho(R)$, the true parameter vector of the multinomial probabilities is estimated and substituted in the expression by its MLE of the relative frequencies of the response patterns. Note that the two types of estimators for the population asymptotic variances of the *diff* coefficients obtained based on the expected Fisher information matrix and the observed Fisher information matrix yield the same result, in the case of the multinomial distribution. Since computation based on the expected Fisher information matrix is faster, this is implemented in `variance`. Note that the current version of the package **DAKS** does not support computing estimated asymptotic variances for the original IITA algorithm. This function returns the estimated asymptotic variance of the MLE *diff*.

Performing a Z-test. The function for performing a Z-test for the *diff* values is:

```
z_test(dataset, imp, imp_alt = NULL, alternative =
c("two.sided", "less", "greater"), mu = 0, conf.level = 0.95, v)
```

For a given `dataset` a one or two sample Z-test for the *diff* values can be performed. The quasi order is specified by `imp` in the case of a one sample test, and an optional set of implications representing the alternative quasi order is specified by `imp_alt` in the case of a two sample test. The true value of the mean, or of the difference in means if a two sample test is performed, is given by `mu`. The alternative hypothesis is specified by `alternative`. For a one sample test, `conf.level` gives the level of the confidence interval for the single *diff* value. For a two sample test, `conf.level` is the level of the confidence interval for the difference of the two *diff* values. The function `z_test` returns the Z- and p-values, the values and level of the confidence interval, the *diff* values of the specified quasi orders, the specified alternative hypothesis, and the assumed true value of the mean or difference in means.

where θ_t is the true parameter vector of multinomial probabilities. Note that $(\frac{1}{n}E_{\theta_t}(-I))^{-1} = (\delta_{ij}\theta_{ti} - \theta_{ti}\theta_{tj})_{i,j}$, where $I = \left(\partial^2 \ln L/\partial\,\theta_i\,\partial\,\theta_j\right)_{i,j}$ is the Hessian matrix of the log likelihood function of the multinomial distribution, and δ_{ij} is the Kronecker delta. Here, A^T denotes the transpose of a matrix A.

Auxiliary functions used for implementing the IITA algorithms

Two auxiliary functions used for implementing the IITA algorithms are:

```
ob_counter(dataset)
ind_gen(b)
```

The main function `iita` (see above) calls `ob_counter` for computation of the numbers of observed counterexamples, and `ind_gen` for the inductive generation procedure.

Computation of the numbers of observed counterexamples. The function `ob_counter` computes from a binary `dataset` for any item pair (i, j) the corresponding number b_{ij} of observed counterexamples, that is, the number of observed response patterns contradicting the item pair's interpretation as $j \to i$. These values are crucial in the formulations of the IITA algorithms (see Section 2.2 for details). This function returns a matrix of the numbers of observed counterexamples for all pairs of items.

Inductive generation procedure. The function `ind_gen` can be used to generate inductively from a matrix `b` of the numbers of observed counterexamples for all pairs of items, for instance obtained from a call to the previous function `ob_counter`, a set of quasi orders. The inductive generation of the selection set of competing quasi orders is a prime component of the IITA algorithms (see Section 2.2 for details). This function returns a list of the inductively generated surmise relations.

Utility functions

Two functions for switching between test item and knowledge state representations (cf. Birkhoff's theorem in Section 2.1) are:

```
state2imp(P)
imp2state(imp, items)
```

Transformation from knowledge states to implications. The function `state2imp` transforms a set of knowledge states (ought to be a quasi ordinal knowledge space) `P` to the corresponding set of implications (the surmise relation). Note that for any set of knowledge states the returned binary relation is a surmise relation. The number of items of the domain taken as basis for `P` is determined from the number of columns of the matrix `P`.

Transformation from implications to knowledge states. The function `imp2state` transforms a set of implications (ought to be a surmise relation) `imp` to the corresponding set of knowledge states (the quasi ordinal knowledge space). Note that for any set of implications the returned knowledge structure is a quasi ordinal knowledge space. The number of items of the domain taken as basis for `imp`, the argument `items`, must be specified explicitly; because some of the items may not be comparable with any other.

Computing absolute frequencies of response patterns and knowledge states. A function for computing the absolute frequencies of the occurring response patterns, and optionally, the absolute frequencies of a collection of knowledge states in a dataset (see Section 2.1) is:

```
pattern(dataset, n = 5, P = NULL)
```

Argument `n` refers to response patterns. If `n` is specified, the response patterns with the `n` highest frequencies are returned (along with their frequencies). If `pattern` is called without specifying `n` explicitly, by default `n = 5` is used. If `n` is larger than the number of different response patterns in the `dataset`, `n` is set the number of different response patterns. The optional matrix `P` gives the knowledge states to be used; `pattern` then additionally returns information about how often the knowledge states occur in the `dataset`. The default `P = NULL` corresponds to no knowledge states being specified; `pattern` then only returns information about response patterns (as described previously).[5]

Data and quasi order simulation tool. A data (based on the BLIM; Section 2.1) and quasi order simulation tool is included in the package:

```
simu(items, size, ce, lg, imp = NULL, delta)
```

