Universität Augsburg

Mathematisch-Naturwissenschaftliche Fakultät

Institut für Mathematik

Lehrstuhl für Rechnerorientierte Statistik und Datenanalyse

Data Analysis Methods in Knowledge Space Theory

Dissertation zur Erlangung des Doktorgrades an der
Mathematisch-Naturwissenschaftlichen Fakultät der
Universität Augsburg

eingereicht von

Anatol Sargin

Augsburg, den 17.11.2009

Gutachter: Prof. Dr. Ali Ünlü Prof. Dr. Dietrich Albert

Mündliche Prüfung: 25. Januar 2010

Prüfer: Prof. Dr. Ali Ünlü,

Prof. Antony Unwin Ph.D.,

Prof. Dr. Friedrich Pukelsheim

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we introduce the topic of knowledge spaces and data analysis methods for deriving knowledge structures. The research reported in this thesis is motivated, the relevant literature is reviewed, and an outline of the present work is given.

1.1 Motivation

Knowledge space theory (KST) provides a framework for the modeling, testing, and training of knowledge. Knowledge can be defined in a very general way. For example, it could be questions taken from a math exam, or attitude towards political issues. Knowledge as a whole can be seen as sum of different pieces of knowledge. A natural assumption is that some pieces of knowledge may imply others. In KST, this results in a knowledge structure representing the organization of knowledge. For instance, a math problem can be a sub-problem of a more complex problem. Note that the theory of knowledge spaces is not restricted to topics of psychology. KST can also be applied to other fields, such as pattern recognition or medical diagnosis (Doignon and Falmagne, 1999).

If all implications would be known and no errors be made, training and testing of knowledge could be done efficiently. An examiner could ask questions based on prior answers of the examinee. The questions could even be administered by a computer, (computer adaptive testing, e.g. (van der Linden and Glas, 2000)). For training students, the implications can be used to develop a learning path, along which students are gradually taught new pieces of knowledge according to their knowledge states.

Since knowledge structures are latent, hence not directly observable, a crucial task in knowledge space theory is to reveal them. There are different ways of building knowledge structures, such as by querying experts, by item construction, or by means of data analysis methods. All methods have advantages and disadvantages. Querying experts, for instance, can lead to ambiguous knowledge structures, and is expensive and time consuming. Data analysis methods are fast and cheap, and most importantly are derived from observed data. Further, techniques such as hypothesis testing can be used for analyzing the results. So far data analysis methods in KST have been treated ad hoc only. In this work, we present new algorithms for deriving knowledge structures by data analysis, and unify them based on approved statistical approaches, such as maximum likelihood methodology.

Due to the computational effort, it is mandatory to use software in real life situations. In the `R` package `DAKS`, all data analysis methods analyzed in this thesis are implemented. This is an important contribution as it is the single software implementing these data analysis methods. Furthermore, it introduces the software `R`, with all of its advantages such as accessibility to statistical methods or being free of charge, to the users of KST.

1.2 Relevant literature

KST was introduced by Doignon and Falmagne (1985). Most of the theory of knowledge spaces is presented in a monograph by Doignon and Falmagne (1999); see also Doignon and Falmagne (1987), Falmagne (1989), and Falmagne et al. (1990). For concrete application examples, see in particular Albert and Lukas (1999). Current references on knowledge spaces can be obtained from `http://wundt.kfunigraz.ac.at/kst.php`.

Item tree analysis (ITA) was introduced by van Leeuwe (1974). The enhancement leading to inductive item tree analysis (IITA) was introduced by Schrepp (1999). This algorithm was analyzed and improved in Schrepp (2002, 2003, 2006, 2007). In Sargin and Ünlü (2009a) the original IITA algorithm was corrected and optimized regarding the used fit measure. Maximum likelihood methodology and statistical concepts, such as asymptotic normality or consistency, are proposed in Ünlü and Sargin (2008a). The use of asymptotic normality, leading to the computation of asymptotic variances, and hence inferential statistics was made in Ünlü and Sargin (2009). For example, this can be used for computing confidence intervals or hypothesis testing. The algorithms and the fundamental concepts of KST are implemented in the `R` package `DAKS` (Sargin and Ünlü, 2008). All simulations and computations in this work were performed in `R` (`R`: Development Core Team, 2009) mainly using the package `DAKS`.

Detecting knowledge states from data visually is discussed in Ünlü and Sargin (2008b). They show that data analysis methods in KST and mosaic plots complement one another, and lead to better results when using both in analyses. A good overview of graphics can be found in Chen et al. (2008). For exploratory data analysis using interactive graphics, see Theus and Urbanek (2008); Unwin et al. (2006).

1.3 Outline

This work deals with data analysis methods in KST. First, in the next section the relevant literature on KST and data analysis methods for deriving knowledge structures is presented. In Chapter 2, the main deterministic and probabilistic concepts of KST are introduced. In Chapter 3, data analysis methods are discussed. Item tree analysis, the predecessor of the three algorithms analyzed in this work, is briefly reviewed. Inductive item tree analysis (IITA) and its two enhancements, corrected and minimized corrected IITA, are thoroughly discussed. The IITA algorithms are compared in two simulation studies and with real datasets. We introduce maximum likelihood methodology for the IITA methods, by interpreting the fit measures of these methods as maximum likelihood estimators. It is shown that these fit measures have several asymptotic quality properties. In Chapter 4, the `R` package `DAKS` is presented, and the use of the package's functions are illustrated with examples. In Chapter 5, a summary is given, and important directions for future research are presented.

Chapter 2

Knowledge space theory

A mathematical framework for the assessment, modeling, and training of knowledge is realized with KST. In this chapter we review the essential deterministic and probabilistic concepts of KST.

2.1 Deterministic concepts

Assume a set Q of dichotomous items, and let n be the number of items. The set Q is called the domain of the knowledge structure. Mastering an item $j \in Q$ may imply mastering another item $i \in Q$. If no response errors are made, these implications, $j \rightarrow i$, 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 $\mathcal{K}$. 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 $j \rightarrow i$ (for $i, j \in Q$) stands for the pair $(i, j) \in \sqsubseteq$, also denoted by $i \sqsubseteq j$. Quasi orders are referred to as surmise relations in KST.

An example helps to illustrate these concepts. Let $Q = \{a, b, c\}$ be a set of three dichotomous items. Consider the surmise relation

$$\sqsubseteq = \{(a, a), (b, b), (c, c), (a, b), (a, c)\}$$

on Q, that is, $a \to a$, $b \to b$, $c \to c$, $b \to a$, and $c \to a$. These implications specify the feasible latent knowledge states. A respondent can master just item a. This does not imply mastery of any other item. In that case, the knowledge state is $\{a\}$. However, if the respondent masters c, for instance, then a must also be mastered. This gives the knowledge state $\{a, c\}$. We see that there are exactly five knowledge states consistent with the surmise relation, and the corresponding knowledge structure is

$$\mathcal{K} = \{\emptyset, \{a\}, \{a, b\}, \{a, c\}, Q\}.$$

Note that this knowledge structure is closed under set-theoretic union and intersection. Such knowledge structures are called quasi ordinal knowledge spaces.

The five knowledge states of the example knowledge structure consistent with the surmise relation are obtained, in fact, applying Birkhoff (1937)'s theorem (see also Doignon and Falmagne, 1999, Theorem 1.49). This theorem provides a linkage between quasi ordinal knowledge spaces and surmise relations on an item set. It states that 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 $\mathcal{Q}$ on Q. More formally:

$$\begin{aligned} p\mathcal{Q}q &\Leftrightarrow \forall K \in \mathcal{K} : q \in K \Rightarrow p \in K \\ K \in \mathcal{K} &\Leftrightarrow \forall (p, q) \in \mathcal{Q} : q \in K \Rightarrow p \in K \end{aligned}$$

Applications of these concepts are, for example, a questionnaire, where people can agree or disagree to a statement, or an aptitude test, where people can

solve or fail to solve a question. In this paper, we use the latter interpretation to illustrate the algorithms. Solving an item is coded as 1 and failing to solve an item is coded as 0.

2.2 Probabilistic concepts

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 is capable of mastering. If careless errors or lucky guess guesses are committed, all kinds of response patterns may be generated. A probabilistic extension of the knowledge structure model covering random response errors is the basic local independence model in KST.

A quadruple $(Q, \mathcal{K}, p, r)$ is called a basic local independence model (BLIM) (Doignon and Falmagne (1999)) if and only if

1. $(Q, \mathcal{K})$ is a knowledge structure,

2. p is a probability distribution on $\mathcal{K}$, i.e., $p : \mathcal{K} \to]0,1[\,, K \mapsto p(K)$, 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 $(Q, \mathcal{K}, p)$, i.e., 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, i.e.,

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

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.

To each state $K \in \mathcal{K}$ is attached a probability $p(K)$ measuring the likelihood that an examinee is in state K (point 2). For $R \in 2^Q$ and $K \in \mathcal{K}$, $r(R, K)$ specifies the conditional probability of response pattern R for an examinee in state K (point 3). The item responses of an examinee are assumed to be independent given the knowledge state of the examinee. The response error probabilities β_q, η_q $(q \in Q)$ are attached to the items and do not vary with the knowledge states (point 4). The resulting probability distribution on the set of all response patterns is

$$\rho(R) = \sum_{K \in \mathcal{K}} r(R, K) p(K).$$

Note that the number of independent model parameters of the BLIM is $2|Q| + (|\mathcal{K}| - 1)$ ($|Q|$ parameters, each for careless error and lucky guess probabilities, and $|\mathcal{K}| - 1$ for the occurrence probabilities of the knowledge states). Because the size of $\mathcal{K}$ generally tends to be prohibitively large in practice, parameter estimation and model testing based on classical maximum likelihood methodology are not feasible in general (see, e.g., Ünlü, 2006).

Next, we consider a random sample of size m. The data are the absolute counts $m(R)$ of response patterns $R \in 2^Q$, i.e., $\mathbf{x} = (m(R))_{R \in 2^Q}$. The examinees are assumed to give their responses independent of each other. The true probability of occurrence $\rho(R)$ of any response pattern R is assumed to stay constant across the examinees, and to be strictly larger than zero. Then the data $\mathbf{x}$ are the realization of a random vector, $\mathbf{X} = (X_R)_{R \in 2^Q}$, which is distributed multinomial over 2^Q.

In other words, the probability of observing the data $\mathbf{x}$, i.e., the realizations $X_R = m(R)$, is

$$\begin{aligned}\mathbb{P}(\mathbf{X} = \mathbf{x}) &= \mathbb{P}(X_\emptyset = m(\emptyset), \ldots, X_Q = m(Q)) \\ &= \frac{m!}{\prod_{R \in 2^Q} m(R)!} \prod_{R \in 2^Q} \rho(R)^{m(R)},\end{aligned}$$

where $\rho(R) > 0$ for any $R \in 2^Q$, $\sum_{R \in 2^Q} \rho(R) = 1$, and $0 \leq m(R) \leq m$ for any $R \in 2^Q$, $\sum_{R \in 2^Q} m(R) = m$.

Chapter 3

Inductive item tree analysis

Data analysis methods are important procedures for deriving knowledge structures. There exist various methods such as the *di* coefficient by Kambouri et al. (1994) or the presently discussed IITA algorithms. We give a brief historical overview and present the ITA algorithm by van Leeuwe (1974), which is the predecessor of IITA. The three IITA algorithms are discussed and compared in simulated and real data examples.

3.1 History

The first variant of ITA was introduced by Airasian and Bart (1973); Bart and Krus (1973). The ITA algorithm was proposed by van Leeuwe (1974), and he developed especially the correlational agreement coefficient (CA). This is a fit measure, such as the *diff* coefficient in IITA, which is used for determining the best fitting quasi order.

Next, we give a sketch of the ITA algorithm (van Leeuwe, 1974).

We use the following notation ($m, n \in \mathbb{N}$):

$$Q \;:=\; \{I_l : 1 \leq l \leq n\} \;\text{ set of dichotomous items,}$$

$$P := \{P_k : 1 \leq k \leq m\} \text{ sample of subjects,}$$

$$D := (d'_{kl}) \text{ corresponding binary } (= 0/1)\ m \times n\text{-data matrix,}$$

and, for every $(I_i, I_j) \in Q \times Q$ $(1 \leq i, j \leq n)$, the 2×2-table notation

$\mathbf{I_i} \setminus \mathbf{I_j}$	**1**	**0**
1	a_{ij}	b_{ij}
0	c_{ij}	d_{ij}

with $a_{ij}, b_{ij}, c_{ij}, d_{ij} \in \mathbb{N} \cup \{0\}$; in respective order, the absolute frequencies of subjects solving items I_i and I_j $[a_{ij}]$, solving I_i, not I_j $[b_{ij}]$, solving I_j, not I_i $[c_{ij}]$, and solving neither I_i, nor I_j $[d_{ij}]$. Then, the *ITA-rule* for generating binary relations $\leq_L$ $(0 \leq L \leq m)$ is given by

$$I_i \leq_L I_j :\Longleftrightarrow c_{ij} \leq L.$$

This L $(0 \leq L \leq m)$ is called *tolerance level*. The ITA-rule represents STEP1 of ITA. The latter consists of five steps, STEP1-STEP5:

STEP1 Determine the binary relations $\leq_L$ for $L = 0, 1, \ldots, m$.

STEP2 From the $\leq_L$ $(0 \leq L \leq m)$ remove those that are not transitive.

STEP3 Set a critical value $0 < c \leq 1$ for the proportions, p_L, of subjects not contradicting the respective surmise relations $\leq_L$ in STEP2.

STEP4 From the surmise relations in STEP2 remove those with $p_L < c$.

STEP5 From the remaining surmise relations (after STEP4)—$\leq_0$ is always contained—, select one with maximal $CA(\leq, D)$-value.

The *CA* coefficient is defined as:

$$CA(\leq, D) := 1 - \frac{1}{n(n-1)} \sum_{i<j} (r_{ij} - r^*_{ij})^2,$$

where r_{ij} is the Pearson correlation and

$$r^*_{ij} := \begin{cases} 1 & : \text{ if } i = j \\ \sqrt{1-p_i)p_j/(1-p_jp_i} & : \text{ if } i \leq j \wedge j \not\leq i \\ \sqrt{1-p_j)p_i/(1-p_ip_j} & : \text{ if } i \not\leq j \wedge j \leq i \\ 0 & : \text{ otherwise} \end{cases}$$

The correlational agreement coefficient is used as a goodness-of-fit measure to handle the selection problem in STEP5. From the remaining surmise relations select an "optimal" one, i. e., one with maximal $CA(\leq, D)$-value.

3.2 Original inductive item tree analysis algorithm

IITA is an enhancement of the ITA algorithm. The idea behind IITA is to generate a more appropriate set of competing quasi orders and to construct a theoretically sound fit measure for determining the most adequate quasi order.

3.2.1 Original algorithm

One of the main parts of IITA is the inductive generation of surmise relations (giving the algorithm its name). 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 R in the data matrix D contradicting $j \rightarrow i$. Based on these values, binary relations $\sqsubseteq_L$ for $L = 0, ..., m$ are defined (note that m is the sample size). Let $i \sqsubseteq_0 j :\Leftrightarrow b_{ij} = 0$. The relation $\sqsubseteq_0$ is transitive, and based on that, all the other transitive relations $\sqsubseteq_L$ are constructed inductively.

Assume $\sqsubseteq_L$ is a transitive relation. Define the set

$$S_{L+1}^{(0)} := \{(i,j)|b_{ij} \leq L+1 \wedge i \not\sqsubseteq_L j\}.$$

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 the item pairs in $S_{L+1}^{(0)}$, those are excluded that cause an intransitivity in $\sqsubseteq_L \cup S_{L+1}^{(0)}$, and the remaining item pairs (of $S_{L+1}^{(0)}$) are referred to as $S_{L+1}^{(1)}$. Then, from the item pairs in $S_{L+1}^{(1)}$, those are excluded that cause an intransitivity in $\sqsubseteq_L \cup S_{L+1}^{(1)}$, and the remaining item pairs (of $S_{L+1}^{(1)}$) are referred to as $S_{L+1}^{(2)}$. This process continues iteratively, say k times, until no intransitivity is caused anymore (i.e., k is the smallest non-negative integer such that $S_{L+1}^{(k)} = S_{L+1}^{(l)}$ for all $l > k$). The generated relation $\sqsubseteq_{L+1} := \sqsubseteq_L \cup S_{L+1}^{(k)}$ is then transitive by construction. Because $\sqsubseteq_0$ is reflexive, all generated relations are. Hence $\sqsubseteq_L$ for $L = 0, ..., m$ are quasi orders. They constitute the selection set $\{\sqsubseteq_L : L = 0, ..., m\}$ of the IITA procedure.

Besides the construction of the quasi orders, it is very important to find that quasi order which fits the data best. In IITA, the idea is to estimate the number of counterexamples for each quasi order, and to find, over all competing quasi orders, the minimum value for the discrepancy between the observed and expected numbers of counterexamples.

Let

$$p_i := |\{R \in D|i \in R\}|/m$$

be the relative solution frequency of an item i. A violation of an underlying implication is only possible due to random errors. To compute the expected number of counterexamples, b_{ij}^*, error probabilities are needed. In this algorithm, the error probabilities are assumed to be equal for all items. This

single error rate is estimated by

$$\gamma_{\sqsubseteq_L} := \frac{\sum\{b_{ij}/(p_j m) | i \sqsubseteq_L j \wedge i \neq j\}}{(|\sqsubseteq_L| - n)},$$

where $|\sqsubseteq_L| - n$ is the number of non-reflexive item pairs in $\sqsubseteq_L$ (note that n is the number of items).

Under every relation of the selection set, the algorithm computes the expected number of counterexamples for each (non-reflexive) item pair. If the relation $\sqsubseteq_L$ provides an implication $j \rightarrow i$, meaning $i \sqsubseteq_L j$, the expected number of counterexamples is computed by $b^*_{ij} = \gamma_L p_j m$. If $(i, j) \notin \sqsubseteq_L$, no dependency between the two items is assumed, and $b^*_{ij} = (1-p_i)p_j m(1-\gamma_L)$. In this formula, $(1-p_i)p_j m$ is the usual probability for two independent items, and the factor $1 - \gamma_L$ is assumed to state that no random error occurred. As we discuss later in detail, the main criticism on the algorithm is on the used estimates b^*_{ij}.