The number of response patterns to be simulated (the sample size) is specified by `size`, the careless error and lucky guess noise parameters are given by `ce` and `lg`, respectively. The single careless error `ce` and lucky guess `lg` probabilities are assumed to be constant over all items, and the underlying knowledge states are assumed to be equiprobable. (The general form of the BLIM allows for varying careless error and lucky guess rates from item to item and for a general distribution of the knowledge states, which is not identifiable in general, however.) The argument `items` gives the number of items of the domain taken as basis for the quasi order underlying the simulation. A specific underlying quasi order can be passed manually via `imp`, or it can be generated randomly. If a quasi order is specified manually, Birkhoff's theorem (Section 2.1) is used to derive the corresponding quasi ordinal knowledge space. The latter is equipped with the error probabilities `ce` and `lg` and the uniform distribution on the set of knowledge states to give the BLIM that is used for simulating the data. From this corresponding knowledge structure $\mathcal{K}$, a 0/1-pattern $K \in \mathcal{K}$ is drawn randomly, that is, with probability $p(K) = 1/|\mathcal{K}|$. For this drawn pattern, all entries are changed from 1 to 0 or from 0 to 1 with the prespecified careless error and lucky guess probabilities `ce` and `lg`, respectively. This is repeated `size` times to generate a data matrix. Note that this is simulating with a specific BLIM, for which the underlying knowledge states are equiprobable. If `imp = NULL`, the underlying quasi order is generated randomly as follows. All reflexive pairs are added to the relation. The constant `delta` is utilized as the probability for adding each of the remaining non-reflexive item pairs to the relation. The transitive closure of this relation is computed, and the resulting quasi order then is the surmise relation underlying the simulation.

This simulation tool returns the simulated binary dataset and the surmise relation and its corresponding quasi ordinal knowledge space used for simulating the data. The probability specified by `delta` does not necessarily correspond to the portion of implications added to

[5] Although throughout this paper all discussion is centered around dichotomously scored items, we want to mention that the function `pattern` even works with polytomous items, but such main functions as `iita` of the package do not.

the randomly generated quasi order, because the transitive closure is formed. In Sargin and Ünlü (2009), a normal sampling scheme for drawing `delta` values is proposed. This sampling scheme provides far better representative samples of quasi orders than simply drawing `delta` values uniformly from the unit interval. (Surmise relations or knowledge structures, and the representativeness of samples of these, are very important in simulation studies investigating IITA type data analysis methods. The IITA algorithms are sensitive to the underlying surmise relation that is used, and to test their performances objectively, a representative sample of the collection of all quasi orders is needed.)

Plotting the Hasse diagram of a surmise relation. Another basic function of the package **DAKS** is a Hasse diagram drawing device:[6]

```
hasse(imp, items)
```

This function plots the Hasse diagram of a surmise relation `imp` (more precisely, of the corresponding quotient set) using the package **Rgraphviz** (Gentry, Long, Gentleman, Falcon, Hahne, Sarkar, and Hansen 2010) from Bioconductor (`http://www.bioconductor.org/`), an interface between R and Graphviz (Graph Visualization Software, `http://graphviz.org/`). Users must install Graphviz on their computers to plot such a diagram. The argument `items` gives the number of items of the domain taken as basis for `imp`. The function `hasse` cannot plot equally informative items. Two items i and j are called equally informative if and only if $j \to i$ and $i \to j$. Only one, the one with the smallest index, of the equally informative items is drawn, and the equally informative items are returned (as tuples) in a list. The plotted Hasse diagram uses as item labels iL, a transformation that takes place internally in the packages **sets** or **relations**.

Table 1 summarizes the functions of the package **DAKS** (`print` and `summary` methods are not listed).

The interdependencies among the functions of the package are as follows. The function `iita` calls the functions `ob_counter` and `ind_gen`, and depending on the value specified for `v`, one of the three IITA functions `orig_iita`, `corr_iita`, and `mini_iita`. Moreover, each of the three functions `orig_iita`, `corr_iita`, and `mini_iita` calls the function `ob_counter`. If the argument `dataset` is specified explicitly, the function `pop_iita` calls the functions `ob_counter` and `ind_gen`. The function `pattern` is called by the function `variance`. The function `z_test` calls the function `variance`, and depending on the value specified for `v`, the function `corr_iita` or the function `mini_iita`.

[6] The Hasse diagram of a partially ordered set $(P, \mathcal{P})$ is defined as the relation consisting of all pairs $p_1 \mathcal{P} p_2$ such that p_1 is covered by p_2, that is, $p_1 \neq p_2$ and there is no $p \in P$, $p \neq p_1$ and $p \neq p_2$, such that $p_1 \mathcal{P} p$ and $p \mathcal{P} p_2$. When P is finite, the Hasse diagram of $(P, \mathcal{P})$ provides an efficient summary of $\mathcal{P}$, in the sense that the Hasse diagram of $(P, \mathcal{P})$ is the smallest relation whose (reflexo-)transitive closure is equal to $\mathcal{P}$. When P is a small set, the Hasse diagram of $\mathcal{P}$ can be conveniently displayed by a graph drawn according to the following conventions: the elements of P are represented by points on a page, with an ascending edge from $p_1 \in P$ to $p_2 \in P$ if p_1 is covered by p_2 (e.g., Doignon and Falmagne 1999, pp. 14–15). Hasse diagrams are named after Helmut Hasse (1898–1979), a German mathematician. When being told that such a type of mathematical diagram was named after him, Helmut Hasse himself did not like that, something so "trivial" being attributed to him.

Function	Short description
`corr_iita`	Computing *diff* values for the corrected IITA algorithm
`hasse`	Plotting a Hasse diagram
`iita`	Computing sample *diff* values and the best fitting quasi order for one of the three IITA algorithms selectively
`imp2state`	Transforming from implications to knowledge states
`ind_gen`	Inductively generating a selection set
`mini_iita`	Computing *diff* values for the minimized corrected IITA algorithm
`ob_counter`	Computing numbers of observed counterexamples
`orig_iita`	Computing *diff* values for the original IITA algorithm
`pattern`	Computing frequencies of response patterns and knowledge states
`pop_iita`	Computing population *diff* values and the selection set for one of the three IITA algorithms selectively
`pop_variance`	Computing population asymptotic variances
`simu`	Data and quasi order simulation tool
`state2imp`	Transforming from knowledge states to implications
`variance`	Computing estimated asymptotic variances
`z_test`	Performing one and two sample Z-tests for *diff* values

Table 1: Summary of the **DAKS** functions.

4. Demonstrating the package DAKS

4.1. An example with real data

We illustrate usage of the package **DAKS** with part of the 2003 Programme for International Student Assessment (PISA; `http://www.pisa.oecd.org/`) data.[7]

The dataset

The dataset consists of the item responses by 340 German students on a 5-item dichotomously scored mathematical literacy test. This is the `pisa` dataset accompanying the package **DAKS**. This dataset resulted from dichotomizing the original multiple-choice or open format test data. The scores are 1 or 0 for a correct or incorrect response, respectively; there are no missing values in the data. Wordings of the test items used in the assessment are not known (not publicly available).

The first six response patterns of the dataset and the five response patterns with largest absolute frequencies in the data:

```
R> head(pisa)

  a b c d e
1 1 0 0 0 0
```

[7]Real applications of KST in a wide range of fields are systematically presented in Albert and Lukas (1999). For a non-technical review of KST including examples as well, see Falmagne *et al.* (1990). A comprehensive bibliography on KST including many references on real applications of KST can be retrieved from `http://wundt.kfunigraz.ac.at/kst.php`.

```
2 0 0 0 0 0
3 1 0 0 0 0
4 1 0 0 0 0
5 0 1 0 0 0
6 1 1 0 0 0
```

```
R> pat <- pattern(pisa)
R> pat
```

```
5 largest response patterns in the data:
11100 11000 10000 11110 00000
   67    61    41    40    20
```

```
R> sum(pat$response.patterns)
```

```
[1] 229
```

We see that the five most frequent response patterns make up for 229 out of the 340 patterns. These are the Guttman patterns of the chain (Footnote 3) $d \to c \to b \to a$ that can likely be assumed to underlie the data. This is also indicated by the following code:

```
R> apply(pisa, 2, table)
```

```
    a   b   c   d   e
0  51  91 167 261 293
1 289 249 173  79  47
```

From items a to e, the sample item popularities (proportions-correct) are well-differentiated and strictly decreasing. For instance, item a is most popular (most frequently solved), item e is least popular (least frequently solved). Since we do not know whether the underlying quasi order may or may not be a chain, we next perform IITA analyses of the PISA data.

IITA analyses of the PISA data

IITA algorithms. We start with running the three IITA algorithms on these data. The results are assigned to variables for later analyses.

```
R> mini <- iita(pisa, v = 1)
R> corr <- iita(pisa, v = 2)
R> orig <- iita(pisa, v = 3)
R> summary(mini)
```

```
          Inductive Item Tree Analysis

Algorithm: minimized corrected IITA
diff values: 143.533 137.399 132.133 115.377 120.168 110.487 82.542 38.976 27.566 107.39 2
```

```
quasi order: {(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (2L, 3L), (2L, 4L),
 (2L, 5L), (3L, 4L), (3L, 5L)}
error rate: 0.116
index in the selection set: 9
```

```
R> summary(corr)
```

```
          Inductive Item Tree Analysis

Algorithm: corrected IITA
diff values: 143.533 137.408 132.214 115.385 121.099 113.423 86.066 40.938 33.319 179.645
quasi order: {(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (2L, 3L), (2L, 4L),
 (2L, 5L), (3L, 4L), (3L, 5L)}
error rate: 0.136
index in the selection set: 9
```

```
R> summary(orig)
```

```
          Inductive Item Tree Analysis

Algorithm: original IITA
diff values: 82.156 80.6 78.63 74.668 134.345 122.664 150.889 141.516 133.269 384.97 822.4
quasi order: {(1L, 4L), (1L, 5L), (2L, 4L), (2L, 5L)}
error rate: 0.081
index in the selection set: 4
```

Inductively generated selection set. We additionally present the inductively generated selection set of competing quasi orders, because that helps investigating the results obtained from applying the IITA algorithms. (Note that this is practicable when the selection set or the number of items are not too large.) For this purpose, the numbers of observed counterexamples for all pairs of items are computed using the function `ob_counter`, and the function `ind_gen` is applied to inductively generate from the returned matrix of the numbers of observed counterexamples a set of quasi orders. The function `ind_gen` returns a list of the inductively generated surmise relations.

```
R> sel_set <- ind_gen(ob_counter(pisa))
R> sel_set
```

```
[[1]]
{(1L, 5L)}

[[2]]
{(1L, 4L), (1L, 5L)}

[[3]]
{(1L, 4L), (1L, 5L), (2L, 5L)}
```

```
[[4]]
{(1L, 4L), (1L, 5L), (2L, 4L), (2L, 5L)}

[[5]]
{(1L, 4L), (1L, 5L), (2L, 4L), (2L, 5L), (3L, 5L)}

[[6]]
{(1L, 3L), (1L, 4L), (1L, 5L), (2L, 4L), (2L, 5L), (3L, 5L)}

[[7]]
{(1L, 3L), (1L, 4L), (1L, 5L), (2L, 4L), (2L, 5L), (3L, 4L), (3L, 5L)}

[[8]]
{(1L, 3L), (1L, 4L), (1L, 5L), (2L, 3L), (2L, 4L), (2L, 5L), (3L, 4L),
 (3L, 5L)}

[[9]]
{(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (2L, 3L), (2L, 4L), (2L, 5L), (3L, 4L),
 (3L, 5L)}

[[10]]
{(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (2L, 3L), (2L, 4L), (2L, 5L), (3L, 4L),
 (3L, 5L), (4L, 5L)}

[[11]]
{(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (2L, 1L), (2L, 3L), (2L, 4L), (2L, 5L),
 (3L, 4L), (3L, 5L), (4L, 5L), (5L, 4L)}

[[12]]
{(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (2L, 1L), (2L, 3L), (2L, 4L), (2L, 5L),
 (3L, 4L), (3L, 5L), (4L, 3L), (4L, 5L), (5L, 3L), (5L, 4L)}

[[13]]
{(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (2L, 1L), (2L, 3L), (2L, 4L), (2L, 5L),
 (3L, 1L), (3L, 2L), (3L, 4L), (3L, 5L), (4L, 1L), (4L, 2L), (4L, 3L), (4L, 5L),
 (5L, 1L), (5L, 2L), (5L, 3L), (5L, 4L)}
```

The quasi order with tenth index in the selection set is a chain (Footnote 3), that is, the items form a Guttman scale. The neighboring quasi orders with indices eight, nine, and eleven are very close to a chain. Therefore we expect the underlying quasi order to be one of these four, most likely. Note that inspecting the selection set for specific quasi orders can be useful in general, because the selection set only contains a small fraction of all possible quasi orders.