A measure for the fit of each relation $\sqsubseteq_L$ to the data matrix D is the *diff* coefficient. It is defined as

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

It gives the averaged sum of the quadratic differences between the observed and expected numbers of counterexamples under the relation $\sqsubseteq_L$. The smaller the *diff* value the better is the fit of the relation to the data. Therefore, the IITA algorithm looks for the smallest value of the *diff* coefficient and returns the corresponding quasi order.

Some remarks are in order with respect to the definition of the *diff* coefficient.

1. The crucial constituent measuring the discrepancy between the observed and expected numbers of counterexamples is $\sum_{i \neq j}(b_{ij} - b^*_{ij})^2$.

The constant factor $1/(n(n-1))$, however, could be replaced by any other (non-zero) constant without affecting the final surmise relation returned by the IITA algorithm. The same quasi order would be obtained independent of what constant is used in the formulation of the coefficient. (Note that such a logic, mathematically at least, would also apply to other selection criteria such as AIC or BIC.) To keep the discussion of the three IITA algorithms in terms of the *diff* coefficient comparable, of course the *same* constant must be used throughout. (Comparing values of *diff* coefficients formulated for different constants would be distorted.) In this paper, all three algorithms use $1/(n(n-1))$, and relative to this (fixed) constant, the *diff* coefficient can be interpreted as the average quadratic difference between the observed and expected numbers of counterexamples, and compared across the algorithms.

2. The fit criterion underlying the *diff* coefficient is to match the observed, two-dimensional summaries b_{ij} of the data. Of course, the ultimate purpose of using the *diff* coefficient (i.e., the corresponding fit criterion) is to select that quasi order which best resembles the underlying (true) relation. Assuming the *diff* coefficient not to be informative for the quality of the returned solution would invalidate the rationale behind the IITA procedure. The selection measure, to some degree, has to reflect the underlying relation. Stated differently, it makes sense, and is important, to address and investigate the relationship between the fit criterion (decision rule) on the one hand, and the underlying structure on the other. Since selection is based on the minimum value of the *diff* coefficient, it is interesting to see whether and to what degree smaller *diff* values (better values of the fit criterion) do correlate with better

reconstructions of underlying quasi orders. (An answer to the latter question is by no means obvious a priori.)

3.2.2 Problems of the original algorithm

The inductive construction of the quasi orders is stated as one of the main advantages of this algorithm (Schrepp, 1999, 2003). However, the inductive construction can be criticized as follows. It is possible that two implications would cause together an intransitivity, but not if added separately. Consider on a set of three items $\{a, b, c\}$ the implication $b \to c$ (in addition to the reflexive ones), representing $\sqsubseteq_L$. Assume that the implications $a \to c$ and $c \to b$ are the possible candidates to be added in the next step $L + 1$. Together these implications lead to an intransitivity ($a \to b$ is not contained in $\sqsubseteq_L \cup S_{L+1}^{(0)}$), and the procedure excludes both implications, until $a \to b$ is added. However, each of the two implications, $a \to c$ and $c \to b$, could be added separately, without $a \to b$ being added, such that transitivity is not violated. But the procedure does not incorporate this. Moreover, the underlying (correct) quasi order is not necessarily contained in the selection set of constructed quasi orders. In the simulation study reported in this paper (see Table 3.4 for individual figures), the underlying quasi orders are contained in the selection sets 57% of the trials. In the other 43% it is impossible to reveal the underlying quasi orders.

The major problem of the original IITA algorithm lies in the computation of the *diff* coefficient. It uses estimates b_{ij}^* of the expected numbers of counterexamples. Two problems arise in the calculation of these estimates. For $(i, j) \notin \sqsubseteq_L$, the estimate is $b_{ij}^* = (1 - p_i)p_j m(1 - \gamma_L)$. But the algorithm does not take two different cases into account, namely $(j, i) \notin \sqsubseteq_L$ and $(j, i) \in \sqsubseteq_L$. In the first case, independence holds, and a corrected estimator

is $b^*_{ij} = (1 - p_i)p_j m$. ('A corrected estimator' in this paper is understood as an estimator which avoids the inconsistencies that arise when using the original estimators, in the sense of the discussion in the next paragraph.) This estimator is used in the first version of IITA (Schrepp, 1999, 2002), but is changed in Schrepp (2003). (Using the product of individual marginal probabilities is the common approach in statistics when independence is present, for instance in the analysis of two-way contingency tables.) In the second case, independence cannot be assumed, as $j \sqsubseteq_L i$. In Schrepp (2003), this problem is briefly mentioned, but not further pursued or even solved. This, in particular, explains why the original IITA version gives bad results when longer chains of items are present in the underlying quasi order (Schrepp, 1999). As explained in detail in the next section, a corrected estimator b^*_{ij} is $(p_j - (p_i - p_i\gamma_{_L}))m$, instead of $(1 - p_i)p_j m(1 - \gamma_{_L})$.

The estimates b^*_{ij} of the original algorithm not only are lacking interpretation, but they do also lead to methodological inconsistencies. Consider the case $(i, j) \notin \sqsubseteq_L$ and $(j, i) \in \sqsubseteq_L$. The observed number of people solving item j is $p_j m$, and using the estimate $(1 - p_i)p_j m(1 - \gamma_{_L})$ of the expected number of people solving item j and failing to solve item i, the expected number of people solving both items is estimated by $p_j m - (1 - p_i)p_j m(1 - \gamma_{_L})$. Another estimate of the expected number of people solving both items is $p_i m - p_i m\gamma_{_L}$. In the same manner, for the expected number of people failing to solve both items, the two estimates $(1 - p_j)m - p_i m\gamma_{_L}$ and $(1 - p_i)m - (1 - p_i)p_j m(1 - \gamma_{_L})$ are derived. If $\gamma_{_L} = 0$ and $p_i = 0$, it holds

$$p_j m - (1 - p_i)p_j m(1 - \gamma_{_L}) = p_i m - p_i m\gamma_{_L}$$

and

$$(1 - p_j)m - p_i m\gamma_{_L} = (1 - p_i)m - (1 - p_i)p_j m(1 - \gamma_{_L}),$$

and the respective estimates do coincide. If $\gamma_{_L} \neq 0$ or $p_i \neq 0$, these equations hold if and only if $p_i = 1$. Apart from these exceptional cases, which are rather rare, the estimation scheme of the original algorithm mostly leads to inconsistent results. In other words, fixing the marginals of the two-by-two table for an item pair $(i, j) \notin \sqsubseteq_{_L}$ and $(j, i) \in \sqsubseteq_{_L}$, and the two entries of it for which estimates are proposed, the results are in contradiction for nearly all datasets. (In the sequel, the expression 'inconsistent estimator' is used to refer to these methodological inconsistencies. It should not be confused with 'an estimator that is not consistent', in the sense of the consistency property in point estimation.)

3.3 Corrected and minimized corrected inductive item tree analysis algorithms

In Sargin and Ünlü (2009a) the problems mentioned in Section 3.2.2 are discussed and a corrected estimation scheme is proposed (see Section 3.3.1). Furthermore an optimization regarding the *diff* coefficient is introduced (see Section 3.3.2). Simulation studies, along with applications to empirical data, comparing the three IITA algorithms are presented in Sargin and Ünlü (2009a,b); Ünlü and Sargin (2008a) (see Section 3.4). In Ünlü and Sargin (2008a) the *diff* coefficient is interpreted as a maximum likelihood estimator. This estimator possesses good asymptotic properties (see Section 3.5). In Particular, inferential statistics can be proposed for the *diff* coefficient (see Section 3.6).

3.3.1 Corrected estimation

In this section, we introduce the corrected estimators b^*_{ij} for the expected numbers of counterexamples. These are very important for computing the *diff* coefficient, which is the fit measure for finding the best quasi order. A correct choice for b^*_{ij} for $(i,j) \notin \sqsubseteq_L$ depends on whether $(j,i) \notin \sqsubseteq_L$ or $(j,i) \in \sqsubseteq_L$.

- If $(i,j) \notin \sqsubseteq_L$ and $(j,i) \notin \sqsubseteq_L$, set $b^*_{ij} = (1-p_i)p_j m$. As stated in Section 2.2, independence holds, and the additional factor $(1-\gamma_L)$ is omitted.

- If $(i,j) \notin \sqsubseteq_L$ and $(j,i) \in \sqsubseteq_L$, set $b^*_{ij} = (p_j - (p_i - p_i\gamma_L))m$. This estimator is derived as follows. The observed number of people who solve item i is $p_i m$. Hence the estimated number of people who solve item i and item j is $p_i m - b^*_{ji} = (p_i - p_i\gamma_L)m$. (Note that $(j,i) \in \sqsubseteq_L$, and the estimator is $b^*_{ji} = p_i\gamma_L m$.) Eventually this gives the estimate $b^*_{ij} = p_j m - (p_i - p_i\gamma_L)m = (p_j - (p_i - p_i\gamma_L))m$. This estimator not only is mathematically motivated, but is also interpretable. The first term, $p_j m$, gives the number of people solving item j. The second term, $(p_i - p_i\gamma_L)m$, stands for the number of people solving both items, because $p_i m$ is the number of people solving item i, and $p_i\gamma_L m$ represents the number of people solving item i and failing to solve item j.

3.3.2 Minimizing the fit measure

Let the *diff* coefficient be based on the corrected estimators. We discuss minimizing the *diff* coefficient as a function of the error probability γ_L, for every quasi order $\sqsubseteq_L$. The idea is to use the corrected estimators and to optimize the fit criterion underlying the selection of competing quasi orders. The fit

measure then favors quasi orders that lead to smallest minimum discrepancies, or equivalently, largest maximum matches, between the observed and expected two-dimensional summaries, b_{ij} and b^*_{ij}, respectively. (Note that the IITA algorithms include the fit measure as a defining main constituent.)

The *diff* coefficient can be decomposed as

$$\begin{aligned}
\mathit{diff} &= \frac{\sum_{i\neq j}(b_{ij}-b^*_{ij})^2}{n(n-1)} \\
&= \frac{\sum_{i\not\sqsubseteq_L j \wedge j\sqsubseteq_L i}\left[b_{ij}^2 - 2b_{ij}(p_j - p_i + p_i\gamma_{_L})m + (p_j - p_i + p_i\gamma_{_L})^2m^2\right]}{n(n-1)} \\
&\quad + \frac{\sum_{i\not\sqsubseteq_L j \wedge j\not\sqsubseteq_L i}\left[b_{ij} - (1-p_i)p_jm\right]^2}{n(n-1)} \\
&\quad + \frac{\sum_{i\sqsubseteq_L j}\left[b_{ij}^2 - 2b_{ij}p_j\gamma_{_L}m + (p_j\gamma_{_L})^2m^2\right]}{n(n-1)}.
\end{aligned}$$

Setting equal to zero the derivative of the *diff* coefficient with respect to $\gamma_{_L}$ gives

$$\begin{aligned}
0 &= \frac{\sum_{i\not\sqsubseteq_L j \wedge j\sqsubseteq_L i}\left[-2b_{ij}p_im + 2p_ip_jm^2 - 2p_i^2m^2 + 2p_i^2m^2\gamma_{_L}\right]}{n(n-1)} \\
&\quad + \frac{\sum_{i\sqsubseteq_L j}\left[-2b_{ij}p_jm + 2p_j^2m^2\gamma_{_L}\right]}{n(n-1)}.
\end{aligned}$$

This is equivalent to

$$\begin{aligned}
0 &= \underbrace{\sum_{i\not\sqsubseteq_L j \wedge j\sqsubseteq_L i}\left[-2b_{ij}p_im + 2p_ip_jm^2 - 2p_i^2m^2\right]}_{=:x_1} \\
&\quad + \underbrace{\sum_{i\sqsubseteq_L j} -2b_{ij}p_jm}_{=:x_2} \\
&\quad + \gamma_{_L}\underbrace{\sum_{i\not\sqsubseteq_L j \wedge j\sqsubseteq_L i} 2p_i^2m^2}_{=:x_3} \\
&\quad + \gamma_{_L}\underbrace{\sum_{i\sqsubseteq_L j} 2p_j^2m^2}_{=:x_4}.
\end{aligned}$$

Solving for $\gamma_{_L}$ results in

$$\gamma_{_L} = -\frac{x_1 + x_2}{x_3 + x_4}.$$

Note that this expression always gives a value in $[0, 1]$. This error probability can now be used for an alternative IITA procedure, in which a minimized *diff* value is computed for every quasi order.

3.4 Comparisons of the three algorithms

The three algorithms are the original IITA version by Schrepp (2003), and the corrected and minimized corrected IITA versions introduced above. In the following, the performances of these procedures are compared in a simulation study.

3.4.1 Settings of the simulation study

The settings

Throughout the simulation study nine items are used. The general simulation scheme consists of three parts. First, quasi orders are generated randomly. Second, each of these quasi orders is used for simulating the data. Third, the three algorithms are applied to and compared on that data.

More precisely:

1. All reflexive pairs are always added to the relation $\mathcal{R}$. A constant δ is set randomly (detailed below), which gives the probability for adding each of the remaining 72 item pairs to the relation. The transitive closure $\sqsubseteq$ of this relation $\mathcal{R}$ is computed, and is the underlying (true) quasi order.

2. From the set $\{K \subset Q : (i \sqsubseteq j \wedge j \in K) \rightarrow i \in K\}$ of all response patterns consistent with $\sqsubseteq$, an element is drawn randomly. For this drawn pattern all entries are changed from 1 to 0 or from 0 to 1, with a same prespecified error probability τ. This is repeated m times to generate a data matrix. (Part 2 is simulating with a special case of the BLIM.)

3. The three algorithms are applied to the simulated data. They are compared with respect to two criteria: the symmetric differences between the data analysis solutions of the algorithms and the underlying quasi order, and the numbers of erroneously detected implications.

The following settings are made in the simulation study. The error probability τ takes the values 0.03, 0.05, 0.08, 0.10, 0.15, and 0.20. The sample sizes 50, 100, 200, 400, 800, 1600, and 6400 are used. For each combination of these settings, 1000 simulations are made. In each of these simulations, an underlying quasi order is generated, a data matrix is simulated, and for each of the three algorithms the data analysis solution is derived.

Important changes made to the simulation study in Schrepp (2003)

The above simulation scheme replicates the one described in Schrepp (2003). However, the following important changes are made. Schrepp (2003) draws δ randomly from the entire unit interval. This leads to the following problem. For δ values greater than (approximately) 0.42, the average number of non-reflexive implications contained in the underlying quasi order already turns out to be not less than (approximately) 70. This can be seen from Figure 3.1.

Figure 3.1 shows the average number of non-reflexive implications as a function of δ. For each δ value ranging from 0 to 1, in steps by 0.01, 100

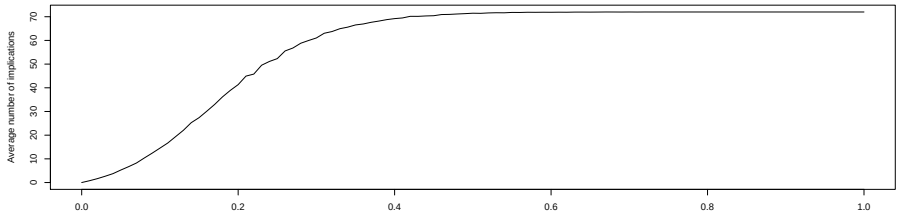

Figure 3.1: Average number of non-reflexive implications as a function of δ. The δ values range from 0 to 1, in steps by 0.01. For each δ value, 100 quasi orders are generated, and the corresponding average number of non-reflexive implications is shown.

quasi orders are generated, and the corresponding average number of non-reflexive implications is calculated. In particular, Figure 3.1 demonstrates that Schrepp's choice of δ values mostly results in generating large quasi orders: 58% of the computed average numbers of non-reflexive implications are at least 70; 29% are even equal to the maximum 72 (yielding the set $Q \times Q$ of all possible item pairs). This definitely does not come from a reasonably representative sample of the collection of all quasi orders (cf. also the remarks below), and leads to substantially biased results as we describe in this paper.

To accommodate this problem, we pursue the following sampling. The δ values are drawn from a normal distribution with $\mu = 0.16$ and $\sigma = 0.06$. Values less than 0 or greater than 0.3 are set to 0 or 0.3, respectively. Figure 3.2 shows the average numbers of non-reflexive implications calculated for 100 generated quasi orders to 500 δ values drawn according to our sampling. Compared to the plot of Figure 3.1 (random sampling from the entire unit interval), the results reported in Figure 3.2 most probably come from a reasonably representative, in any case considerably improved, sample of quasi orders (see also the remarks that follow).

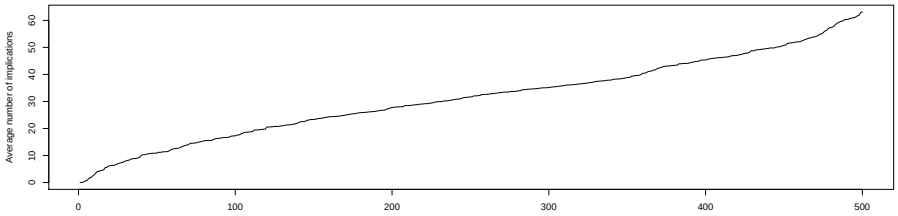

Figure 3.2: Average numbers of non-reflexive implications calculated for 100 generated quasi orders to 500 δ values drawn according to our sampling. Points are ordered by average number of non-reflexive implications.

Some remarks are in order regarding the representativeness of samples of quasi orders drawn in such simulation studies as for investigating IITA type data analysis methods. The three 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. However, defining representativeness as sampling uniformly from this collection (i.e., drawing each element with the same probability) is a theoretical concept, which is basically not feasible. A general approach to handling representativeness of samples of quasi orders is through investigating tractable consequences of that theoretical definition. For instance, a necessary condition following from a uniform distribution on the set of all surmise relations, by and large, is having a (not necessarily symmetric) bell-shaped type of distribution on the set of all (attained) numbers of non-reflexive implications, centered around, approximately, the middle of the scale, and decreasing towards the edges of the scale. This reflects the fact that, on the whole, there are many more surmise relations around the middle of the scale than around the edges. In addition, at least in this study with nine items, there seems to be more surmise relations around the left edge than the right. (There are

surmise relations containing $0, 1, 2, \ldots, 10$ non-reflexive implications, whereas there are surmise relations with just 72, 64, or 58 implications.) Correspondingly, we expect that more mass of the resulting distribution is located around the left than the right edge.