General remarks. The corrected and minimized corrected IITA algorithms yield the same solution quasi order, which is close to a chain (cf. Figure 1). The original IITA algorithm

selects a quasi order which is clearly different from that returned by the other two algorithms, and which is far from being a chain (cf. Footnote 8). This is also reflected by the corresponding *diff* values. They are similar for the corrected and minimized corrected IITA algorithms, and considerably smaller than the *diff* value obtained for the original algorithm. There is evidence that the original IITA algorithm fails in revealing underlying "close-to-chain" quasi orders. Furthermore, fitting the classical Rasch model to this dataset corroborates the chain hierarchy among the five mathematical literacy test items. Since the Rasch model assumes unidimensionality of the latent trait, the items can be ordered linearly along the continuum in terms of their difficulties (with respect to the natural ordering in the reals), resulting in a deterministic Guttman scale; in this regard, see also Ünlü (2007). Due to the highly confirmatory fit statistics obtained for this dataset, the items most likely form a chain. For details on comparing the different data analysis methods and psychometric approaches, see Sargin and Ünlü (2009) and Ünlü and Sargin (2010a). See also the "General remarks" in "Inductive item tree analysis algorithms in sample values" of Section 2.2. The present paper rather is on introducing the R package **DAKS**.

Comparing and plotting the solution quasi orders obtained from IITA analyses

*Comparing using functions of the package **sets**.* One can use functions of the package **sets**, for example when comparing the solution quasi orders obtained from different IITA algorithms. The next two functions that we describe are from the package **sets**. (Of course, other functions of the package **sets** can be helpful and used as well.)

The symmetric set difference between the solutions of the original and minimized corrected IITA algorithms can be computed by:

```
R> set_symdiff(orig$implications, mini$implications)

{(1L, 2L), (1L, 3L), (2L, 3L), (3L, 4L), (3L, 5L)}
```

The symmetric set difference gives the implications in which the two relations differ.

In the example here we see that all implications of the original IITA algorithm solution are contained in the quasi order derived using the minimized corrected IITA algorithm:

```
R> set_is_proper_subset(orig$implications, mini$implications)

[1] TRUE
```

Plotting the Hasse diagram. Graphics are convenient to use and they can present information effectively. The graphic that is used throughout KST is the Hasse diagram (see Footnote 6). It is utilized for presenting information, not for exploring data. For approaches to graphically exploring KST data based on mosaic plots for instance, see Ünlü and Sargin (2009). A Hasse diagram can be plotted by:

```
R> hasse(mini$implications, 5)

list()
```

This gives the Hasse diagram of the solution quasi order of the minimized corrected algorithm shown in Figure 1. From Figure 1 we see that, for example, item 3 implies items 1 and 2, and that item 3 is implied by items 4 and 5. Note that the returned list of equally informative items is empty; therefore the diagram faithfully represents the quasi order. The plotted Hasse diagram uses as item labels iL (cf. Section 3.2).[8]

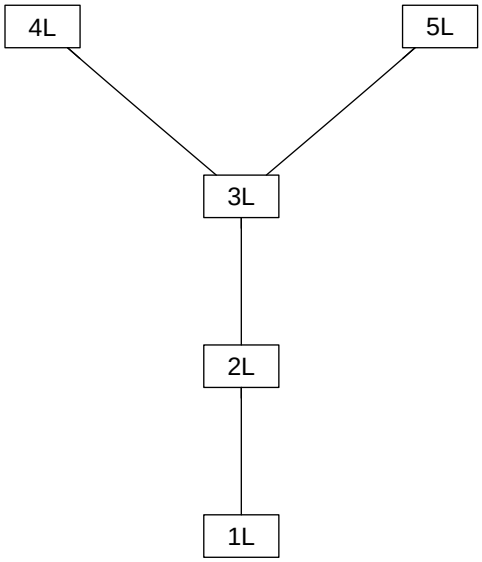

Figure 1: Hasse diagram of the quasi order obtained for the PISA dataset under the minimized corrected IITA algorithm.

Comparing using the Z-test. We perform a Z-test to check whether the quasi order obtained by the corrected and minimized corrected IITA algorithms has a smaller *diff* value than the quasi order forming a chain. If this is not the case, we cannot say with certainty whether the derived quasi order is significantly better than the chain hierarchy.

```
R> z_test(pisa, sel_set[[10]], sel_set[[9]], alternative = "less", v = 1)

          Two sample Z-test

z =  2.2666  p-value =  0.0117
alternative hypothesis: true mean is less 0
95 percent confidence interval:
```

[8]Note that the solution quasi order obtained for the PISA dataset under the original IITA algorithm is given by `{(1L, 4L), (1L, 5L), (2L, 4L), (2L, 5L)}`. If we denote this quasi order on $Q = \{a, b, c, d, e\}$ by $\sqsubseteq$, then $a \sqsubseteq d$, $a \sqsubseteq e$, $b \sqsubseteq d$, and $b \sqsubseteq e$. In particular, item c is $\sqsubseteq$-incomparable with any of the other items a, b, d, e, items a and b are $\sqsubseteq$-incomparable, and items d and e are $\sqsubseteq$-incomparable. Here we call $q_1, q_2 \in Q$ $\sqsubseteq$-incomparable if and only if $q_1 \not\sqsubseteq q_2$ and $q_2 \not\sqsubseteq q_1$.

```
  0.0001894101 Inf
sample estimates:
    mean in imp mean in imp_alt
        0.00093         0.00024
```

The p-value is 0.0117, hence it can be assumed that the *diff* value of the derived quasi order is significantly smaller than the *diff* value of the chain. According to the *diff* criterion, therefore the obtained quasi order has a distinctly better fit to the data.