To compare the two ways of sampling the δ values, unit interval versus normal, Figure 3.3 shows histograms of the average numbers of non-reflexive implications depicted in Figures 3.1 and 3.2; upper and lower plots, respectively.

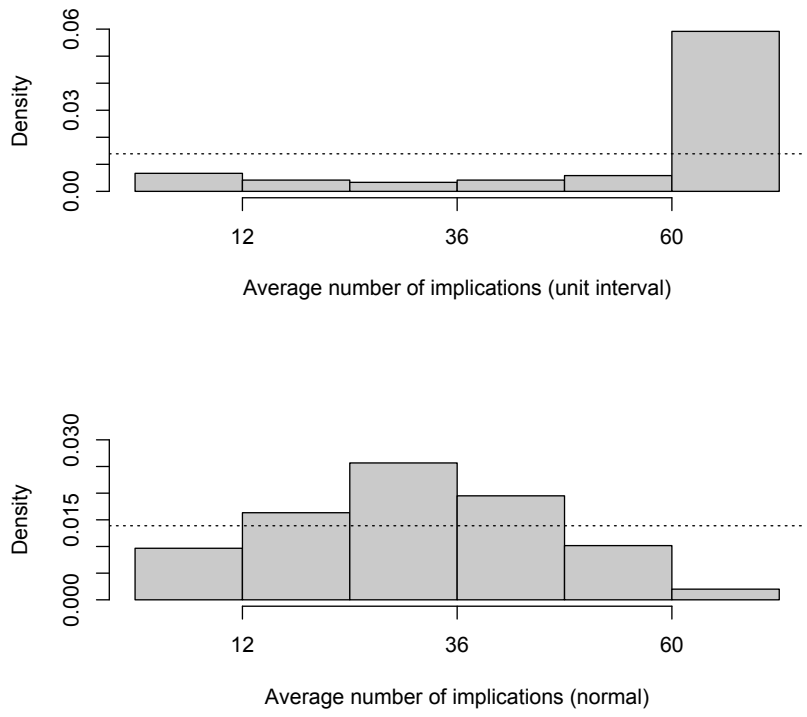

Figure 3.3: Histograms of the average numbers of non-reflexive implications for the unit interval and normal sampling methods (upper and lower plots, respectively). The dotted line shows the probability density function of the uniform distribution on the interval $[0, 72]$.

The histograms corroborate what we have stated about the two sampling methods. The random sampling using the entire unit interval is far from producing reasonably representative samples of quasi orders, and without question, the normal sampling clearly improves on the latter.

The above discussion is, to our knowledge, the first of a kind so far presented on the issue of assessing the representativeness of samples of quasi orders drawn in simulation studies for investigating the IITA algorithms. Much of the discussion here is meant as a starting point for further research on this issue. Work in this direction is important but still lacking.

3.4.2 Results of the simulation study

Average symmetric differences

First we compare the three IITA algorithms with respect to the average symmetric differences. For each of the three algorithms, for every combination of error probability and sample size, the mean of the numbers of elements in the 1000 symmetric differences between the underlying quasi orders and the data analysis solutions is computed; in the sequel referred to as *dist* value. This summary statistic is reported in Table 3.1; first, second, and third lines refer to the original, corrected, and minimized corrected IITA algorithms, respectively. (In addition, the means of the 1000 *diff* values obtained for the data analysis solutions are listed in parentheses.)

Table 3.1 shows the following results:

1. The average *dist* values are quite similar (maximum discrepancy of 0.76) for the corrected and minimized corrected algorithms. Moreover, in 24 of the 42 combinations the corrected algorithm performs better, in three they perform (almost) identically, and in 15 the performance of

Table 3.1: Average *dist* values under original, corrected, and minimized corrected IITA algorithms (first, second, and third lines, respectively; average *diff* values in parentheses)

	Sample size						
	50	100	200	400	800	1600	6400
τ							
0.03	3.71(3.79)	2.89(11.67)	2.14(41.16)	2.29(152.10)	1.87(599.61)	1.69(2438.30)	1.99(36528.70)
	5.44(1.66)	5.18(4.28)	4.54(11.58)	4.86(38.37)	4.30(128.13)	3.90(489.21)	3.81(7414.34)
	5.44(1.58)	5.30(4.04)	4.67(10.76)	5.11(34.92)	4.77(114.65)	4.58(438.49)	4.51(6635.50)
0.05	4.24(3.73)	4.05(11.32)	2.87(38.28)	2.59(138.35)	2.11(517.96)	1.77(2094.56)	1.10(32968.50)
	6.89(1.93)	5.69(5.13)	4.91(14.35)	5.09(44.49)	4.22(152.74)	4.33(587.63)	3.63(8172.90)
	6.90(1.81)	5.67(4.68)	5.02(12.67)	5.40(39.24)	4.91(130.82)	4.89(489.52)	4.39(6892.42)
0.08	7.66(3.96)	5.95(12.18)	5.12(41.09)	4.91(149.30)	4.43(606.56)	4.47(2392.01)	3.69(37444.40)
	8.61(2.24)	6.36(6.05)	5.70(17.35)	5.28(52.26)	4.46(196.09)	4.5(702.46)	3.99(9870.19)
	8.45(2.09)	6.30(5.58)	5.90(15.38)	5.70(45.48)	4.88(163.70)	5.06(582.79)	4.55(8102.32)
0.10	9.01(4.23)	7.87(12.71)	7.89(45.51)	6.14(166.79)	6.67(682.10)	6.37(2808.66)	6.87(44472.91)
	9.61(2.43)	7.65(6.75)	6.37(18.70)	5.37(59.49)	5.26(203.67)	4.35(765.31)	4.25(11373.20)
	9.60(2.31)	7.47(6.21)	6.42(16.84)	5.66(52.69)	5.58(175.46)	4.85(644.15)	4.58(9491.62)
0.15	16.68(4.55)	14.96(14.81)	14.22(58.53)	13.88(221.76)	15.06(935.71)	14.50(3664.33)	14.92(62646.07)
	12.18(2.59)	10.11(7.45)	7.77(21.90)	7.11(67.43)	6.06(250.24)	5.89(877.53)	5.10(14659.14)
	11.93(2.48)	9.89(7.09)	7.71(20.49)	7.11(62.26)	6.06(226.58)	6.08(790.18)	5.21(13027.16)
0.20	23.38(4.53)	25.41(16.69)	24.93(62.96)	24.02(276.97)	23.72(1148.28)	24.65(4699.31)	23.46(76842.70)
	14.81(2.59)	11.40(7.34)	9.81(22.34)	8.00(71.94)	7.96(254.09)	6.79(930.46)	6.79(14769.23)
	14.68(2.52)	11.36(7.12)	9.62(21.53)	7.91(68.79)	7.93(240.95)	6.75(879.06)	6.58(13893.23)

the minimized corrected version is better. In particular, the minimized corrected version gives smaller *dist* values for an error probability of 0.20. On average, however, the corrected algorithm shows a smaller *dist* value.

2. For the very small error rates 0.03 and 0.05, the original version gives better *dist* results than the corrected and minimized corrected algorithms (however, see Table 3.2 for worse $dist^*$ results). It may seem surprising that, though of the inconsistent estimators used in the original IITA algorithm, this algorithm gives better results. We suppose that the inconsistent estimation, in the case of very small error rates, has a considerably less negative effect for the underlying quasi order

than for the other relations; see also 'Important remarks regarding the simulation study in Schrepp (2003)' in this section below. However, for $\tau = 0.08$, the results are approximately the same, and for the higher error rates $0.10, 0.15$ and 0.20, the original version is outperformed. (It is important to note that for small *dist* values the underlying quasi orders are still reconstructed with acceptable accuracy, as in the case of the corrected and minimized corrected algorithms for small error rates. By contrast, the underlying quasi orders are clearly missed by the original algorithm for high error rates, due to the very large *dist* values.) On average, the corrected and minimized corrected versions show smaller *dist* values than the original algorithm.

3. The differences in the cases when the corrected and minimized corrected algorithms perform better are substantially larger than the differences obtained when the original version performs better. This is true not only in absolute differences, but also in relative. For instance, for the error rate 0.03 and sample size 50, the *dist* value for the corrected algorithm is 1.47 times larger than the *dist* value for the original, whereas for the error rate 0.20 and sample size 50, the corrected version is 1.58 times better. The ratio increases with increasing sample size. For the error rate 0.03 and sample size 6400, the *dist* value for the corrected algorithm is 1.91 times larger than the *dist* value for the original, whereas for the error rate 0.20 and sample size 6400, the corrected version is 3.46 times better.

4. With increasing sample size, the improvements obtained for the two new algorithms are greater than the improvements for the original algorithm. For $\tau = 0.10$, for instance, the original algorithm improves

from a *dist* value of 9.01 to 6.87 (difference of 2.14), the corrected algorithm from a value of 9.61 to 4.25 (difference of 5.36), and the minimized corrected version from 9.60 to 4.58 (difference of 5.02).

5. An interesting observation is the following one. For our two algorithms, for any two error probabilities, the differences between the *dist* values decrease as the sample size increases. For the original algorithm, these differences range around a constant. For instance, consider the error probabilities 0.05 and 0.15. The sequence of differences for the original algorithm is 12.44, 10.91, 11.35, 11.29, 12.95, 12.73, and 13.82. The sequences for the other algorithms are 5.29, 4.43, 2.86, 2.02, 1.84, 1.56, and 1.47 (corrected version), and 5.03, 4.22, 2.69, 1.71, 1.15, 1.19, and 0.82 (minimized corrected version).

6. Table 3.1 serves to compare the different IITA algorithms with respect to the average *dist* values, which is the main comparison that is made here. Nevertheless, inspecting the average *diff* values gives the following information. For all combinations of settings, the same ranking is obtained. The minimized corrected version gives the smallest average *diff* value, second comes the corrected version, and the original algorithm has the largest *diff* value. Hence, the matches between the observed and expected numbers of counterexamples (the fit criterion underlying *diff*) can be ranked accordingly. It is also seen that smaller (average) *diff* values do not necessarily imply smaller (average) *dist* values.

To give more information about the performances of the IITA algorithms, we also present the symmetric differences at the level of knowledge states ($dist^*$). This is justified and important since, according to Birkhoff (1937)'s theorem, there exists a one-to-one correspondence between quasi orders and

their corresponding knowledge structures. The results obtained at the two levels do differ in general; for example, the original IITA algorithm may have moderately lowest $dist$ but considerably highest $dist^*$ values (cf. Tables 3.1 and 3.2). This can be explained primarily by the following two facts, which are true especially when the error probabilities are small (see 'Important remarks regarding the simulation study in Schrepp (2003)' below).

1. For an underlying quasi order with many implications, missing the true relation already implies a large $dist$ value; there are large differences of the sizes of the true and neighboring quasi orders. The corresponding true knowledge structure has few knowledge states, and hence there are not large differences of the sizes of the true and neighboring quasi ordinal knowledge spaces. Compared to the other two algorithms, the original IITA algorithm produces good results specifically for quasi orders with many implications, therefore yielding relatively smaller $dist$ than $dist^*$ values.

2. For an underlying quasi ordinal knowledge space with many knowledge states, missing the true knowledge structure already implies a large $dist^*$ value; there are large differences of the sizes of the true and neighboring quasi ordinal knowledge spaces. The corresponding true relation has few implications, and hence there are not large differences of the sizes of the true and neighboring quasi orders. Compared to the other two algorithms, the original IITA algorithm produces bad results specifically for quasi orders with few implications, therefore yielding relatively larger $dist^*$ than $dist$ values.

We performed the simulation study described in Schrepp (2003), with the following changes. The error probability τ takes the values $0.03, 0.05, 0.08,$

and 0.15. The sample sizes 50, 400, and 1600 are used. For every combination of these settings, 100 simulations are made. For each of the three algorithms, for every combination of error probability and sample size, the mean of the numbers of elements in the 100 symmetric differences between the underlying knowledge structures and the knowledge structures obtained from data analysis is computed; in the sequel referred to as $dist^*$ value. The $dist^*$ values are reported in Table 3.2; first, second, and third lines refer to the original, corrected, and minimized corrected IITA algorithms, respectively.

Table 3.2: Average $dist^*$ values under original, corrected, and minimized corrected IITA algorithms (first, second, and third lines, respectively)

	Sample size		
	50	400	1600
τ			
0.03	13.94	10.31	13.85
	14.67	2.51	5.75
	14.23	2.58	6.45
0.05	33.55	26.49	22.67
	22.90	7.31	6.54
	22.30	7.74	6.99
0.08	60.45	63.79	79.21
	34.84	8.59	3.70
	29.04	8.77	3.94
0.15	120.88	173.30	182.00
	45.36	14.67	12.05
	40.27	12.85	7.66

Except for the error rate 0.03 and sample size 50, the corrected and minimized corrected IITA algorithms give clearly smaller $dist^*$ values than the original algorithm. Compared to the results in Table 3.1, even for the very small error rates the two new algorithms perform better than the original. For $\tau = 0.05$ and sample size 400, for instance, the original, corrected, and minimized corrected versions yield the $dist^*$ values 26.49, 7.31, and 7.74, respectively. Regarding the $dist^*$ statistic, hence for small error rates, the new IITA algorithms are more capable of reconstructing the underlying knowledge structure than the original algorithm. For the higher error rates, the $dist$ results in Table 3.1 being confirmed here using $dist^*$, the original version is clearly outperformed. Whereas the original algorithm solutions are far off from the underlying knowledge structures, the corrected and minimized corrected algorithms still produce reasonably accurate results. For $\tau = 0.15$ and sample size 400, for instance, the original, corrected, and minimized corrected versions give the $dist^*$ values 173.30, 14.67, and 12.85, respectively.

Average numbers of erroneously detected implications

From a practical point of view, it may be important to have only few false implications being added to the correct underlying quasi order (Schrepp, 2003, 2007). False implications can lead to wrong conclusions, and it may be inefficient to try to exclude them afterwards. (This should not be interpreted as a general statement, and of course, depends on the research context and the costs and risks associated with such errors.) In the following, we compare the three IITA algorithms with respect to the average numbers of erroneously detected implications. This summary statistic is reported in Table 3.3; first, second, and third lines refer to the original, corrected, and minimized corrected IITA algorithms, respectively.

Table 3.3: Average numbers of erroneously detected implications under original, corrected, and minimized corrected IITA algorithms (first, second, and third lines, respectively)

	Sample size						
	50	100	200	400	800	1600	6400
τ							
0.03	2.69	2.23	1.80	1.91	1.47	1.43	1.65
	1.82	0.92	0.48	0.38	0.23	0.17	0.18
	1.90	0.96	0.48	0.37	0.21	0.16	0.17
0.05	2.30	2.08	1.24	1.28	1.11	0.51	0.23
	2.10	1.14	0.64	0.45	0.30	0.20	0.13
	2.20	1.20	0.66	0.42	0.28	0.19	0.11
0.08	2.69	1.79	1.57	1.37	0.99	1.23	0.98
	2.26	1.47	0.92	0.58	0.42	0.40	0.35
	2.44	1.54	0.95	0.59	0.42	0.40	0.34
0.10	1.95	1.40	1.73	0.95	1.45	1.22	1.81
	2.30	1.49	0.92	0.73	0.55	0.50	0.46
	2.50	1.56	0.99	0.73	0.55	0.49	0.43
0.15	3.02	2.03	2.33	3.13	3.78	4.08	3.84
	2.57	1.72	1.28	1.08	0.97	0.82	0.82
	2.76	1.85	1.36	1.13	1.02	0.82	0.83
0.20	3.46	5.68	5.80	6.38	6.89	8.38	7.00
	2.71	2.08	1.52	1.27	1.09	1.11	0.99
	2.87	2.16	1.57	1.34	1.16	1.17	1.02

Table 3.3 shows the following results:

1. Except for $\tau = 0.10$ and sample sizes 50 and 100, the corrected and minimized corrected IITA algorithms yield smaller average numbers of falsely detected implications. For example, for the error rates 0.15 and 0.20, the original version is clearly outperformed. On average, the corrected and minimized corrected algorithms falsely detect 1.01 and 1.05 implications, respectively, while the original version adds 2.59 false implications.

2. The results are quite similar (maximum discrepancy of 0.20) for the corrected and minimized corrected algorithms. Moreover, in 25 of the 42 combinations the corrected algorithm performs better, in six they perform (almost) identically, and in 11 the performance of the minimized corrected version is better. For smaller sample sizes, the corrected algorithm performs better than the minimized corrected one. For larger sample sizes, there seems to be no noticeable difference.

3. The results for the corrected and minimized corrected versions improve for increasing sample sizes. The original version, however, jitters between smaller and larger values, with no decreasing trend observable for larger error probabilities. For $\tau = 0.10$, for instance, the sequences of decreasing values for the corrected and minimized corrected versions are 2.30, 1.49, 0.92, 0.73, 0.55, 0.50, and 0.46, and 2.50, 1.56, 0.99, 0.73, 0.55, 0.49, and 0.43, respectively. The sequence for the original version is 1.95, 1.40, 1.73, 0.95, 1.45, 1.22, and 1.81.

Important remarks regarding the simulation study in Schrepp (2003)

Some important remarks are in order regarding the simulation study in Schrepp (2003). The results reported in this simulation study are much better than the results we have obtained for the original IITA algorithm. There are substantial discrepancies between the average *dist* values and the average numbers of falsely detected implications. For instance, for $\tau = 0.08$ and sample size 200, Schrepp's study gives 1.67 and 0.09, respectively, while our simulation study yields 5.12 and 1.57. This can be explained by the following flaw in the simulation methodology in Schrepp (2003). As mentioned in Section 3.4.1, the choice of $(0, 1)$-uniformly distributed δ values leads to the problem that mostly large quasi orders are generated. The inconsistent estimation scheme of the original IITA algorithm now produces good results specifically for large quasi orders. For a large quasi order $\sqsubseteq$, there are predominantly the cases $i \sqsubseteq j$, for which correct estimators are used. For the cases $i \not\sqsubseteq j$, however, inconsistent estimators are applied, and hence the discrepancies between the observed and expected numbers of counterexamples are large. This implies that, for an underlying large quasi order, the *diff* values for small quasi orders of the selection set are large (pulling apart the *diff* value for the true quasi order from the *diff* values obtained for the other relations). As a result, the underlying quasi order is more frequently recovered. This is true particularly for smaller error probabilities.

That also explains why the original algorithm gives smaller *dist* values for the error rates 0.03 and 0.05 in our simulation study (see Table 3.1). In addition to pulling apart *diff* values because of distorted estimation, Note that in the case of a large number of implications in the underlying quasi order, there are large differences of the sizes of the true and the neighboring relations in the selection set (due to transitivity). For instance, for nine

items used in the simulation study, an underlying quasi order consisting of 64 implications has possible nearest neighbors which contain 58 or 72 implications, and the former even may not be included in the selection set. As a consequence, for an underlying large quasi order, missing the true relation already implies a large *dist* value.

Moreover, it is not astonishing that in Schrepp (2003) smaller average numbers of falsely detected implications are obtained. For quasi orders containing an average number of not less than 70 non-reflexive implications, there are, on average, no more than two implications left to be added erroneously.

A first assessment of the inductively generated selection set

Finally, we briefly summarize few results obtained from our simulation study concerning the quality of the inductive construction procedure for generating the selection set of competing quasi orders. Table 3.4 reports, for each combination of error probability and sample size, the numbers of times out of 1000 simulations the underlying quasi orders are contained in the selection sets. (Note that in all three IITA algorithms the same inductive construction procedure is used.)

Overall, the results get worse for larger error probabilities or smaller sample sizes. Note that these figures, strictly speaking, do not give information about reconstructing the underlying surmise relation with acceptable accuracy.

Table 3.4: Numbers of times (out of 1000) the underlying quasi orders are contained in the inductively generated selection sets

	Sample size						
	50	100	200	400	800	1600	6400
τ							
0.03	439	692	838	932	970	976	984
0.05	350	520	707	840	903	944	964
0.08	242	374	571	689	752	808	844
0.10	215	345	490	578	685	707	760
0.15	157	236	342	433	466	534	538
0.20	99	144	241	299	381	419	480

3.4.3 A second simulation study

In Sargin and Ünlü (2009b) a different simulation scheme is presented, we discuss this second simulation study and point out the differences between the simulation study previously described and this second simulation study.

Settings of the second simulation study

Except for the simulation of the underlying quasi order, the second simulation study uses the same settings as the first one. The simulation of the underlying quasi order had to be changed, because this first simulation study put to much emphasize on quasi orders with many implications. Nevertheless we describe the settings thoroughly.

Throughout the simulation study nine items are used. The general simulation scheme consists of three parts. First, quasi orders are generated

randomly. Second, each of these quasi orders is used for simulating the data. Third, the three algorithms are applied to and compared on that data. More precisely:

1. All reflexive pairs are always added to the relation $\mathcal{R}$. A constant $\delta := 0.285$ is set, giving the probability for adding an item pair to the relation. Whenever 19 implications are added to the relation $\mathcal{R}$, δ is set to $\delta - 0.08$. Finally, the transitive closure $\sqsubseteq$ of this relation $\mathcal{R}$ is computed, and is the underlying quasi order.

2. From the set $\{K \subset Q : (i \sqsubseteq j \wedge j \in K) \rightarrow i \in K\}$ of all response patterns consistent with $\sqsubseteq$, an element is drawn randomly. For this drawn pattern all entries are changed from 1 to 0 or from 0 to 1, with a same prespecified error probability τ.

3. The three algorithms are applied to the simulated data. They are compared with respect to two criteria: the symmetric differences between the data analysis solutions of the algorithms and the underlying quasi order at the level of items ($dist$) and knowledge states ($dist^*$), and the average *diff* values.

Part 1 only deviates from the simulation scheme in Sargin and Ünlü (2009a). There are 363 083 quasi orders for nine discriminable (isomorphic quasi orders are not considered) items (Brinkmann and McKay, 2007). It is impossible to use all of them in a sampling scheme. However, one can try to draw as representative as possible samples of quasi orders. The new simulation scheme takes into account that quasi orders with many and few implications can be obtained by only few combinations, while medium sized quasi orders have more possible combinations. The following two graphics (see Figures 3.4 and 3.5) illustrate the difference between the two simulation schemes.

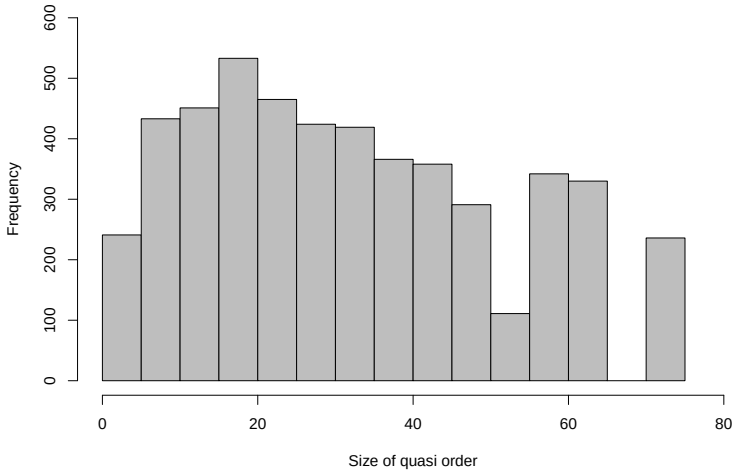

Figure 3.4: Histogram of the size of 5000 quasi orders simulated using the scheme described in Sargin and Ünlü (2009a). Quasi orders with many implications are overrepresented.

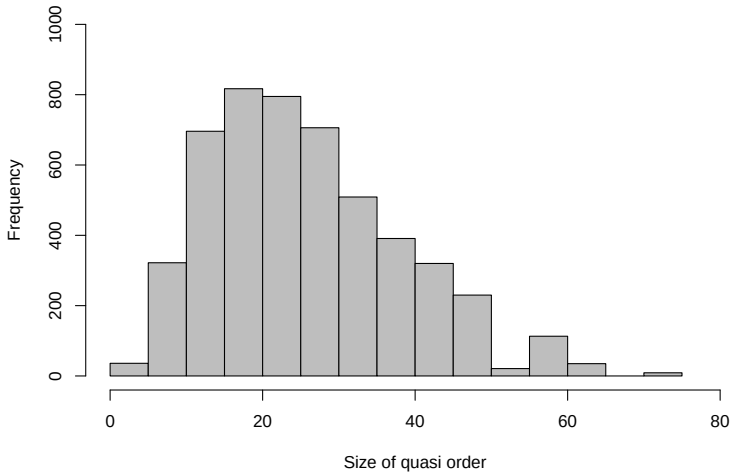

Figure 3.5: Histogram of the size of 5000 quasi orders simulated using the scheme described in Sargin and Ünlü (2009b).

One can see that the scheme depicted in Figure 3.5 puts more emphasis on the medium sized quasi orders, while Figure 3.4 indicates that the sampling scheme treats all sizes of quasi orders equally. Note that in the simulation study in Schrepp (2003) the very large quasi orders were highly overrepresented (see Figure 3.3).

Due to the representativeness of the simulation scheme of the underlying quasi orders, the highest quality of the simulation studies has the second simulation study (Sargin and Ünlü, 2009b), second comes the first simulation study (Sargin and Ünlü, 2009a), and worst results are obtained by the simulation study in Schrepp (2003).

Results of the second simulation study

For each of the three algorithms, for every combination of error probability and sample size, three summary statistics are computed. Two are the means of the numbers of elements in the 1000 symmetric differences between the underlying quasi order (at the level of items) and knowledge structure (at the level of states) and the data analysis solutions; in the sequel referred to as *dist* and $dist^*$ values. Another is the mean of the 1000 *diff* values obtained for the data analysis solutions. These summary statistics are reported in Tables 3.5 and 3.6; first, second, and third lines refer to the original, corrected, and minimized corrected IITA algorithms, respectively.

Table 3.5 and 3.6 show the following results:

1. For all settings the minimized corrected version gives the smallest *diff* value, second smallest is the corrected version, and largest is the original IITA algorithm. This shows that a better fit between the expected and observed numbers of counterexamples is obtained by the corrected and minimized corrected IITA algorithms.

Table 3.5: Average *dist* and *dist** (first and second entries, respectively) values under original, corrected, and minimized corrected IITA algorithms (first, second, and third lines, respectively) using the second simulation scheme

	Sample size						
	50	100	200	400	800	1600	6400
τ							
0.03	4.92, 21.17	3.69, 17.15	3.05, 18.83	2.53, 17.44	2.36, 20.45	2.31, 16.73	2.14, 15.21
	4.82, 13.51	3.54, 5.93	2.77, 4.64	2.66, 3.12	2.57, 2.78	2.81, 2.56	2.86, 2.85
	4.64, 9.88	3.42, 5.10	2.72, 3.86	2.78, 3.33	2.77, 3.11	3.00, 2.91	3.26, 3.43
0.05	6.38, 34.26	5.31, 38.43	4.58, 42.18	4.43, 45.48	4.62, 50.74	3.95, 44.03	3.28, 34.25
	5.97, 14.69	4.21, 8.13	3.65, 5.63	3.38, 4.85	2.99, 4.21	3.22, 4.00	2.82, 3.24
	5.67, 11.49	4.07, 7.14	3.51, 4.38	3.41, 4.01	3.16, 4.06	3.37, 3.84	3.18, 3.55
0.08	9.67, 72.56	9.33, 90.09	8.68, 103.23	8.77, 104.50	7.69, 100.19	6.94, 92.30	7.14, 86.80
	7.79, 22.07	5.66, 15.46	4.86, 11.36	3.94, 6.93	3.91, 5.68	3.83, 5.58	3.40, 4.46
	7.61, 18.35	5.43, 11.57	4.61, 8.29	3.91, 5.48	3.89, 4.88	3.95, 4.78	3.64, 4.32
0.10	12.89, 106.55	12.08, 129.17	11.72, 142.28	11.08, 146.13	10.27, 139.11	10.23, 132.77	9.82, 122.27
	9.20, 28.80	6.66, 17.28	5.31, 12.98	4.65, 9.69	4.38, 7.87	4.11, 6.54	3.87,6.17
	8.71, 23.12	6.41, 13.61	5.13, 10.10	4.57, 7.17	4.28, 6.47	4.12, 5.54	4.02, 5.45
0.15	18.07, 172.89	18.19, 198.54	18.71, 221.40	17.44, 221.57	18.29, 224.09	17.73, 208.66	16.60, 189.34
	12.14, 42.12	9.20, 34.39	7.43, 25.57	6.85, 21.56	6.07, 17.59	5.56, 16.46	5.22, 15.14
	11.84, 36.58	8.82, 28.67	7.12, 20,20	6.45, 15.59	5.85, 13.45	5.34, 12.11	4.99, 11.42
0.20	22.51, 232.76	24.40, 251.24	24.95, 260.73	25.10, 251.45	25.12, 243.85	25.68, 237.83	24.81, 206.81
	14.18, 59.44	12.24, 53.10	10.45, 41.90	9.18, 35.43	7.90, 32.52	8.00, 33.96	7.56, 30.88
	13.83, 52.56	12.01, 48.76	20.23, 37.54	8.86, 31.44	7.64, 27.95	7.63, 29.29	7.19, 26.29

Table 3.6: Average *diff* value under original, corrected, and minimized corrected IITA algorithms (first, second, and third lines, respectively) using the second simulation scheme

	Sample size						
	50	100	200	400	800	1600	6400
τ							
0.03	3.70	11.92	43.37	161.14	618.21	2495.78	38245.63
	1.74	4.83	15.52	49.84	191.26	672.67	10359.63
	1.61	4.41	13.88	43.65	164.98	574.91	8908.12
0.05	3.90	12.52	43.34	169.68	671.86	2598.53	43288.03
	1.94	5.82	17.05	58.70	210.34	769.39	12072.80
	1.79	5.14	14.64	48.88	172.62	630.12	9863.98
0.08	4.21	13.95	50.94	203.20	820.12	3382.32	55885.89
	2.32	6.80	20.57	71.66	242.06	922.36	15514.49
	2.14	6.01	17.71	59.77	200.99	756.46	12397.85
0.10	4.46	15.07	58.11	235.76	980.21	3949.65	66508.00
	2.46	7.19	22.56	76.57	271.03	1051.75	16441.65
	2.27	6.49	19.83	65.10	229.43	887.51	13570.93
0.15	4.56	17.49	71.81	308.31	1289.43	5494.77	90699.35
	2.54	7.64	24.61	84.65	315.40	1182.94	18043.73
	2.42	7.20	22.82	76.97	251.73	1060.33	16122.29
0.20	4.98	18.45	84.06	371.98	1582.87	6464.70	108890.71
	2.56	7.26	23.23	82.16	394.51	1160.47	18769.55
	2.47	7.03	22.39	78.63	289.62	1099.82	17781.04

2. For the corrected and minimized corrected IITA algorithms the average *dist* and $dist^*$ values are quite similar, with a maximum discrepancy of 0.40 and 0.58, respectively. Nevertheless, the minimized corrected version is slightly better in more cases (in 30 cases for the *dist* value and in 37 cases for the $dist^*$ value).

3. Except for four settings the original IITA algorithm performs worse than the corrected and minimized corrected IITA algorithms, in terms of *dist* values. For $dist^*$ the new algorithms perform always better. Especially for the very high error rates, 0.15 and 0.20 the results of the original version are far off compared to the results obtained under the new algorithms. For instance, for $\tau = 0.20$, the mean $dist^*$ value of the original IITA algorithm is 240.67 and for the new algorithms it is 41.03 and 36.26, respectively.

4. All algorithms have in common that for increasing sample sizes and decreasing error rate the *dist* and $dist^*$ values become better. Note that the improvements with increasing sample size are larger for the new algorithms. For example, for $\tau = 0.10$, the original version gives a *dist* value of 12.89 for a sample size of 50, and 9.82 for a sample size of 6400 (with a difference of 3.07). However, the minimized corrected version gives the values of 8.71 and 4.02 (with a difference of 4.69).

The second simulation study takes into account that very large quasi orders are more seldom than medium sized quasi orders. The results of this simulation study confirm the superiority of the two new methods as compared to the original IITA algorithm. It is important to note that the original IITA algorithm produces very good results if the underlying quasi order is very large. Hence the first simulation study is strongly favoring the

original algorithm (which nevertheless is inferior to the new algorithms for higher error rates).

Note, that a perfect simulation scheme would assume randomly drawing the underlying quasi order from the set of all possible quasi orders. This set for nine items already is larger than 200 000 for unlabeled (that is using non-discriminable items) quasi orders, and larger than 44 billion for labelled quasi orders (Brinkmann and McKay, 2007). This perfect sampling scheme is not feasible in practice.

3.4.4 Applications to empirical data

In this section, we apply the three IITA algorithms to two empirical datasets. One is the Aphasic dataset, which is also used in Schrepp (2003), and the other is from the Programme for International Student Assessment (PISA; http://www.pisa.oecd.org/).

IITA analyses of the Aphasic dataset

The Aphasic dataset (Gloning et al., 1972) consists of 162 aphasic patients tested on five tasks. These tasks are:

1. point to an object on a picture (Example: Please show me the ship.)

2. name an object on a picture (Example: Please tell me how this object is calles.)

3. repeat a sentence (Example: Please repeat exactly what I say.)

4. name as fast as possible words beginning with a given letter (Example: Please tell ma as many words as possible starting with M)

5. the number of verbal and phonemic errors produced when the patient performs tasks 2, 3, and 4

The items were dichotomized at the median and coded that 1 stands for aphasic behavior and 0 for normal behavior This dataset is used in Schrepp (2003) for comparing the original IITA algorithm to feature pattern analysis and configural frequency analysis. For details on the dataset, the latter two methods, and the obtained results, see Schrepp (2003).

Analyses of the Aphasic dataset using the corrected and minimized corrected IITA algorithms give the same quasi order as obtained for the original algorithm. The quasi order consists of the following implications $\{(1,1)$, $(1,2)$, $(1,3)$, $(1,4)$, $(2,1)$, $(2,2)$, $(2,3)$, $(2,4)$, $(3,1)$, $(3,2)$, $(3,3)$, $(3,4)\}$. The three IITA versions reproduce the scaling of items obtained by feature pattern analysis and also derive all the knowledge states obtained by configural frequency analysis (Schrepp, 2003). The fact that all three IITA algorithms produce virtually the same results as obtained by these approved (for the Aphasic dataset) data analysis methods is positive and confirms their usefulness. Interestingly, though the same quasi order is obtained for the three algorithms, the computed *diff* values (i.e., the matches between the observed and expected numbers of counterexamples; the fit criterion) are considerably smaller for the corrected (61.54) and minimized corrected (60.93) versions than for the original (165.98) algorithm, showing a better fit of the b^*_{ij} to the data for the two new algorithms.

IITA analyses of the PISA dataset

We analyze part of the 2003 PISA data consisting of 340 German students answering eight questions on mathematical literacy. These items are chosen to form a Rasch scale. That is, the dichotomous one-parameter logistic model

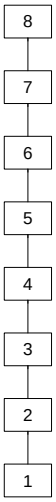

Figure 3.6: Rasch scale of the eight assessment items (from bottom to top, items sorted according to increasing difficulty). Assumed to underlay the PISA dataset.

(Fischer and Molenaar, 1995) fits (goodness-of-fit and item fit) the data very well. Under this model, the following item difficulties are estimated for the eight questions: -2.09, -1.58, -1.23, -0.04, 0.28, 0.66, 1.46, and 2.20. 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 (1944) 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, which is considered as the underlying quasi order (see Figure 3.6) in the subsequent analyses.

Analyzing the PISA dataset using the original IITA algorithm and the corrected and minimized corrected IITA algorithms gives the quasi orders shown in Figures 3.7 and 3.8, respectively.

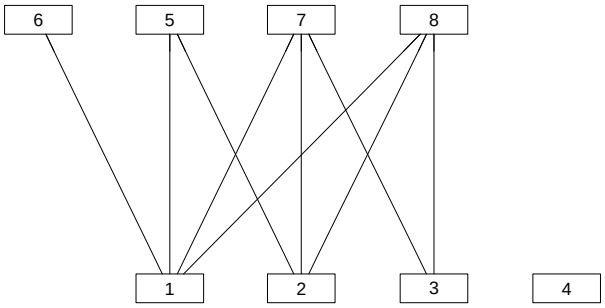

Figure 3.7: Quasi order obtained for the PISA dataset under the original IITA algorithm.