Through previous analyses we have gained information about the dependencies between the test items of the PISA dataset. We have seen that, for instance, items 4 and 5 imply all other items. Therefore we can surmise from a student's solving the items 4 or 5 that this student will also be able to solve items 1, 2, and 3. Such information can be used in computerized adaptive testing, in order to reduce the number of items administered to the student.

4.2. An example with simulated data

To illustrate the other functions of the package **DAKS**, we start with simulating a quasi order and a dataset. Note that every simulation is individual, in the sense that different results are obtained from simulation to simulation.[9]

Data and quasi order simulation

Since `imp = NULL`, a quasi order is generated randomly using a probability of `delta = 0.15` for adding an implication to the relation. Based on this underlying surmise relation, a binary dataset consisting of 9 items and 1500 examinees is simulated using a same careless error and lucky guess rate of 0.1 over all items. The simulated binary dataset and the simulated surmise relation and its corresponding quasi ordinal knowledge space are returned.

```
R> ex_data <- simu(9, 1500, 0.1, 0.1, delta = 0.15)
```

The randomly generated quasi order underlying the simulated data is:

```
R> ex_data$implications
```

```
{(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (1L, 6L), (1L, 7L), (1L, 8L), (1L, 9L),
 (2L, 1L),
 (2L, 3L), (2L, 4L), (2L, 5L), (2L, 6L), (2L, 7L), (2L, 8L), (2L, 9L), (3L, 1L),
 (3L, 2L),
 (3L, 4L), (3L, 5L), (3L, 6L), (3L, 7L), (3L, 8L), (3L, 9L), (4L, 1L), (4L, 2L),
 (4L, 3L),
 (4L, 5L), (4L, 6L), (4L, 7L), (4L, 8L), (4L, 9L), (5L, 1L), (5L, 2L), (5L, 3L),
 (5L, 4L),
```

[9]The following simulation is meant for demonstrating the functions of the package **DAKS**. Extensive simulation studies based on or investigating the BLIM in KST are presented, for instance, in Sargin and Ünlü (2009), Schrepp (2003, 2005), Stefanutti and Robusto (2009), Ünlü (2006), and Ünlü and Sargin (2010a). For example, Stefanutti and Robusto (2009), among other things, assess goodness-of-fit of the BLIM to simulated data via Pearson's X^2. The comprehensive bibliography on KST at http://wundt.kfunigraz.ac.at/kst.php includes many more references on theoretical and simulation studies in KST.

```
 (5L, 6L), (5L, 7L), (5L, 8L), (5L, 9L), (6L, 1L), (6L, 2L), (6L, 3L), (6L, 4L),
 (6L, 5L),
 (6L, 7L), (6L, 8L), (6L, 9L), (7L, 1L), (7L, 2L), (7L, 3L), (7L, 4L), (7L, 5L),
 (7L, 6L),
 (7L, 8L), (7L, 9L), (9L, 1L), (9L, 2L), (9L, 3L), (9L, 4L), (9L, 5L), (9L, 6L),
 (9L, 7L), (9L, 8L)}
```

Corrected IITA analyses of the simulated data

In the following, analyses are performed under the corrected IITA algorithm only; under the other two algorithms the analyses are analogous. We run the corrected IITA procedure on the simulated dataset.

```
R> ex_corr <- iita(ex_data$dataset, v = 2)
R> ex_corr
```

```
          Inductive Item Tree Analysis

Algorithm: corrected IITA

quasi order: {(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (1L, 6L), (1L, 7L), (1L, 8L),
 (1L, 9L), (2L, 1L),
 (2L, 3L), (2L, 4L), (2L, 5L), (2L, 6L), (2L, 7L), (2L, 8L), (2L, 9L), (3L, 1L),
 (3L, 2L),
 (3L, 4L), (3L, 5L), (3L, 6L), (3L, 7L), (3L, 8L), (3L, 9L), (4L, 1L), (4L, 2L),
 (4L, 3L),
 (4L, 5L), (4L, 6L), (4L, 7L), (4L, 8L), (4L, 9L), (5L, 1L), (5L, 2L), (5L, 3L),
 (5L, 4L),
 (5L, 6L), (5L, 7L), (5L, 8L), (5L, 9L), (6L, 1L), (6L, 2L), (6L, 3L), (6L, 4L),
 (6L, 5L),
 (6L, 7L), (6L, 8L), (6L, 9L), (7L, 1L), (7L, 2L), (7L, 3L), (7L, 4L), (7L, 5L),
 (7L, 6L),
 (7L, 8L), (7L, 9L), (9L, 1L), (9L, 2L), (9L, 3L), (9L, 4L), (9L, 5L), (9L, 6L),
 (9L, 7L), (9L, 8L)}
```

The quasi order obtained by data analysis is the true quasi order underlying the data. (This of course may not always be the case.)

```
R> ex_corr$implications == ex_data$implications
```

```
[1] TRUE
```

Corrected IITA analyses in the population

Next we discuss the functions which provide the basis for statistical inference methodology.