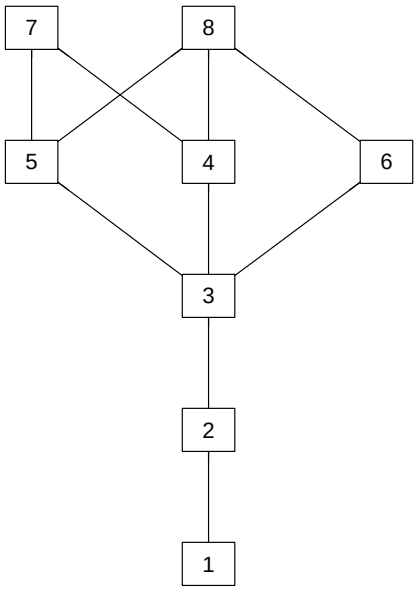

Figure 3.8: Quasi order obtained for the PISA dataset under the corrected and minimized corrected IITA algorithms.

The original IITA algorithm yields a *dist* value of 19, in contrast to the corrected and minimized corrected versions, which give a clearly smaller *dist* value of 5. Since under all three algorithms no false implications are added, these are the numbers of true implications missed by the algorithms. The corrected and minimized corrected versions outperform the original algorithm.

The better performance of the two new algorithms is even more evident, if multiple barcharts are used for exploring the data. Multiple barcharts are a variant of mosaic plots, in which each tile has the same width and the height is computed according to the number of cases in the cell. Mosaic plots are a good graphic for exploring categorical data (Unwin et al., 2006). For dichotomous data, as we have in KST, multiple barcharts provide an appropriate way of visually displaying the data (Ünlü and Sargin, 2008b). If interactive techniques are incorporated, those graphics can become a powerful tool for detecting knowledge states (for interactive graphics, see Theus and Urbanek (2008); Ünlü and Sargin (2008b)). Figure 3.9 shows the multiple barcharts view of Items $1, 3, 4, 7,$ and 8 of the PISA dataset. We used only five items for illustrating the usage of mosaic plots, because it gives a clearer picture of the benefits of using mosaic plots.

The multiple barcharts in Figure 3.9 give a satisfactory picture. The two tiles in the upper left and lower right corners of the mosaic plot correspond to the knowledge states $\emptyset$ and Q. The tiles representing the remaining states reasonably emerge, as compared to the ones that do not correspond to the states.

In Figure 3.10 the knowledge states obtained by the original (left) and corrected / minimized corrected (right) IITA algorithms are highlighted in multiple barcharts.

The original and corrected / minimized corrected IITA algorithms both

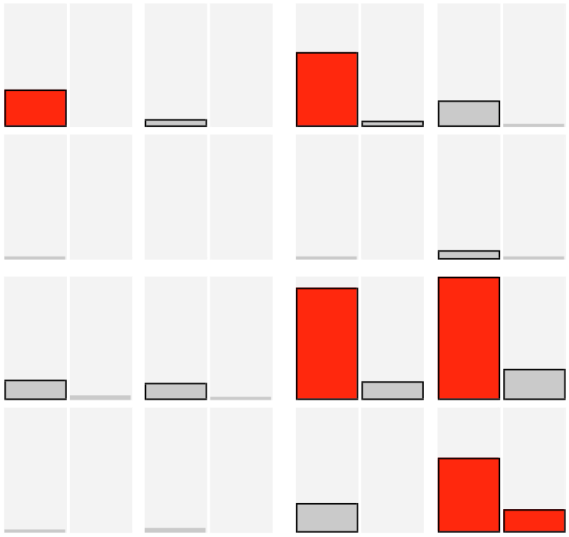

Figure 3.9: Mosaic plot of the PISA dataset. The assumed underlying knowledge states are highlighted.

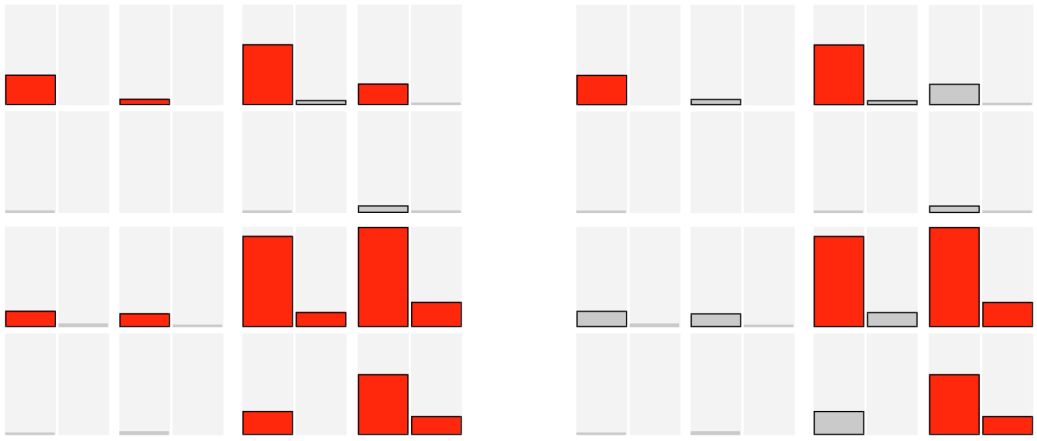

Figure 3.10: Mosaic plot of the PISA dataset. The knowledge states obtained for this dataset under the original (left) and corrected / minimized corrected (right) IITA algorithms are highlighted.

detect all the underlying knowledge states. The original IITA algorithm additionally includes seven non-states; for instance, the non-state represented by the tile in the first row and third column. Using multiple barcharts, this tile would certainly be discarded. (Note that the tiles in the upper left and lower right corners correspond to the knowledge states $\emptyset$ and Q.) The corrected / minimized corrected IITA algorithm, on the other hand, only includes one non-state. Yet the tile representing this non-state (third row, last column) has a relatively large height. However, it is obvious from the graphic that the original IITA algorithm is includes non-states in the derived quasi order.

The better performance of the two new algorithms can be explained by the fact that the original IITA version, in general, gives bad results when longer chains of items are present in the underlying quasi order. In the PISA example, the underlying Rasch scale $\sqsubseteq$, which is a chain, consists only of cases $i \sqsubseteq j$ and $j \not\sqsubseteq i$. As mentioned in Section 3.2, for these cases inconsistent estimators are used in the original algorithm. This leads to larger discrepancies between the observed and expected numbers of counterexamples, hence to a larger *diff* value. The corrected and minimized corrected IITA algorithms, however, use the corrected estimators and therefore detect true implications more properly.

3.5 Maximum likelihood methodology

In this section, we introduce the population analogs of the *diff* fit measures, interpret the coefficients as maximum likelihood estimators (MLE) for the corresponding population values, and show for these estimators the quality properties asymptotic efficiency, asymptotic normality, asymptotic unbiased-

ness, and consistency. The use of asymptotic normality in practice is further commented on in Section 3.6.

3.5.1 The *diff* coefficients as maximum likelihood estimators

Consider the transformed sample *diff* coefficients

$$diff_t := diff/m^2.$$

The division is necessary to cancel out sample size m in replacements of sample quantities with population quantities. Given the multinomial probability distribution on the set of all response patterns, make the following replacements in the arguments, b_{ij} and p_i, of the sample $diff_t$ coefficients:

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

This gives three population $diff_t$ coefficients corresponding to the sample $diff_t$ coefficients.

The sample $diff_t$ coefficients are the obvious sample analogs of these population fit measures. They are reobtained by replacing the arguments $\rho(R)$ of the population $diff_t$ measures with the maximum likelihood estimates $m(R)/m$ of the multinomial distribution. According to the invariance property of maximum likelihood estimation, the sample $diff_t$ coefficients are the maximum likelihood estimators for the corresponding population $diff_t$ coefficients. The invariance property states that if $\hat{\theta}$ is the maximum likelihood estimator for θ, then for any function $f(\theta)$, the maximum likelihood estimator for $f(\theta)$ is $f(\hat{\theta})$ (Casella and Berger, 2002; Zehna, 1966).

3.5.2 Asymptotic properties of the *diff* coefficients

Next, we present an application of established maximum likelihood asymptotics. Though this is a straightforward application, it is novel and important in the so far ad hoc discussion of data analysis methods in KST. Since the following techniques are well-known, the explanations are kept succinct. For technical details on asymptotic properties and regularity conditions, see Bishop et al. (1975), Casella and Berger (2002), and Witting and Müller-Funk (1995).

Maximum likelihood estimators possess a number of asymptotic quality properties, given certain regularity conditions are satisfied. Important properties are asymptotic efficiency (the most precise estimates are produced), and implied by this property, asymptotic normality, asymptotic unbiasedness (estimates converge in expectation to the true values), and consistency (estimates converge in probability to the true values). It can be verified that the maximum likelihood estimator for the multinomial distribution fulfills required regularity conditions and hence is asymptotically efficient (Witting and Müller-Funk, 1995).

The population diff_t coefficients are continuous functions of the multinomial cell probabilities $\rho(R)$. (Note that $\rho(R) > 0$ for all response patterns $R \in 2^Q$. This assumption is essential for assuring continuity of the population diff_t coefficients. Therefore the corresponding sample diff_t coefficients are asymptotically efficient, asymptotically normal, asymptotically unbiased, and consistent estimators for the population values (Casella and Berger, 2002).

3.5.3 Illustrating consistency

One possibility to assess and compare the quality of asymptotic properties for finite samples for the three IITA algorithms is by simulation. We exemplify that with the consistency property. First, we visually illustrate consistency using one quasi order. Theoretically, consistency is formulated and holds for any single quasi order. The rate of convergence may vary from quasi order to quasi order. Second, to get a rough structure-independent evaluation, we aggregate the results obtained for 100 quasi orders.

The simulation study illustrating consistency is based on nine items and is as follows. This simulation study is not to be mixed up with the simulation studies for comparing the three IITA approaches discussed earlier.

1. All reflexive pairs are always added to the relation $\mathcal{R}$. A constant δ is set randomly (Sargin and Ünlü (2009a)), which gives the probability for adding each of the remaining 72 item pairs to the relation. The transitive closure $\sqsubseteq$ of this relation $\mathcal{R}$ is computed, and is the underlying (true) quasi order.

2. Fifty data matrices are simulated for each of the increasing sample sizes 100, 1000, 10000, and 25000 in the following way. From the set $\{K \in 2^Q : (i \sqsubseteq j \wedge j \in K) \rightarrow i \in K\}$ of all response patterns consistent with $\sqsubseteq$ an element is drawn randomly. For this drawn pattern all entries are changed from 1 to 0 or from 0 to 1, with a same prespecified error probability τ. This is simulating with a special case of the BLIM.

3. Under all three algorithms, the sample and population $diff_t$ coefficients are computed.

In Figure 3.11 a graphical display of consistency for one quasi order is given (for $\tau = 0.10$); running the previous three simulation steps once.

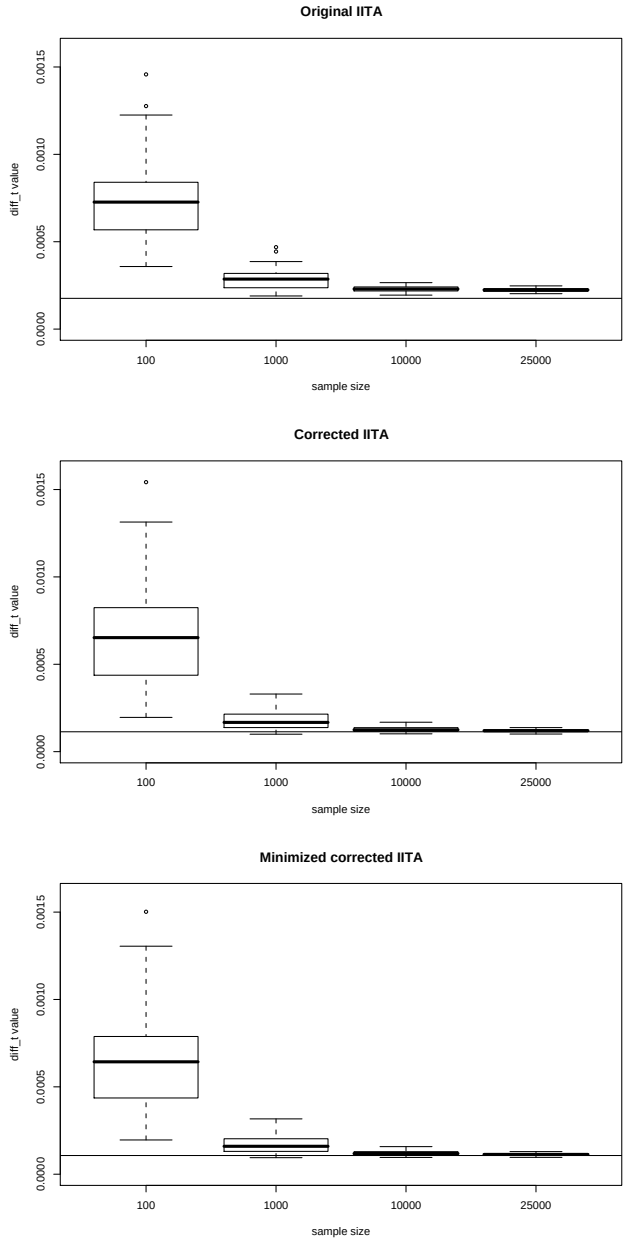

Figure 3.11: Boxplots for the three IITA algorithms, within each of the sample sizes of the 50 computed sample $diff_t$ values. The three population $diff_t$ values are shown as horizontal lines in the plots.

Figure 3.11 shows boxplots for all three IITA algorithms, within each of the sample sizes of the 50 computed sample $diff_t$ values. The three population $diff_t$ values are shown as horizontal lines in the plots. This graphic illustrates that the population values are better attained and the sample values are less dispersed with increasing sample size, for all three algorithms. The results are better for the corrected and minimized corrected IITA versions than for the original. The corrected and minimized corrected algorithms have a higher speed of convergence. In particular, they achieve the population values with a much higher accuracy than the original algorithm, which shows, even for a sample size of 25000, clear discrepancies between sample and population values. Hence consistency, which is guaranteed by theory, manifests in smaller finite sample sizes for the two new IITA versions.

Table 3.7 summarizes the aggregated results obtained for 100 quasi orders (for $\tau = 0.10$); running the three simulation steps 100 times.

Table 3.7 shows, for each combination of ϵ (0.01, 0.001, 0.0001) and sample size, the relative frequencies of 5000 data matrices satisfying $|\hat{\theta}_n - \theta| > \epsilon$, where $\hat{\theta}_n$ and θ stand for the sample and population $diff_t$ coefficients, respectively. The entries represent estimates of the probabilities $\mathbb{P}(|\hat{\theta}_n - \theta| > \epsilon)$ used in the definition of consistency, where the probability is taken with respect to the true multinomial distribution. For instance, the first entry says that the probability for obtaining a sample $diff_t$ value, for a sample size of 100, differing more than 0.01 from the population $diff_t$ value is, approximately, 0.0010. This is on average, independent of the underlying quasi order.

Under all three algorithms, for each ϵ, the relative frequencies are decreasing with increasing sample size (except for one case, mentioned below). Again, the two new IITA versions outperform the original. The original version shows the lowest speed of convergence, and for $\epsilon = 0.0001$, from sample

Table 3.7: Relative frequencies of 5000 data matrices (50 data matrices per one out of 100 quasi orders) satisfying $|\hat{\theta}_n - \theta| > \epsilon$; first, second, and third lines refer to the original, corrected, and minimized corrected IITA algorithms, respectively.

	Sample size			
	100	1000	10000	25000
ϵ				
0.01	0.0010	0	0	0
	0	0	0	0
	0	0	0	0
0.001	0.2402	0.0278	0	0
	0.0466	0	0	0
	0.0326	0	0	0
0.0001	0.9266	0.5636	0.4910	0.5240
	0.9540	0.2306	0.0158	0.0066
	0.9646	0.1878	0.0032	0.0002

sizes 10000 to 25000, the relative frequency is even increasing. The corrected and minimized corrected IITA algorithms perform well and quite similar, with a slight advantage for the minimized corrected.

In sum, we have seen that the *diff* coefficients of the IITA algorithms can be interpreted as maximum likelihood estimators possessing desirable asymptotic properties. Based on the consistency property, next we propose evaluating the *diff* fit measures via rank ordered population values.

3.5.4 Comparisons of the population values of the three algorithms

Prior, only sample, not population, quantities have been considered. The simulation study in this section is theoretical, in the sense of solely dealing with values for a known population. The following summary statistics (evaluation criteria) are investigated in population, not sample, quantities.

The symmetric difference, at the level of items (*dist*), of the obtained and underlying quasi orders is used as a distance measure. Since there is a bijection between quasi orders and their corresponding knowledge structures, the symmetric difference can also be considered at the level of knowledge states ($dist^{\star}$). The results obtained at the two levels may differ; for example, the original IITA algorithm may have moderately lowest *dist* but considerably highest $dist^{\star}$ values (see Table 3.8). Therefore we introduce the rank statistic (*rk*) as a third useful measure. Given a set of competing quasi orders, which is required to include the underlying one, this statistic computes the rank of the true quasi order in the ordered list of population $diff_t$ values.

This population based approach is justified according to the asymptotic theory discussed previously. The sample $diff_t$ values converge in probability (and expectation) to the population $diff_t$ values.

3.5.5 Procedure of the simulation study

In the simulation study nine items are used. The general simulation scheme consists of five parts. First, the underlying quasi order is generated randomly. Second, the set of competing quasi orders is constructed according to the inductive procedure of the IITA algorithms. Third, the underlying quasi order is added to the selection set. Fourth, the population $diff_t$ coefficients

are computed. Fifth, the three algorithms are compared regarding symmetric differences and ranks. More precisely:

1. All reflexive pairs are always added to the relation $\mathcal{R}$. A constant δ is set randomly (Sargin and Ünlü (2009a)), which gives the probability for adding each of the remaining 72 item pairs to the relation. The transitive closure $\sqsubseteq$ of this relation $\mathcal{R}$ is computed, and is the underlying quasi order.

2. To generate a selection set of quasi orders, a binary 5000×9 data matrix is simulated. From the set $\{K \in 2^Q : (i \sqsubseteq j \wedge j \in K) \rightarrow i \in K\}$ of all response patterns consistent with $\sqsubseteq$ an element is drawn randomly. For this drawn pattern all entries are changed from 1 to 0 or from 0 to 1, with a same prespecified error probability τ. The inductive construction procedure is applied to the simulated data matrix.[1]

3. If the underlying quasi order $\sqsubseteq$ is not contained in the selection set, it is added.

4. Under all three algorithms, the population $diff_t$ coefficients are computed for all quasi orders of the selection set.