Corrected IITA algorithm. The corrected IITA algorithm can be performed in population quantities, yielding information about the population *diff* values, population occurrence probabilities of response patterns, population error rates, and the inductively generated selection set:

```
R> pop <- pop_iita(ex_data$implications, 0.1, 0.1, 9, dataset = ex_data$dataset,
+    v = 2)
attributes(pop)

$names
[1] "pop.diff"      "pop.matrix"    "error.pop"     "selection.set" "v"

$class
[1] "popiita"
```

For the argument `imp` we use the simulated surmise relation `ex_data$implications`, with 9 items of the domain for this quasi order. The knowledge structure corresponding to `ex_data$implications` is equipped with a same careless error and lucky guess rate of 0.1 over all items. Since `dataset = ex_data$dataset` (and `A = NULL`), the simulated binary data are used to generate the set of competing quasi orders based on the sample version of the inductive generation procedure (cf. Section 3.2).

Sample and population diff values. To compare sample with population *diff* values, the sample *diff* coefficient is transformed to become the MLE for the corresponding population *diff* coefficient (see Section 2.2 for the definition of *diff*):

```
R> round(ex_corr$diff / 1500^2, 4)

 [1] 0.0160 0.0159 0.0156 0.0155 0.0153 0.0.0152 0.0151 0.0145 0.0133 0.0133 0.0127
[12] 0.0116 0.0094 0.0083 0.0079 0.0068 0.0053 0.0037 0.0016 0.0011 0.0011 0.0000
[23] 0.0000 0.0058

R> round(pop$pop.diff, 4)

 [1] 0.0167 0.0166 0.0163 0.0162 0.0160 0.0159 0.0157 0.0152 0.0141 0.0141 0.0135
[12] 0.0124 0.0101 0.0090 0.0085 0.0073 0.0056 0.0040 0.0017 0.0012 0.0012 0.0000
[23] 0.0000 0.0056
```

The respective sample and population values are quite similar, already for a sample size of 1500. This is obvious given the fact that the sample *diff* values converge in probability (and expectation) to the population *diff* values (see Section 2.2).

The quasi order with minimum population *diff* value can be queried:

```
R> mp <- which.min(pop$pop.diff)
R> pop$selection.set[[mp]]
```

```
{(1L, 2L), (1L, 3L), (1L, 4L), (1L, 5L), (1L, 6L), (1L, 7L), (1L, 8L),
 (1L, 9L), (2L, 1L), (2L, 3L), (2L, 4L), (2L, 5L), (2L, 6L), (2L, 7L), (2L, 8L),
 (2L, 9L), (3L, 1L), (3L, 2L), (3L, 4L), (3L, 5L), (3L, 6L), (3L, 7L),
 (3L, 8L), (3L, 9L), (4L, 1L), (4L, 2L), (4L, 3L), (4L, 5L), (4L, 6L), (4L, 7L),
 (4L, 8L), (4L, 9L), (5L, 1L), (5L, 2L), (5L, 3L), (5L, 4L), (5L, 6L),
 (5L, 7L), (5L, 8L), (5L, 9L), (6L, 1L), (6L, 2L), (6L, 3L), (6L, 4L), (6L, 5L),
 (6L, 7L), (6L, 8L), (6L, 9L), (7L, 1L), (7L, 2L), (7L, 3L), (7L, 4L),
 (7L, 5L), (7L, 6L), (7L, 8L), (7L, 9L), (9L, 1L), (9L, 2L), (9L, 3L), (9L, 4L),
 (9L, 5L), (9L, 6L), (9L, 7L), (9L, 8L)}
```

This quasi order is the true quasi order underlying the simulated dataset. Of course this may not always be the case, especially for smaller sample sizes or higher response error rates.

The population analogs are useful for comparing the IITA algorithms (Ünlü and Sargin 2010a). In Ünlü and Sargin (2010a), a thorough simulation study is performed. It is shown that the original IITA algorithm leads to bad results in population (and sample) values. In this regard, see also the "General remarks" in "Inductive item tree analysis algorithms in sample values" of Section 2.2. Hence the corrected and minimized corrected IITA algorithms are recommended for use in real applications.

Estimated and population asymptotic variances. As mentioned in Section 2.2, the MLEs *diff* are asymptotically normal. Large sample normality with associated standard errors can be used to construct confidence intervals for the population values of and to test hypotheses about the *diff* coefficients (cf. Sections 3.2 and 4.1). For instance, using the function `z_test` we can test whether one of two quasi orders has a significantly smaller *diff* value in the population. The quasi orders could, for example, be derived from querying experts. In order to do such a test, the asymptotic variances need to be estimated. Population asymptotic variances and consistent estimators thereof can be computed using the delta method (cf. Section 3.2).

The estimated asymptotic variance of the MLE *diff* in the sample version corrected IITA algorithm can be computed by:

```
R> var_sample <- variance(ex_data$dataset, ex_data$implications, v = 2)
R> var_sample

[1] 8.944665e-05

R> sqrt(var_sample)

[1] 0.009457624
```

This estimated asymptotic variance is formulated for the simulated data `ex_data$dataset` and the randomly generated surmise relation `ex_data$implications` underlying these data.

The corresponding population asymptotic variance of the MLE *diff* in the population version corrected IITA algorithm is:

```
R> pop_variance <- pop_variance(pop$pop.matrix, pop$selection.set[[mp]],
+    pop$error.pop[mp], v = 2)
R> pop_variance
```

```
[1] 4.453308e-07
```

```
R> sqrt(pop_variance)
```

```
[1] 0.0006673311
```

This population asymptotic variance is formulated for the population cell probabilities `pop$pop.matrix` of the multinomial distribution and the population error rate obtained for the true quasi order (with minimum population *diff* value) underlying the simulated data. Note that in this example the arguments `pop_matrix` and `error_pop` are obtained from a call to the function `pop_iita`. For the argument `imp` we use the true quasi order.

The sample and population values are quite similar. The sample variance is a consistent estimator for the population variance (convergence in probability). This and the function `variance` are important, because in real applications the estimated asymptotic variance has to be used (e.g., for calculating standard deviations of the *diff* measures or for performing such significance tests as the `z_test`).

5. Conclusion

Summary. This paper has introduced the R package **DAKS**. This package contains several basic functions for KST, and it primarily implements the IITA methods for data analysis in KST, at the level of both sample and population values. Functions for computing various population values and for estimating asymptotic variances and performing Z-tests are also contained. These tools provide the basis for statistical inference methodology and for further analyses in KST. We have described the functions of the package **DAKS** and demonstrated their usage by real and simulated data examples.

Some suggestions for future implementations of the package. In future research, we plan to implement other fit measures such as the *di* (discrepancy) index (Kambouri *et al.* 1994) or the *CA* (correlational agreement) coefficient (Van Leeuwe 1974). Functions for computing confidence intervals and for performing hypothesis tests for the *diff* and other fit measures will also be implemented. The present functions of the package are to be extended; for example, the function `hasse` should incorporate drawing diagrams for knowledge structures, or the simulation tool could allow for individual response error probabilities for each item.

General remarks about the interplay between KST and IRT. By contributing the R package **DAKS** we hope to have established a basis for computational work in the so far combinatorial theory of knowledge spaces. Implementing KST procedures in R can help to bring together KST and IRT. A number of R packages are available for IRT; for instance, **ltm** (Rizopoulos 2006) or **mokken** (Van der Ark 2007). KST and IRT are split directions of psychological test theories and have recently been partly compared at a theoretical level (Stefanutti 2006; Stefanutti and Robusto 2009; Ünlü 2006, 2007). Using R as an interface between these theories may prove valuable in comparing them at a computational level.

Why should one be interested in trying to unify KST and IRT? What can KST contribute to IRT, and vice versa? The following lists some arguments supporting the importance of a possible fusion of KST and IRT (cf. Ünlü 2007).

Statistical inference methodologies. An IRT-type modeling in KST could provide feasible new statistical inference methodologies. For IRT, unlike KST, has plenty of sophisticated statistical methods that could be suited to and applied in KST (e.g., Skrondal and Rabe-Hesketh 2004).

Restrictivity. IRT models that simultaneously imply person and item orderings are restrictive models with respect to real data (e.g., Sijtsma and Molenaar 2002). In general, they will not fit many empirical datasets. A unified test theory combining KST and IRT could positively contribute to and improve on this observation. For a strength of KST is that it implies very general combinatorial structures, both at the levels of persons and items, contrary to IRT, implying more restrictive linear orderings. KST further provides mathematical results on the linkage between these levels, offering flexibility in the choice of a representation. A unified approach could deliver as general as possible probabilistic models that could imply both a person ordering and an item ordering, extend linear orderings to more general surmise relations or even surmise systems, allow for flexibility in representation, encompass most of the existing IRT and KST models as special cases, and thus fit far more datasets in practice.

Adaptive testing. A unified test theory could also positively contribute to the problem of adaptive testing in nonparametric IRT using ordinal measurement information (e.g., Huisman and Molenaar 2001). Adaptive testing, however, is a major strength of KST.

Qualitative derivation of hierarchies among items. KST offers a number of 'a priori' qualitative, psychological theory driven methods for the derivation of hierarchies among items (e.g., Albert and Lukas 1999). In IRT, however, orderings of items are obtained 'a posteriori' by using quantitative, statistical methods (e.g., by estimating the difficulty parameter of each item). A unified framework could provide qualitative, theory driven, or quantitative, statistical, or hybrid derivation methods.

The R environment is ideally suited for a unified test theory combining KST and IRT. Such a comprehensive environment can encompass many existing KST and IRT models and software, unified under one umbrella. This implies easy access to and use of software for the practical application of the unified test theory, KST, and IRT models to empirical data.

Acknowledgments

We thank the two anonymous reviewers for their critical and valuable comments that helped to improve the manuscript greatly.

References

Airasian PW, Bart WM (1973). "Ordering Theory: A New and Useful Measurement Model." *Educational Technology*, **13**, 56–60.

Albert D, Lukas J (eds.) (1999). *Knowledge Spaces: Theories, Empirical Research, and Applications*. Lawrence Erlbaum Associates, Mahwah.

Bart WM, Krus DJ (1973). "An Ordering-theoretic Method to Determine Hierarchies Among Items." *Educational and Psychological Measurement*, **33**, 291–300.

Birkhoff G (1937). "Rings of Sets." *Duke Mathematical Journal*, **3**, 443–454.

Casella G, Berger RL (2002). *Statistical Inference*. Duxbury, Pacific Grove, CA, 2nd edition.

Dayton CM, Macready GB (1976). "A Probabilistic Model for the Validation of Behavioral Hierarchies." *Psychometrika*, **41**, 189–204.

Doignon JP, Falmagne JC (1985). "Spaces for the Assessment of Knowledge." *International Journal of Man-Machine Studies*, **23**, 175–196.

Doignon JP, Falmagne JC (1987). "Knowledge Assessment: A Set Theoretical Framework." In "Beiträge zur Begriffsanalyse: Vorträge der Arbeitstagung Begriffsanalyse, Darmstadt, 1986," pp. 129–140. B.I. Wissenschaftsverlag, Mannheim, Germany.

Doignon JP, Falmagne JC (1999). *Knowledge Spaces*. Springer-Verlag, Berlin.

Falmagne JC (1989a). "A Latent Trait Theory via a Stochastic Learning Theory for a Knowledge Space." *Psychometrika*, **54**, 283–303.

Falmagne JC (1989b). "Probabilistic Knowledge Spaces: A Review." In F Roberts (ed.), "Applications of Combinatorics and Graph Theory to the Biological and Social Sciences," volume 17, pp. 283–303. Springer-Verlag, New York.