5. The three algorithms are compared with respect to three criteria: the symmetric differences *dist* and $dist^\star$ of the obtained (with smallest population $diff_t$ value) and underlying quasi orders and corresponding knowledge structures, respectively, and the rank *rk* of the underlying quasi order among the population $diff_t$ values.

[1]The idea is to obtain a large as possible number of quasi orders in the selection set. Experimentation (not reported here) has shown that for sample sizes greater than 5000 barely any improvement of the selection set is achieved. Sample sizes smaller than 5000 have led to smaller selection sets.

The error probabilities take the values 0.03, 0.05, 0.08, 0.10, 0.15, and 0.20. For each of these error probabilities, the previous five simulation steps are run 1000 times.

3.5.6 Results of the simulation study

For each of the three algorithms, for every error probability, three population summary statistics are computed. They are the means of the 1000 *dist*, $dist^{\star}$, and *rk* values. These summary statistics are reported in Table 3.8).

Table 3.8) shows the following results:

1. Summary statistic *dist*: For the small error rates 0.03 and 0.05, the original algorithm gives better average *dist* results than the corrected and minimized corrected. For all other τ values, the two new versions perform clearly better than the original. This is especially the case for the large error probabilities 0.15 and 0.20.

   The average population *dist* values show a similar pattern as the average sample *dist* values reported in Sargin and Ünlü (2009a). Those descriptive results hence are substantiated through theoretical considerations. In both simulation studies, the two new versions outperform the original, yet the difference in performance is larger in terms of population quantities.

   For any τ value, the minimized corrected IITA algorithm performs slightly better than the corrected. This shows that the minimized corrected version is better asymptotically.

2. Summary statistic $dist^{\star}$: For all error probabilities, the average $dist^{\star}$ statistic gives the same ranking; listed from worst to best, original, corrected, and minimized corrected IITA. The results are quite similar

Table 3.8: Average *dist*, *dist**, and *rk* values; first, second, and third lines refer to the original, corrected, and minimized corrected IITA algorithms, respectively.

	Summary statistic		
	dist	*dist**	*rk*
τ			
0.03	0.74	2.42	1.78
	3.10	1.72	1.60
	2.99	0.77	1.43
0.05	1.16	11.73	2.30
	2.76	2.23	1.68
	2.31	0.91	1.35
0.08	4.05	40.85	3.88
	3.72	2.17	1.95
	3.50	1.13	1.57
0.10	6.17	79.44	6.54
	3.59	2.89	2.35
	3.00	1.65	1.67
0.15	15.11	142.90	11.76
	3.62	6.56	3.18
	3.49	3.54	2.42
0.20	32.79	174.80	16.96
	4.56	14.76	4.79
	3.82	10.81	3.86

for the corrected and minimized corrected algorithms. Compared to the original version, the corrected and minimized corrected IITA algorithms perform very well. For error probabilities up to 0.10, their average $dist^\star$ values are smaller than 3. The original version, however, shows a bad performance already for $\tau = 0.05$. The results strongly worsen, reaching a maximum average $dist^\star$ value of 174.80 for $\tau = 0.20$. For the corrected and minimized corrected versions, the corresponding average $dist^\star$ values are 14.76 and 10.81, respectively.

3. Summary statistic *rk*: For all error probabilities, the average *rk* statistic gives the same ranking; listed from worst to best, original, corrected, and minimized corrected IITA. The corrected and minimized corrected IITA algorithms perform quite similar. Compared to the original version, they produce good results, especially for larger error rates. For $\tau = 0.20$, the corrected and minimized corrected versions give average *rk* values of 4.79 and 3.86, respectively, while the original algorithm shows a considerably larger average *rk* value of 16.96.

Some remarks are in order regarding the results of the simulation study.

1. Overall, the minimized corrected version performs best, second comes the corrected, and worst is the original (with respect to all three summary statistics). We have obtained similar results for the two new algorithms. For each of the three summary statistics, the original version has shown considerably bad results for larger error probabilities.

2. Further analyses made using ranks (of underlying quasi orders) show that the original version, compared to the other two algorithms, not only performs worse based on average ranks, but also has higher maximum ranks. For every error probability, the maximum of the 1000 *rk*

values is greater. For instance, we obtained the maximum ranks 22, 7, 6 (for $\tau = 0.03$) and 40, 31, 15 (for $\tau = 0.10$) for the original, corrected, and minimized corrected algorithms. Moreover, the original version is outperformed concerning the number of *rk* values that are at most as large as 3 (first three ranks). For instance, we obtained the first three ranks 893, 940, 960 times (for $\tau = 0.03$) and 645, 830, 919 times (for $\tau = 0.10$) for the original, corrected, and minimized corrected algorithms. These summary statistics measure rank variability and show that the original IITA algorithm has a wider range for the *rk* values.

3. That the original algorithm gives better average *dist* results in population quantities for the error probabilities 0.03 and 0.05 can be explained in the same way as we did for sample quantities in Sargin and Ünlü (2009a). The incorrect estimation scheme of the original algorithm produces good results specifically when the size of the underlying quasi order is large. For a large quasi order $\sqsubseteq$, there are predominantly the cases $i \sqsubseteq j$, for which correct estimators are used. For the cases $i \not\sqsubseteq j$, however, incorrect estimators are applied, and the discrepancies between the observed and expected numbers of counterexamples are large. This implies that, for an underlying large quasi order, the diff_t values for small quasi orders of the selection set are large (pulling apart the diff_t value for the true quasi order from the diff_t values obtained for the other relations). As a result, the underlying quasi order is more frequently recovered. This is true particularly for smaller error probabilities. In addition, note that in the case of a large number of implications in the underlying quasi order, there are large differences of the sizes of the true and the neighboring relations in the selection set (due to transitivity). For instance, for nine items used in the simula-

tion study, an underlying quasi order consisting of 64 implications has possible nearest neighbors which contain 58 or 72 implications, and the former even may not be included in the selection set. As a consequence, for an underlying large quasi order, missing the true relation already implies a large *dist* value.

3.6 Inferential statistics for the *diff* coefficients

So far we could only tell which quasi order fits the data best. However it is important to know to which degree one quasi order is better than another. Furthermore the *diff* coefficient was treated as a single number. However, for an estimator it is important to know its variability. To tackle these problems, hypothesis testing and computation of confidence intervals are the proper tools in statistics. These tools require the variances of the *diff* coefficients to be computed.

Maximum likelihood estimators satisfy several asymptotic properties, if certain regularity conditions are fulfilled. Assume a sequence of estimators $W_n = W_n(X_1, ..., X_n)$ with $\mathbb{E}(X_i) = \mu$, then one of these properties is asymptotic normality

$$\sqrt{n}\frac{W_n - \mu}{\sigma} \to Z,$$

where $Z \sim N(0, 1)$ (Casella and Berger, 2002). Further, the delta method (Goodman and Kruskal, 1979) states that, for any function $f(\theta)$ satisfying the property that $f'(\theta)$ exists and is non-zero valued,

$$\sqrt{n}\frac{f(W_n) - f(\mu)}{\sigma[f'(\theta)]} \to Z.$$

For computing σ, one can use the inverse of the Fisher information matrix

$$I(\theta) = \mathbb{E}\left[\frac{\partial}{\partial\theta}\log(L(\theta))^2|\theta\right],$$

where $L(\theta)$ is the likelihood function. The variance can be computed by

$$Var(f(\hat{\theta})|\theta) = f'(\theta)I^{-1}(\theta)[f'(\theta)]^T,$$

if $I(\theta)$ is nonsingular (Casella and Berger, 2002).

3.6.1 Gradients of the *diff* coefficients

In the following, we derive the gradients of the three $diff_t$ coefficients.

First, we present the gradients of the corrected and minimized corrected IITA $diff_t$ coefficients:

$$\begin{aligned}
diff_t &= \frac{diff}{m^2} \\
&= \frac{\sum_{i\neq j}\frac{(b_{ij}-b^*_{ij})^2}{n(n-1)}}{m^2} \\
&= \frac{1}{n(n-1)}\left(\sum_{\substack{i\neq j\\ i\sqsubset j}}\left(\frac{b_{ij}}{m}-p_j\gamma\right)^2+\sum_{\substack{i\neq j\\ i\not\sqsubset j, j\sqsubset i}}\left(\frac{b_{ij}}{m}-(p_j-p_i+p_i\gamma)\right)^2\right) \\
&+ \frac{1}{n(n-1)}\sum_{\substack{i\neq j\\ i\not\sqsubset j, j\not\sqsubset i}}\left(\frac{b_{ij}}{m}-(1-p_i)p_j\right)^2 \\
&= \frac{1}{n(n-1)}\left(\sum_{\substack{i\neq j\\ i\sqsubset j}}\underbrace{\left(\varrho_{\bar{i}j}-\varrho_j\gamma\right)^2}_{=:\mu_1}+\sum_{\substack{i\neq j\\ i\not\sqsubset j, j\sqsubset i}}\underbrace{\left(\varrho_{\bar{i}j}-(\varrho_j-\varrho_i+\varrho_i\gamma)\right)^2}_{=:\mu_2}\right) \\
&+ \frac{1}{n(n-1)}\sum_{\substack{i\neq j\\ i\not\sqsubset j, j\not\sqsubset i}}\underbrace{\left(\varrho_{\bar{i}j}-(1-\varrho_i)\varrho_j\right)^2}_{=:\mu_3}.
\end{aligned}$$

Next, $\frac{\partial}{\partial\rho(\hat{R})}$ is computed for a fixed $\hat{R}\in 2^Q$. Since the coefficient is a sum, we can derive each part separately.

According to the chain rule the derivatives of μ are

$$\frac{\partial \mu_k}{\partial \rho(\hat{R})} = 2\sqrt{\mu_k}\frac{\partial \sqrt{\mu_k}}{\partial \rho(\hat{R})}, k = 1, 2, 3.$$

The corresponding derivatives are as follows:

$$\frac{\partial \sqrt{\mu_1}}{\partial \rho(\hat{R})} = \begin{cases} 1 - \gamma - \varrho_j \frac{\partial \gamma}{\partial \rho(\hat{R})} & : \quad \hat{R} = R_{\bar{i}j} \\ -(\gamma + \varrho_j \frac{\partial \gamma}{\partial \rho(\hat{R})}) & : \quad \hat{R} = R_j \wedge \hat{R} \neq R_{\bar{i}j} \\ -\varrho_j \frac{\partial \gamma}{\partial \rho(\hat{R})} & : \quad \hat{R} = R_{\bar{j}} \wedge \hat{R} \neq R_{\bar{i}j} \end{cases}$$

$$\frac{\partial \sqrt{\mu_2}}{\partial \rho(\hat{R})} = \begin{cases} -\varrho_i \frac{\partial \gamma}{\partial \rho(\hat{R})} & : \quad \hat{R} = R_{\bar{i}j} \\ -(\gamma + \varrho_i \frac{\partial \gamma}{\partial \rho(\hat{R})}) & : \quad \hat{R} = R_{ij} \\ 1 - (\gamma + \varrho_i \frac{\partial \gamma}{\partial \rho(\hat{R})}) & : \quad \hat{R} = R_{i\bar{j}} \\ -\varrho_i \frac{\partial \gamma}{\partial \rho(\hat{R})} & : \quad \hat{R} = R_{\bar{i}\bar{j}} \end{cases}$$

$$\frac{\partial \sqrt{\mu_3}}{\partial \rho(\hat{R})} = \begin{cases} \varrho_i & : \quad \hat{R} = R_{\bar{i}j} \\ \varrho_j - (1 - \varrho_i) & : \quad \hat{R} = R_{ij} \\ \varrho_j & : \quad \hat{R} = R_{i\bar{j}} \\ 0 & : \quad \hat{R} = R_{\bar{i}\bar{j}} \end{cases}$$

The derivative of γ for the corrected version is

$$\frac{\partial \gamma}{\partial \rho(\hat{R})} = \begin{cases} \frac{1}{|\sqsubset| - n} \sum\limits_{\substack{i \neq j \\ i \sqsubset j}} \frac{\varrho_j - \varrho_{\bar{i}j}}{\varrho_j^2} & : \quad \hat{R} = R_{\bar{i}j} \\ \frac{1}{|\sqsubset| - n} \sum\limits_{\substack{i \neq j \\ i \sqsubset j}} \frac{\varrho_{\bar{i}j}}{\varrho_j^2} & : \quad \hat{R} \neq R_{\bar{i}j} \wedge \hat{R} = R_j \\ 0 & : \quad \text{else} \end{cases} .$$

For the minimized corrected version the derivative is as follows. (Recall that $\gamma = -\frac{x_1 + x_2}{x_3 + x_4}$ where $x_1, x_2, x_3,$ and x_4 are defined as in Section 3.3.2.) The derivative of the error rate γ is

$$\frac{\partial \gamma}{\partial \rho(\hat{R})} = -\frac{\frac{\partial (x_1 + x_2)}{\partial \rho(\hat{R})}(x_3 + x_4) - \frac{\partial (x_3 + x_4)}{\partial \rho(\hat{R})}(x_1 + x_2)}{(x_3 + x_4)^2}.$$

Since $\frac{\partial(x_1+x_2)}{\partial\rho(\hat{R})} = \frac{\partial(x_1)}{\partial\rho(\hat{R})} + \frac{\partial(x_2)}{\partial\rho(\hat{R})}$ and $\frac{\partial(x_3+x_3)}{\partial\rho(\hat{R})} = \frac{\partial(x_3)}{\partial\rho(\hat{R})} + \frac{\partial(x_4)}{\partial\rho(\hat{R})}$, the derivatives are:

$$\frac{\partial(x_1)}{\partial\rho(\hat{R})} + \frac{\partial(x_2)}{\partial\rho(\hat{R})} = \begin{cases} -2\sum\limits_{\substack{i\neq j\\ i\sqsubseteq j}}(\varrho_j + \varrho_{\bar{i}j}) & : \quad \hat{R} = R_{\bar{i}j} \\ \sum\limits_{\substack{i\neq j\\ i\not\sqsubseteq j}}(-2\varrho_{\bar{i}j} + 2\varrho_j - 2\varrho_i) - 2(\sum\limits_{\substack{i\neq j\\ i\sqsubseteq j}}\varrho_{\bar{i}j}) & : \quad \hat{R} = R_{ij} \\ \sum\limits_{\substack{i\neq j\\ i\not\sqsubseteq j}}(-2\varrho_{\bar{i}j} + 2\varrho_j - 4\varrho_i) & : \quad \hat{R} = R_{i\bar{j}} \\ 0 & : \quad \hat{R} = R_{\bar{i}\bar{j}} \end{cases}$$

and

$$\frac{\partial(x_3)}{\partial\rho(\hat{R})} + \frac{\partial(x_4)}{\partial\rho(\hat{R})} = \begin{cases} 4\sum\limits_{\substack{i\neq j\\ i\sqsubseteq j}}\varrho_j & : \quad \hat{R} = R_{\bar{i}j} \\ 4(\sum\limits_{\substack{i\neq j\\ j\sqsubseteq i, i\not\sqsubseteq j}}\varrho_i + \sum\limits_{\substack{i\neq j\\ i\sqsubseteq j}}\varrho_j) & : \quad \hat{R} = R_{ij} \\ 4(\sum\limits_{\substack{i\neq j\\ j\sqsubseteq i, i\not\sqsubseteq j}}\varrho_i) & : \quad \hat{R} = R_{i\bar{j}} \\ 0 & : \quad \hat{R} = R_{\bar{i}\bar{j}} \end{cases}$$

For the original IITA the gradient follows. The derivative of γ is the same as for the corrected IITA version. The $diff_t$ coefficient for the original IITA algorithm is

$$\begin{aligned} diff_t &= \frac{1}{n(n-1)}\left(\sum_{\substack{i\neq j\\ i\sqsubseteq j}}\left(\frac{b_{ij}}{m} - p_j\gamma\right)^2 + \sum_{\substack{i\neq j\\ i\not\sqsubseteq j}}\left(\frac{b_{ij}}{m} - (1-p_i)p_j(1-\gamma)\right)^2\right) \\ &= \frac{1}{n(n-1)}\left(\sum_{\substack{i\neq j\\ i\sqsubseteq j}}\underbrace{\left(\varrho_{\bar{i}j} - \varrho_j\gamma\right)^2}_{:=\nu_1} + \sum_{\substack{i\neq j\\ i\not\sqsubseteq j}}\underbrace{\left(\varrho_{\bar{i}j} - (1-\varrho_i)\varrho_j(1-\gamma)\right)^2}_{=:\nu_2}\right). \end{aligned}$$

As above

$$\frac{\partial\nu_k}{\partial\rho(\hat{R})} = 2\sqrt{\nu_k}\frac{\partial\sqrt{\nu_k}}{\partial\rho(\hat{R})}, k = 1, 2,$$

where the corresponding derivatives are as follows:

$$\frac{\partial\sqrt{\nu_1}}{\partial\rho(\hat{R})} = \begin{cases} 1-\gamma-\varrho_j\frac{\partial\gamma}{\partial\rho(\hat{R})} & : \quad \hat{R} = R_{\bar{i}j} \\ -(\gamma+\varrho_j\frac{\partial\gamma}{\partial\rho(\hat{R})}) & : \quad \hat{R} = R_j \wedge \hat{R} \neq R_{\bar{i}j} \\ -\varrho_j\frac{\partial\gamma}{\partial\rho(\hat{R})} & : \quad \hat{R} = R_{\bar{j}} \wedge \hat{R} \neq R_{\bar{i}j} \end{cases}$$

$$\frac{\partial\sqrt{\nu_2}}{\partial\rho(\hat{R})} = \begin{cases} \gamma - \varrho_j \frac{\partial(1-\gamma)}{\partial\rho(\hat{R})} + \varrho_i \left(1 - \gamma + \varrho_j \frac{\partial(1-\gamma)}{\partial\rho(\hat{R})}\right) : \hat{R} = R_{\bar{i}j} \\ \gamma - 1 - \varrho_j \left(\frac{\partial(1-\gamma)}{\partial\rho(\hat{R})} - 1 + \gamma\right) + \varrho_i \left(1 - \gamma + \varrho_j \frac{\partial(1-\gamma)}{\partial\rho(\hat{R})}\right) : \hat{R} = R_{ij} \\ -\varrho_j \left(\frac{\partial(1-\gamma)}{\partial\rho(\hat{R})} - 1 + \gamma\right) + \varrho_i \left(\varrho_j \frac{\partial(1-\gamma)}{\partial\rho(\hat{R})}\right) : \hat{R} = R_{i\bar{j}} \\ -\varrho_j \frac{\partial(1-\gamma)}{\partial\rho(\hat{R})} + \varrho_i \left(\varrho_j \frac{\partial(1-\gamma)}{\partial\rho(\hat{R})}\right) : \hat{R} = R_{\bar{i}\bar{j}} \end{cases}$$

3.6.2 Expected Fisher information matrix

In Section 2.2 we showed that a multinomial distribution underlies the data. In general, we consider now a multinomial distribution $M(m, p)$ with m trials and cell probabilities $p = (p_1, \ldots, p_n)$, with $p_i > 0$ for all $i = 1, \ldots, n$. Let $X = (X_1, \ldots, X_n) \sim M(m, p)$ with realization $x = (m_1, \ldots, m_n)$. The likelihood function is

$$L = \frac{m!}{\prod_{i=1}^{n} m_i!} \prod_{i=1}^{n} {p_i}^{m_i}.$$

Thus

$$\log(L) = C + \left(m - \sum_{i=2}^{n} m_i\right) \log\left(1 - \sum_{i=2}^{n} p_i\right) + \sum_{i=2}^{n} m_i \log(p_i),$$

where C does not depend on p.