Falmagne JC, Koppen M, Villano M, Doignon JP, Johannesen L (1990). "Introduction to Knowledge Spaces: How to Build, Test and Search Them." *Psychological Review*, **97**, 201–224.

Gentry J, Long L, Gentleman R, Falcon S, Hahne F, Sarkar D, Hansen K (2010). ***Rgraphviz**: How To Plot A Graph Using **Rgraphviz***. R package version 1.26.0, URL http://www.bioconductor.org/packages/release/bioc/html/Rgraphviz.html.

Goodman LA, Kruskal WH (1979). *Measures of Association for Cross Classifications*. Springer-Verlag, New York.

Guttman L (1944). "A Basis for Scaling Qualitative Data." *American Sociological Review*, **9**, 139–150.

Hornik K, Meyer D (2010). ***relations**: Data Structures and Algorithms for Relations*. R package version 0.5-8, URL http://CRAN.R-project.org/package=relations.

Huisman M, Molenaar IW (2001). "Imputation of Missing Scale Data with Item Response Models." In A Boomsma, MAJ van Duijn, TAB Snijders (eds.), "Essays on Item Response Theory," pp. 221–244. Springer, New York.

Kambouri M (1991). *Knowledge Assessment: A Comparison of Human Experts and Computerized Procedures*. New York University, New York. Ph.D. dissertation.

Kambouri M, Koppen M, Villano M, Falmagne JC (1994). "Knowledge Assessment: Tapping Human Expertise by the QUERY Routine." *International Journal of Human-Computer Studies*, **40**, 119–151.

Lazarsfeld PF, Henry NW (1968). *Latent Structure Analysis*. Houghton Mifflin, Boston.

Meyer D, Hornik K (2009). "Generalized and Customizable Sets in R." *Journal of Statistical Software*, **31**(2), 1–27. URL http://www.jstatsoft.org/v31/i02/.

Mokken RJ (1971). *A Theory and Procedure of Scale Analysis*. De Gruyter, The Hague, Mouton/Berlin.

Proctor CH (1970). "A Probabilistic Formulation and Statistical Analysis of Guttman Scaling." *Psychometrika*, **35**, 73–78.

R Development Core Team (2010). *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/.

Rasch G (1960). *Probabilistic Models for some Intelligence and Attainment Tests*. Nielsen & Lydiche, Copenhagen. Expanded Edition, University of Chicago Press, Chicago, 1980.

Rizopoulos D (2006). "**ltm**: An R Package for Latent Variable Modeling and Item Response Theory Analyses." *Journal of Statistical Software*, **17**(5), 1–25. URL http://www.jstatsoft.org/v17/i05/.

Sargin A, Ünlü A (2009). "Inductive Item Tree Analysis: Corrections, Improvements, and Comparisons." *Mathematical Social Sciences*, **58**, 376–392.

Schrepp M (2003). "A Method for the Analysis of Hierarchical Dependencies Between Items of a Questionnaire." *Methods of Psychological Research Online*, **19**, 43–79.

Schrepp M (2005). "About the Connection Between Knowledge Structures and Latent Class Models." *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, **1**, 93–103.

Schrepp M (2006). "**ITA 2.0**: A Program for Classical and Inductive Item Tree Analysis." *Journal of Statistical Software*, **16**(10), 1–14. URL http://www.jstatsoft.org/v16/i10/.

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

Skrondal A, Rabe-Hesketh S (2004). *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Chapman & Hall/CRC, Boca Raton, FL.

Stefanutti L (2006). "A Logistic Approach to Knowledge Structures." *Journal of Mathematical Psychology*, **50**, 545–561.

Stefanutti L, Robusto E (2009). "Recovering a Probabilistic Knowledge Structure by Constraining its Parameter Space." *Psychometrika*, **74**, 83–96.

Ünlü A (2006). "Estimation of Careless Error and Lucky Guess Probabilities for Dichotomous Test Items: A Psychometric Application of a Biometric Latent Class Model with Random Effects." *Journal of Mathematical Psychology*, **50**, 309–328.

Ünlü A (2007). "Nonparametric Item Response Theory Axioms and Properties Under Nonlinearity and their Exemplification with Knowledge Space Theory." *Journal of Mathematical Psychology*, **51**, 383–400.

Ünlü A (2010). "A Note on the Connection Between Knowledge Structures and Latent Class Models." *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*. Manuscript in press.

Ünlü A, Sargin A (2009). "Mosaics for Visualizing Knowledge Structures." *Manuscript under revision*.

Ünlü A, Sargin A (2010a). "Maximum Likelihood Methodology for *diff* Fit Measures for Quasi Orders." *Manuscript submitted for publication*.

Ünlü A, Sargin A (2010b). "**DAKS**: An R Package for Data Analysis Methods in Knowledge Space Theory." *Journal of Statistical Software*, **37**(2), 1–31. URL http://www.jstatsoft.org/v37/i02/.

Van der Ark LA (2007). "Mokken Scale Analysis in R." *Journal of Statistical Software*, **20**(11), 1–19. URL http://www.jstatsoft.org/v17/i05/.

Van der Linden WJ, Hambleton RK (eds.) (1997). *Handbook of Modern Item Response Theory*. Springer, New York.

Van Leeuwe JFJ (1974). "Item Tree Analysis." *Nederlands Tijdschrift voor de Psychologie*, **29**, 475–484.

Affiliation:

Ali Ünlü
Statistics in the Social and Educational Sciences
Faculty of Statistics
University of Dortmund
D-44221 Dortmund, Germany
E-mail: uenlue@statistik.tu-dortmund.de
URL: http://www.statistik.tu-dortmund.de/uenlue.html

Anatol Sargin
Statistics in the Social and Educational Sciences
Faculty of Statistics
University of Dortmund
D-44221 Dortmund, Germany
E-mail: sargin@statistik.tu-dortmund.de
URL: http://www.statistik.tu-dortmund.de/sargin1.html