The Hessian matrix $I = \left(\partial^2 \log L / \partial p_i \partial p_j\right)_{i,j}$ (of dimension $(n-1)\times(n-1)$) is next computed. Consider an p_k (for some $k \neq 1$). Then

$$\frac{\partial}{\partial p_k} \log L = \frac{m_k}{p_k} - \frac{m - \sum_{i=2}^{n} m_i}{1 - \sum_{i=2}^{n} p_i}.$$

On the diagonal of I, for $k = 2, \ldots, n$,

$$\frac{\partial^2}{\partial^2 p_k} \log L = -\frac{m_k}{p_k^2} - \frac{m - \sum_{i=2}^{n} m_i}{(1 - \sum_{i=2}^{n} p_i)^2}.$$

Off-diagonal elements of I, for $k, l = 2, \ldots, n$, $k \neq l$,

$$\frac{\partial^2}{\partial p_k \partial p_l} \log L = -\frac{m - \sum_{i=2}^{n} m_i}{(1 - \sum_{i=2}^{n} p_i)^2}.$$

In particular, the off-diagonal elements of I are identical. Set $\theta := -\frac{m-\sum_{i=2}^{n} m_i}{(1-\sum_{i=2}^{n} p_i)^2}$, and

$$I = \begin{pmatrix} -\frac{m_2}{p_2^2} + \theta & \theta & \theta & \cdots & \theta \\ \theta & -\frac{m_3}{p_3^2} + \theta & \theta & \cdots & \theta \\ & & \ddots & & \\ \theta & \theta & \theta & \cdots & -\frac{m_n}{p_n^2} + \theta \end{pmatrix}.$$

The expected Fisher information matrix is $-E_p(I)$. Off-diagonal elements of $-E_p(I)$, for $k, l = 2, \ldots, n$, $k \neq l$ are

$$-E_p(I_{kl}) = \frac{m}{1 - \sum_{i=2}^{n} p_i}.$$

On the diagonal of $-E_p(I)$, for $k = 2, \ldots, n$ we have

$$-E_p(I_{kk}) = \frac{m}{p_k} + \frac{m}{1 - \sum_{i=2}^{n} p_i}.$$

Set $\theta' := \frac{m}{1-\sum_{i=2}^{n} p_i}$, and

$$-E_p(I) = \begin{pmatrix} \frac{m}{p_2} + \theta' & \theta' & \theta' & \cdots & \theta' \\ \theta' & \frac{m}{p_3} + \theta' & \theta' & \cdots & \theta' \\ & & \ddots & & \\ \theta' & \theta' & \theta' & \cdots & \frac{m}{p_n} + \theta' \end{pmatrix}.$$

It can be seen that the inverse of $-\frac{1}{m}E_p(I)$ is the variance-covariance matrix $(\delta_{ij}p_i - p_i p_j)_{i,j}$ for $i, j = 2, \ldots, n$ (Cramér, 1946). Here δ_{ij} is the Kronecker delta.

In Ünlü and Sargin (2009) it is shown that the observed and expected Fisher information matrices are equal, if maximum likelihood estimators are used. Hence the inverses are the same. Figure 3.12 summarizes and illustrates this results.

Figure 3.12 is to be understood as follows. The expected Fisher information (EF) and observed Fisher information (OF) matrices can be inverted in order to obtain EF^{-1} and OF^{-1} and vice verse.

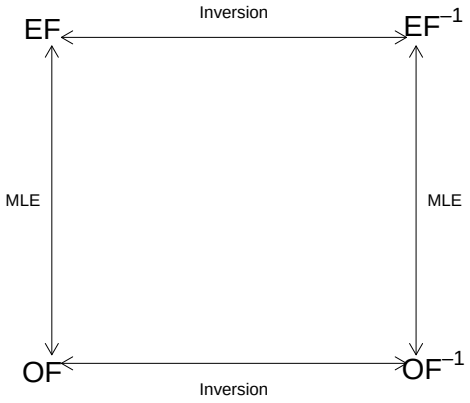

Figure 3.12: Diagram of relations between expected and observed Fisher information matrices. The diagram shows that one can either invert the Fisher information matrix and then use the MLE, or first use the MLE and then invert the matrix.

3.6.3 Applications to empirical and simulated data

In this section, we illustrate the use and performance of the above approach in finite sample sizes by real and simulated data. We start with giving an example using the PISA data (cf. Section 3.4.5). In the following we only focus on the minimized corrected IITA algorithm. For the other IITA versions the approach is analogous.

In Section 3.4.5 we obtained a quasi order with smallest *diff* value for the minimized corrected IITA version. It has the implications $\{(1,1),(1,2),(1,3),(1,4),(1,5),(2,2),(2,3),(2,4),\ (2,5),(3,3),(3,4),(3,5),(4,4),(5,5)\}$, where (i,j) stands for solving item j implies solving item i. This relation is depicted in Figure 3.8. Further, the PISA test items form a Rasch scale (see

Figure 3.6) with significant goodness-of-fit and item fit tests, hence the Rasch model suggests a chain as the underlying quasi order. With the upper approach we can statistically analyze if a significant difference between the *diff* values exists.

The transformed $diff_t$ value of the quasi order implied by the minimized corrected IITA algorithm is 0.0002384614 with variance 0.0000037372, and of the chain 0.0009289828 with variance 0.0000278196. We perform a normal hypothesis test for comparing the means of the $diff_t$ values under the null hypothesis that the chain has a larger $diff_t$ value than the quasi order obtained by minimized corrected IITA algorithm. We obtain a p-value of 0.0117, which indicates a significant difference of these two results. However, one should note that the result is not highly significant. Therefore the chain as the underlying quasi order (as proposed by the Rasch model) can be questioned and for further analyses the quasi order obtain by the minimized corrected IITA algorithm should rather be used.

This example shows the possibilities gained through the availability of variances. First, it is now possible to judge whether the quasi orders in the selection set have a significant difference to the quasi order that fits the data best. This information can be taken into consideration in further analyses. Second, a common approach in KST for deriving a quasi order is the querying of experts. Different quasi orders, obtained by querying different experts, can be tested against each other. Third, many methods exist in psychometrics for deriving quasi orders by data analysis. The results gained by these data analysis methods can be compared with the results of other data analysis methods. This is what was done in the upper example, where the quasi order proposed by the Rasch model was tested against the solution of the minimized corrected IITA algorithm.

Next, we illustrate the performance of the asymptotic behavior by simulated data examples. For the fixed quasi order, $\{(1,1),(1,8),(2,2),(2,3),(2,4),(2,5),(2,8),(3,3),(3,4),(3,5),(4,4),(4,5),(5,5),(6,6),(7,6),(7,7),(8,8),(9,3),(9,4),(9,5),(9,9)\}$ displayed in Figure 3.13, 200 datasets are simulated, 50 for each of the four sample sizes, $100, 500, 5000$, and 100000. In all cases the sample variances are computed using the minimized corrected IITA version, and a boxplot is drawn for each of the sample sizes. The population asymptotic variance is shown as a horizontal line in the graphic.

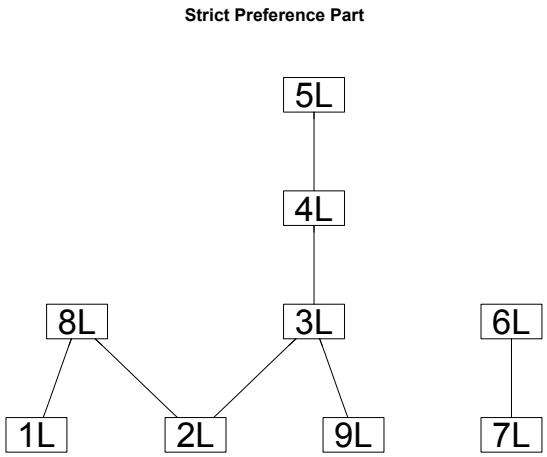

Figure 3.13: Underlying fixed quasi order used for simulating the data.

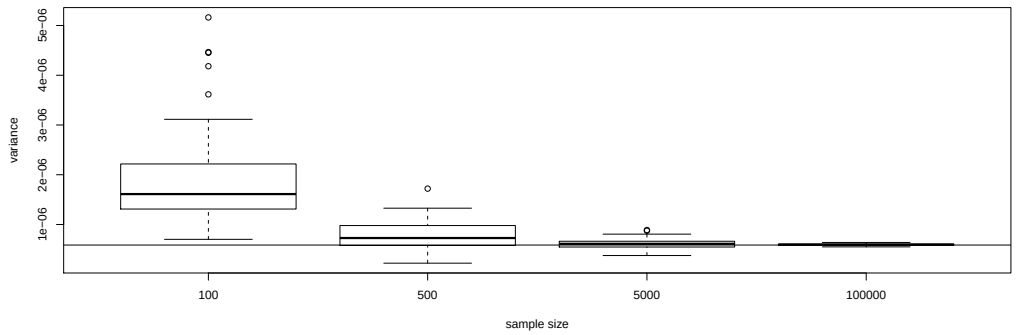

Figure 3.14: Boxplots of the sample variance computed for a fixed quasi order under the minimized corrected IITA version. For each of the sample sizes 50, datasets are simulated and the sample variances are computed. The corresponding population value is shown as a horizontal line in the plot.

Figure 3.14 shows that the sample variances approach the population value with increasing sample size. Further, the values are less dispersed and fewer outliers are produced. This indicates that the asymptotic behavior of the the computation of the sample variance is, already for a sample size of 500, very good.

Chapter 4

DAKS - Data analysis and knowledge spaces in R

Due to the large amount of computational effort, which is needed in KST and for the IITA algorithms, it is indispensable to use computer software. Currently available software implementing the original IITA algorithm is ITA 2.0 by Schrepp (2006). Compared to this stand-alone software that runs on Windows only, the package DAKS by Sargin and Ünlü (2008, 2009c) is implemented in the comprehensive R computing environment and provides much more functionalities, such as more flexible input/output features.

R (R: Development Core Team, 2009, http://www.r-project.org/) is a language and environment for statistical computing and graphics. It gives users the possibility to include own software packages for handling specific tasks. Besides the three IITA algorithms, the package DAKS implements functions for computing population and estimated asymptotic variances of the fit measures, and for switching between test item and knowledge state representations. Other features are a Hasse diagram drawing device, a data simulation tool, a function for computing response pattern and knowledge

state frequencies, and a Z-test for comparing *diff* values of quasi orders.

In this chapter, we give an overview of the package `DAKS` and illustrate its usage with the PISA dataset (see Sections 3.4.5 and 3.6.3).

4.1 Description of the package `DAKS`

In this section, we present the functions implemented in the package `DAKS` and discuss their functionalities. Table 4.1 summarizes all functions of the package `DAKS`.

4.1.1 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 $j \rightarrow i$. This is implemented in `DAKS` using the package `sets` (Meyer and Hornik, 2009). The latter, in combination with the package `relations` (Hornik and Meyer, 2009), 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 $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$. Note that in the second code line an item i is represented by iL. This transformation takes place internally in the packages

Table 4.1: Summary of the `DAKS` functions

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 simulation tool
`state2imp`	Transforming from knowledge states to implications
`variance`	Computing estimated asymptotic variances
`z_test`	Z-test for *diff* values

`sets` or `relations`, but it does not have any influence. Both representations are equal:

```
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
```

4.1.2 Functions of the package `DAKS`

The 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)
```

The first function 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`. The second function 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.

A function to compute the absolute frequencies of the occurring response patterns, and optionally, the absolute frequencies of a collection of knowledge states in a dataset 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 to 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).

A data simulation tool based on the BLIM 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. (The general form of the BLIM allows for varying careless error and lucky guess rates from item to item, 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 is used to derive the corresponding quasi ordinal knowledge space. The latter is equipped with the error probabilities `ce` and `lg` to give the BLIM that is used for simulating the data. 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 the randomly generated quasi order, because the transitive closure is formed. In Sargin and Ünlü (2009b), 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 (see Section 3.4.1 for details). In Sargin and Ünlü (2009a) a second sampling is proposed, which puts more weight on sampling medium sized quasi orders

than very large or very small ones (see Section 3.4.4 for details). (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.)

Another basic function of the package `DAKS` is a Hasse diagram drawing device:

```
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` from `Bioconductor` (`http://www.bioconductor.org/`), which is 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.

Two auxiliary functions for implementing the IITA algorithms are:

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

The first function computes from a `dataset` for all item pairs the corresponding numbers of observed counterexamples. These values are crucial in the formulations of the IITA algorithms. This function returns a matrix of the

numbers of observed counterexamples for all pairs of items. The second function can be used to generate inductively, from a matrix `b` of the numbers of observed counterexamples, a set of quasi orders. The inductive generation of the selection set of competing quasi orders is a prime component of the IITA algorithms. This function returns a list of the inductively generated surmise relations. The main function `iita` (see below) calls `ob_counter` for computation of the numbers of counterexamples, and `ind_gen` for the inductive generation procedure.

Three functions of the package `DAKS` realizing the original, corrected, and minimized corrected IITA algorithms 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. The function `iita` (see below) automatically generates a selection set from the data using the inductive generation procedure implemented in `ind_gen` (see above). The latter 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 the 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. All three functions return vectors of the *diff* values and error rates corresponding to the competing quasi orders in `A`.

The function that can be used to perform one of the three IITA procedures selectively is:

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

Whereas for the above three functions selection sets of competing quasi orders have to be passed via an argument manually, this function automatically generates a selection set from the `dataset` using the inductive generation procedure implemented in `ind_gen` (see above). The parameter `v` specifies the IITA algorithm to be performed; `v = 1` (minimized corrected), `v = 2` (corrected), and `v = 3` (original). Compared to the above three functions, this function 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 and its index in the selection set. (In case of ties in minimum *diff* value, a quasi order with smallest size is returned.)

The package `DAKS` also contains functions which provide the basis for statistical inference methodology. 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). 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.

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`. 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 `A` is specified the passed set of competing quasi orders is used for computing population values. 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 inductively generated selection set, and the used IITA version.

The function for computing population (exact) asymptotic variances of the MLEs *diff* 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 here is formulated for the relation and error rate specified in `imp` and `error_pop`, respectively. This population variance is obtained using the delta method (see Section 3.6), which requires calculating the Jacobian matrix of the *diff* coefficient and the inverse of the expected Fisher information matrix for the multinomial distribution. 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*.

The function for computing estimated asymptotic variances of the MLEs *diff* 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 here 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`). In the expression for the exact asymptotic variance (expressed in Jacobian matrix and inverse expected Fisher information), the true parameter vector of the multinomial probabilities is estimated 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 using 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*.

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 can be performed. The quasi orders are specified by `imp` for a one sample test, and additionally by `imp_alt` for a two sample test. The value which the test is based on is given by `mu`, and the alternative hypothesis is specified by `alternative`. For a one sample test, `conf.level` gives the confidence interval for the *diff value*, for a two sample test, the confidence interval for the difference of the two *diff* values is computed. The function `z_test` returns the Z- and p-values, the type and values of the confidence interval, the *diff* values of the specified quasi orders, the specified alternative, and the assumed true value.

4.2 Illustration

We illustrate usage of the package `DAKS` with another part of the 2003 PISA data. The dataset consists of the item responses by 317 German students on a 12-item dichotomously scored mathematical literacy test. (Note that this dataset is different from the `pisa` dataset accompanying the package `DAKS` and used in Sections 3.4.5 and 3.6.3.

An overview of the data is given by tables of all variables and the function `pattern`.

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

  Item.5 Item.6 Item.7 Item.8 Item.37 Item.38 Item.39 Item.64
0    121    245    112    212     129     272     202     112
1    196     72    205    105     188      45     115     205
  Item.67 Item.72 Item.73 Item.75
0     154      87     247      22
1     163     230      70     295
```

```
R> pattern(pisa)

5 largest response patterns in the data:
000000000001 000000000101 101010111101 101110111101 111110111111
          12            8            6            6            6
```

We see, for instance, that the last item (Item.75) is most frequently and the sixth item (Item.38) least frequently solved. The patterns occurring most frequently are the ones where the last item is solved, especially those where only the last or only the last and the tenth item are solved. This shows that the last item is very easy compared to the other items. The last pattern shown above is the one where only the sixth item is not solved, indicating a high difficulty of this item. To analyze the dependencies between the items more accurately, we perform analyses based on the IITA algorithms.

```
R> mini<-iita(pisa, v = 1)
R> mini

   Inductive Item Tree Analysis

Algorithm: minimized corrected IITA
diff values: 257.238 257.117 253.152 239.742 221.593 220.916
216.392 216.724 209.855 204.699 205.684 205.413 202.327 201.471
202.859 201.8 207.349 205.754 199.303 193.052 187.615 174.749
168.928 161.145 153.269 161.902 172.85 179.379 160.938 155.544
145.234 159.922 173.313 167.625 162.24 166.072 170.558 215.021
220.063 231.783 236.943 257.827 1683.669
quasi order:{(1L, 2L), (1L, 4L), (1L, 6L), (1L, 7L), (1L, 11L),
(2L, 6L), (3L, 2L), (3L, 4L), (3L, 5L), (3L, 6L), (3L, 7L),
```

```
(3L,9L), (3L, 11L), (4L, 2L), (4L, 6L), (4L, 11L), (5L, 2L),
(5L, 4L), (5L, 6L), (5L, 7L), (5L, 9L), (5L, 11L), (7L, 6L),
(7L, 11L), (8L, 2L), (8L, 4L), (8L, 6L), (8L, 7L), (8L, 9L),
(8L, 11L), (9L, 2L), (9L, 4L), (9L, 6L), (9L,11L), (10L, 2L),
(10L, 3L), (10L, 4L), (10L, 5L), (10L,6L), (10L, 7L),
(10L, 8L), (10L, 9L), (10L,11L), (11L,6L), (12L, 1L),
(12L, 2L), (12L, 3L), (12L, 4L), (12L, 5L), (12L, 6L),
(12L, 7L), (12L, 8L), (12L, 9L), (12L, 10L), (12L, 11L)}
error rate: 0.143
index in the selection set: 31
```

The *diff* values for all quasi orders of the selection set are computed. The quasi order with minimum *diff* value is shown, and the corresponding error rate and its index in the selection set are output.

The results of the other two algorithms are computed analogously.

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

For analyzing the results, functions of `R` or of other packages can be helpful. For example, the symmetric difference between two quasi orders can easily be computed, showing the implications in which the two quasi orders differ.

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

{(1L, 2L), (1L, 4L), (1L, 7L), (1L, 11L), (2L, 6L), (3L,5L),
(3L, 7L), (3L, 9L), (4L, 2L), (4L, 6L), (4L, 11L),(5L, 2L),
(5L, 4L), (5L, 7L), (5L, 9L), (7L, 6L), (7L,11L), (8L, 2L),
(8L, 4L), (8L, 7L), (8L, 9L), (9L, 2L), (9L, 4L), (9L, 6L),
```

```
(9L, 11L), (10L, 3L), (10L, 5L), (10L, 7L), (10L, 8L),
(10L, 9L), (11L, 6L), (12L, 1L), (12L, 3L),(12L, 10L)}
```

As the quasi orders of the selection set are nested, this symmetric difference shows the implications which are contained in minimized corrected IITA solution, but not in the solution obtained under the original IITA version. These additional implications give a more refined structure of the dependencies between the items. This can also be seen from the Hasse diagrams in Figures 4.1 and 4.2.

```
R> hasse(mini$implications, 12)

list()
```

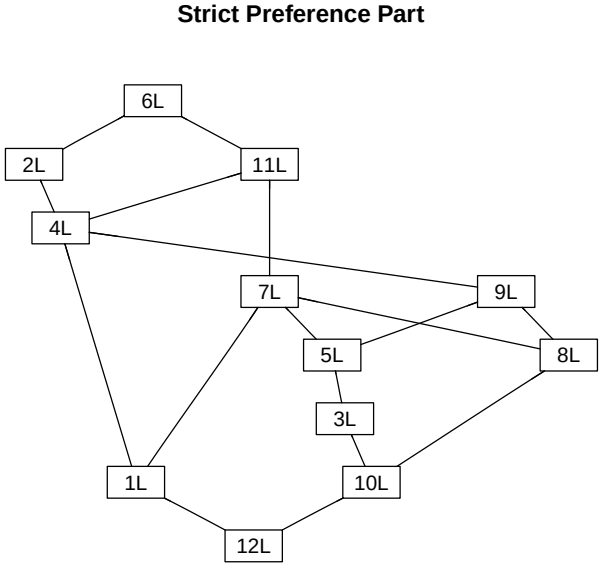

Figure 4.1: Hasse diagram of the quasi order obtained for the PISA dataset with twelve items under the minimized corrected IITA algorithm.

```
R> hasse(orig$implications, 12)

list()
```

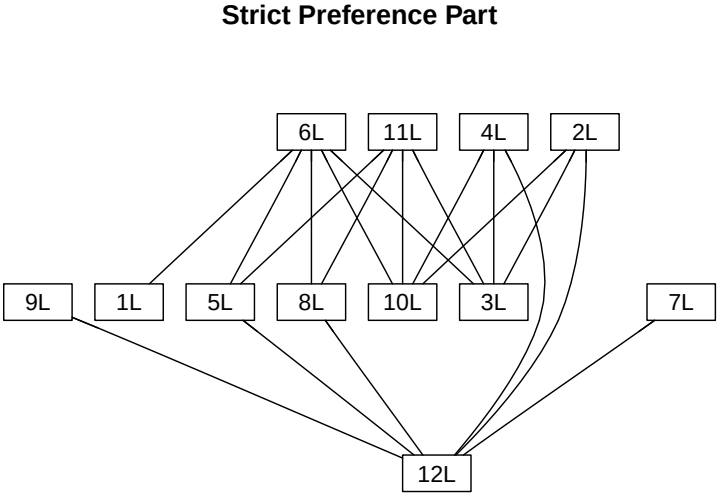

Figure 4.2: Hasse diagram of the quasi order obtained for the PISA dataset with twelve items under the original IITA algorithm.

The empty lists show that there are no parallel items in the obtained quasi orders. In Figure 4.1, a well-structured quasi order is displayed, with item 12 (Item.75) being the easiest and item 6 (Item.38) the most difficult one. On the other hand, Figure 4.2 has a more simple structure, mainly consisting of three layers. This structure barely reveals the overall structure; for instance, item 6 is not the unique most difficult item and item 12 is not implied by all other items. This, again (see Section 3.5.6), can be explained by the fact that the likely underlying structure contains long chains (the longest chain in the quasi order obtained by the minimized corrected algorithm consists of eight items: $6 \rightarrow 2 \rightarrow 4 \rightarrow 9 \rightarrow 5 \rightarrow 3 \rightarrow 10 \rightarrow 12$).

Next, the states obtained by the algorithms are analyzed. Using the function `pattern` we see how often the states occur in the dataset.

```
R> patmini<-pattern(pisa, P = imp2state(mini$implications, 12))
R> patcorr<-pattern(pisa, P = imp2state(corr$implications, 12))
R> patorig<-pattern(pisa, P = imp2state(orig$implications, 12))
```

The frequencies of the states can be used for further investigations. For example, it is interesting to compute the proportion of obtained states that do not occur in the dataset:

```
R> sum(patmini$states[,13] == 0)/nrow(patmini$states)

[1] 0.1153846

R> sum(patcorr$states[,13] == 0)/nrow(patcorr$states)

[1] 0.1333333

R> sum(patorig$states[,13] == 0)/nrow(patorig$states)

[1] 0.5608108
```

The results clearly show that the original IITA algorithm identifies far more states that cannot be observed in the dataset. This indicates that too many states (hence too few implications) are contained in the obtained quasi order. The ratio of non-observable and observable states is much better for the minimized corrected and corrected IITA versions.

To gain certainty we perform some hypothesis testing. In the following, we present three Z-tests for comparing the *diff* values for the obtained quasi orders. First, we compare the minimized corrected and corrected IITA algorithm results based on the estimates calculated under both versions.

```
R> z_test(pisa, mini$implications, corr$implications, v = 1)

 Two sample Z-test

z =  -0.3918  p-value =  0.6952
alternative hypothesis: true mean is not equal 0
95 percent confidence interval:
  -0.0004798994 0.0003199932
sample estimates:
    mean in imp mean in imp_alt
        0.00145         0.00153

R> z_test(pisa, mini$implications, corr$implications, v = 2)

 Two sample Z-test

z =  0.1205  p-value =  0.9041
alternative hypothesis: true mean is not equal 0
95 percent confidence interval:
  -0.0008655576 0.000978937
sample estimates:
    mean in imp mean in imp_alt
        0.00177         0.00171
```

Both p-values are high, hence it cannot be assumed that the *diff* values differ significantly. The two quasi orders differ only in eight implications (computed by `length(set_symdiff(mini$implications,corr$implications))`), which explains the similar results.

However, the *diff* value of the original IITA algorithm is significantly different from the *diff* value obtained for the minimized corrected version, at the significance level of 0.01:

```
R> z_test(pisa, mini$implications, orig$implications, v = 1)

   Two sample Z-test


z =  -2.579  p-value =  0.0099
alternative hypothesis: true mean is not equal 0
95 percent confidence interval:
  -0.001058706 -0.0001443989
sample estimates:
    mean in imp mean in imp_alt
        0.00145         0.00205
```

4.3 Summary

We have performed a first analysis of this part of the PISA data. We have derived possible quasi orders giving us the potential implications between the test items, and have compared the obtained results. We have illustrated the features of the package `DAKS`, and we have indicated the advantages of using `R` by performing further analyses based on the results gained through the functions in the package `DAKS`.

In the future the package will have to be enhanced, for example it is planned to implement other fit measures such as the *di* (discrepancy) index (Kambouri et al., 1994) or the *CA* (correlational agreement) coefficient (van Leeuwe, 1974). As the package will be extended, users will be offered a powerful and free software tool for handling tasks in KST.

By contributing the `R` package `DAKS` a basis for computational work in the so far combinatorial theory of knowledge spaces is established. Implementing KST procedures in `R` can help to bring together KST and such other psychometric approaches as item response theory (IRT). A number of `R` packages are available for IRT; for instance, `eRm`, `ltm`, or `mokken`. KST and IRT are split directions of psychological test theories and are currently compared at a theoretical level (Stefanutti, 2006; Stefanutti and Robusto, 2009; Ünlü, 2007). Using `R` as an interface between these theories may prove valuable in comparing them at a computational level.

Chapter 5

Discussion

5.1 Summary

Data analysis methods in KST are becoming more and more important. With advancing computing power and more sophisticated algorithms, it is now possible to tackle the combinatorial and statistical problems involved with these data analysis methods.

In Chapter 2, the necessary basic concepts of KST were introduced. In Chapter 3, inductive item tree analysis was discussed. In Section 3.2, the original IITA algorithm and its problems were presented. In Section 3.3, two new algorithms, minimized corrected and corrected IITA, were established and thorough comparisons of the original and new methods were given (Section 3.4). We have shown that the *diff* fit measures can be interpreted as maximum likelihood estimators (Section 3.5), which possess a number of good asymptotic properties. Based on these properties, techniques for inferential statistics were presented in Section 3.6. To perform all computations the `R` package `DAKS` was developed. This package was presented in Chapter 4 and used for the computations and analyses in this work.

To sum up: Two new algorithms have been proposed, which are superior to the present original IITA algorithm. Tools from statistics were used to introduce well-established techniques for inference (e.g. MLE, hypothesis testing). So far, only ad hoc quality properties have been considered (Schrepp, 2007), without taking advantage of such techniques as previously discussedd.

Finally, some perspectives on future research and open problems are discussed.

5.2 Directions for future research

IITA is still a very young method for deriving quasi orders from data. Hence, enhancements and modifications are possible and can be pursued.

Work on the generated selection set should definitely be pursued in future research. So far, for the IITA algorithms, the quality of the inductively generated set of quasi orders has not been systematically investigated. In our simulation study, on average (see Table 3.4 for individual figures), the underlying quasi orders are contained 569 (out of 1000) times in the selection sets. Since it is computationally intractable to evaluate all possible quasi orders in large-scale applications, better search methods may be needed to improve the selection set. A data analysis method operating on a set of candidate models is only as good as the quality of the selection set is.

An interesting direction for further research is to modify the *diff* coefficient. As apparent from the presented simulation study (see Section 3.4.2 and Section 3.4.3), smaller *diff* values do not necessarily imply better reconstructions of underlying quasi orders. It seems that an aggregation (*diff* coefficient) of local, two-dimensional views of the data (b_{ij}) does not pro-

vide acceptable results on the relationships among all items mutually in $|Q|$ dimensions. One could consider developing fit measures incorporating higher-dimensional views of the data.

The fit measures around in KST (e.g. *di* or *CA*), whether they are formulated at the level of items or at the level of knowledge states, all aggregate the manifest multinomial cell counts into a single real number. This is why, uniformly, they can be based theoretically using the maximum likelihood approach (see Section 3.5). However, it is important to note that, in practice, the quality of the asymptotics has to be checked for finite sample sizes. For example, this can be pursued by graphical approaches.

Incorporating latent parameters into the formulations of the *diff* coefficients (or of other fit measures) is important. The manifest γ_{L} parameter in *diff* is used as an estimate of the latent response error probability. Instead, the expected numbers of counterexamples could be parameterized directly in terms of latent (e.g., careless error and lucky guess) parameters. Though the introduction of latencies may complicate theory and computation, it can provide more realistic and interpretable results.

In its current forms, IITA works only for dichotomous items. In questionnaires or aptitude tests, for example, it is common to have polytomous items. An important direction for future research is to enhance IITA to polytomous, continuous or mixed indicators. This would provide a powerful tool for deriving knowledge structures.

Bibliography

Airasian, P. and Bart, W. (1973). Ordering theory: A new and useful measurement model. *Educational Technology*, 13:56–60.

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

Bart, W. and Krus, D. (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.

Bishop, Y., Fienberg, S., and Holland, P. (1975). *Discrete Multivariate Analysis: Theory and Practice*. M.I.T. Press, Cambridge, MA.

Brinkmann, G. and McKay, B. (2007). Posets on up to 16 points. *http://cs.anu.edu.au/~bdm/papers/posets.pdf*.

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

Chen, C., Härdle, W., and Unwin, A., editors (2008). *Handbook of Data Visualization*. Springer-Verlag, Berlin.

Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press, Princeton, New Jersey.

Doignon, J.-P. and Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, 23:175–196.

Doignon, J.-P. and Falmagne, J.-C. (1987). Knowledge assessment: A set theoretical framework. In *Beiträge zur Begriffsanalyse: Vorträge der Arbeitstagung Begriffsanalyse, Darmstadt, 1986*, pages 129–140, Mannheim, Germany. B.I. Wissenschaftsverlag.

Doignon, J.-P. and Falmagne, J.-C. (1999). *Knowledge Spaces*. Springer-Verlag, Berlin.

Falmagne, J.-C. (1989). Probabilistic knowledge spaces: A review. In Roberts, F., editor, *Applications of Combinatorics and Graph Theory to the Biological and Social Sciences*, volume 17, pages 283–303. Springer-Verlag, New York.

Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., and Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test and search them. *Psychological Review*, 97:201–224.

Fischer, G. and Molenaar, I., editors (1995). *Rasch Models: Foundations, Recent Developments, and Applications*. Springer-Verlag, New York.

Gloning, J., Lienert, G., and Quatamber, R. (1972). Konfigurationsfrequenzanalyse aphasiespezifischer Testleistungen. *Zeitschrift für Klinische Psychologie und Psychotherapie*, 20:115–122.

Goodman, L. and Kruskal, W. (1979). *Measures of Association for Cross Classification*. Springer-Verlag, New York.

Guttman, L. (1944). A basis for scaling qualitative data. *American Sociological Review*, 9:139,150.

Hornik, K. and Meyer, D. (2009). *relations: Data Structures and Algorithms for Relations*. R package version 0.5-2.

Kambouri, M., Koppen, M., Villano, M., and Falmagne, J.-C. (1994). Knowledge assessment: Tapping human expertise by the query routine. *International Journal of Human-Computer Studies*, 40:119–151.

Meyer, D. and Hornik, K. (2009). Generalized and customizable sets in R. *Journal of Statistical Software*, 31(2):1–27.

R: Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R: Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Sargin, A. and Ünlü, A. (2008). DAKS: An R package for data analysis methods in knowledge space theory. *Manuscript under revision*.

Sargin, A. and Ünlü, A. (2009a). Inductive item tree analysis: Corrections, improvements, and comparisons. *Mathematical Social Sciences*, 58(3):376–392.

Sargin, A. and Ünlü, A. (2009b). Simulation schemes for quasi orders: A comment to Sargin and Ünlü (2009a). *Manuscript under preparation*.

Sargin, A. and Ünlü, A. (2009c). *DAKS: Data Analysis and Knowledge Spaces*. R package version 1.0-0.

Schrepp, M. (1999). On the empirical construction of implications between bi-valued test items. *Mathematical Social Sciences*, 38:361–375.

Schrepp, M. (2002). Explorative analysis of empirical data by boolean analysis of questionnaires. *Zeitschrift für Psychologie*, 210:99–109.

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. (2006). `ITA 2.0`: A program for classical and inductive item tree analysis. *Journal of Statistical Software*, 16.

Schrepp, M. (2007). On the evaluation of fit measures for quasi-orders. *Mathematical Social Sciences*, 53:196–208.

Stefanutti, L. (2006). A logistic approach to knowledge structures. *Journal of Mathematical Psychology*, 50:545–561.

Stefanutti, L. and Robusto, E. (2009). Recovering a probabilistic knowledge structure by constraining its parameter space. *Psychometrika*, 74:83–96.

Theus, M. and Urbanek, S. (2008). *Interactive Graphics for Data Analysis*. CRC Press, London.

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. and Sargin, A. (2008a). Maximum likelihood methodology for diff fit measures for quasi orders. *Manuscript submitted for publication*.

Ünlü, A. and Sargin, A. (2008b). Mosaics for visualizing knowledge structures. *Manuscript under revision*.

Ünlü, A. and Sargin, A. (2009). Asymptotic variances for the inductive item tree analysis algorithms. *Manuscript in preparation*.

Unwin, A., Theus, M., and Hofmann, H. (2006). *Graphics of Large Datasets*. Springer-Verlag, New York.

van der Linden, W. and Glas, C., editors (2000). *Computerized adaptive testing: Theory and practice*. Kluwer, Norwell, MA.

van Leeuwe, J. (1974). Item tree analysis. *Nederlands Tijdschrift voor de Psychologie*, 29:475–484.

Witting, H. and Müller-Funk, U. (1995). *Mathematische Statistik II*. Teubner-Verlag, Stuttgart, Germany.

Zehna, P. (1966). Invariance of maximum likelihood estimators. *The Annals of Mathematical Statistics*, 37:744.

Lebenslauf

Persönliches

Name:	Anatol Sargin
Geboren:	10. November 1980
Staatsangehörigkeit:	deutsch

Ausbildung

05/2000	High School Diploma, Aucilla Christian Academy, Florida/USA
06/2002	Allgemeine Hochschulreife, Dürer-Gymnasium Nürnberg
04/2003 - 09/2007	Studium der Wirtschaftsmathematik, Universität Augsburg, Diplom 09/2007

Berufstätigkeit

06/2002 - 03/2003	Zivildienst, Rotes Kreuz, Nürnberg
seit 10/2007	Wissenschaftliche Hilfskraft, Lehrstuhl für Rechnerorientierte Statistik und Datenanalyse, Universität Augsburg