Journal of Statistical Software

September 2006, Volume 16, Issue 10. *http://www.jstatsoft.org/*

ITA 2.0: A Program for Classical and Inductive Item Tree Analysis

Martin Schrepp

Abstract

Item Tree Analysis (ITA) is an explorative method of data analysis which can be used to establish a hierarchical structure on a set of dichotomous items from a questionnaire or test. There are currently two different algorithms available to perform an ITA. We describe a computer program called **ITA** 2.0 which implements both of these algorithms. In addition we show with a concrete data set how the program can be used for the analysis of questionnaire data.

Keywords: item tree analysis, exploratory data analysis, boolean analysis.

1. Introduction

Item tree analysis (ITA) is a data analytical method which allows constructing a hierarchical structure on the items of a questionnaire or test from observed response patterns. Assume that we have a questionnaire I with m items and that subjects can answer positive (1) or negative (0) to each of these items, i.e. the items are dichotomous. If n subjects answer the items in I this results in a binary data matrix D with m columns and n rows.

Typical examples of this data format are test items which can be solved (1) or failed (0) by subjects. Other typical examples are questionnaires where the items are statements to which subjects can agree (1) or disagree (0).

Depending on the content of the items it is possible that the response of a subject to an item j determines her or his responses to other items. It is, for example, possible that each subject who agrees to item j will also agree to item i. In this case we say that *item j implies item i* and write shortly $i \leq j$. The goal of an ITA is to uncover such deterministic implications from the data set D.

ITA was originally developed by Van Leeuwe (1974). The result of his algorithm, which we refer in the following as *Classical ITA*, is a logically consistent set of implications $i \leq j$. Logically consistent means that $i \leq j$ and $j \leq k$ implies $i \leq k$ for each triple i,j,k of items, i.e.

the relation $\leq$ is transitive. Thus the outcome of an ITA is a reflexive and transitive relation on the item set, i.e. a quasi-order on the items.

Recently Schrepp (1999) developed a different algorithm to perform an ITA, which we refer in the following as *Inductive ITA*. Classical ITA and inductive ITA both construct a quasi-order on the item set by explorative data analysis. But both methods use a different algorithm to construct this quasi-order. For a given data set the resulting quasi-orders from classical and inductive ITA will usually differ.

ITA belongs to a group of data analysis methods called *Boolean analysis of questionnaires.* Boolean analysis was introduced by Flament (1976). The goal of a Boolean analysis is to detect deterministic dependencies (formulas from Boolean logic connecting the items, like for example $i \rightarrow j$, $i \wedge j \rightarrow k$, and $i \vee j \rightarrow k$) between the items of a questionnaire or test.

Since the basic work of Flament (1976) a number of different methods for boolean analysis have been developed. See, for example, van Van Buggenhaut and Degreef (1987), Duquenne (1987), Theuns (1994) or Theuns (1998).

These methods share the goal to derive deterministic dependencies between the items of a questionnaire from data, but differ in the algorithms to reach this goal. A comparison of ITA to other methods of boolean data analysis can be found in Schrepp (2003).

Boolean analysis is closely related to the GUHA method (Hájek, Havel, and Chytil (1966) or Hájek and Havránek (1977)). The basic idea of this method is to use formal logic to generate all hypotheses which are of interest in a given research task and supported by the data. Statistical methods are used to evaluate these generated hypotheses. There is also a close connection of boolean analysis to data mining techniques, especially the extraction of association rules from data. See, for example, Klemettinen, Mannila, Ronkainen, Toivonen, and Verkamo (1994) or Toivonen (1996).

There is a close connection between item tree analysis and *knowledge space theory.* The theory of knowledge spaces (Doignon and Falmagne (1985), Doignon and Falmagne (1998) or Albert and Lukas (1999)) provides a theoretical framework for the formal description of human knowledge. A knowledge domain is in this approach represented by a set I of problems. The knowledge of a subject in the domain is then described by the subset of problems from I he or she is able to solve. This set is called the *knowledge state* of the subject. Because of dependencies between the items (for example, if solving problem j implies solving problem i) not all elements of the power set of I will, in general, be possible knowledge states. The set of all possible knowledge states is called the *knowledge structure.* Obviously, item tree analysis can be used to construct a knowledge structure from data. See, for example, Schrepp (1999).

The investigation of deterministic dependencies has some tradition in educational psychology. The items represent in this area usually skills or cognitive abilities of subjects. Bart and Airasian (1974) use ITA to establish implications on a set of Piagetian tasks. Other examples in this tradition are the learning hierarchies of Gagné (1968) or the theory of structural learning of Scandura (1991).

A recent application of classical ITA on educational test items can be found in Held and Korossy (1998) who extracted logical implications on a set of mathematical problems from observed response patterns. The extracted implications are then compared to implications obtained by querying an expert.

Another example for the use of deterministic dependencies in psychology are approaches to formalize the diagnostic process of psychologists. The goal of this approach is to uncover the rules on which the decisions of diagnosticians are based. See, for example, Härtner, Mattes, and Wottawa (1980) or Wottawa and Echterhoff (1982).

An example for the application of ITA to sociological data is Bart and Krus (1973) who used an ITA related procedure to establish a hierarchical order on items that describe socially unaccepted behavior. Janssens (1999) used a method of boolean analysis to investigate the integration process of minorities into the value system of the dominant culture. Applications of inductive ITA on the analysis of questionnaire data can be found in Schrepp (2002) or Schrepp (2003). In these papers subsets of questions from the *International Social Science Survey Program* (ISSSP) are analyzed by inductive ITA.

2. Algorithms of classical and inductive ITA

The main purpose of this paper is to describe the program **ITA** 2.0. But for an understanding of the program it is important to give a rough description of the implemented algorithms. For a more detailed introduction into this algorithms please refer for classical ITA to Van Leeuwe (1974) and for inductive ITA to Schrepp (2003).

Let us first define some basic notational conventions which are necessary for both algorithms:

- I is a set of m dichotomous items.
- $\leq$ is a quasi-order (reflexive and transitive relation) on I. Such a quasi-order on I can be represented as a set $\{(i,j) \mid i \leq j\}$ of item pairs.
- $D = \{d_1, ..., d_n\}$ is a set of n observed response patterns to the items in I. Each d_s is a mapping $d_s : I \rightarrow \{0,1\}$ which assigns to each item i the response $d_s(i) \in \{0,1\}$ of subject s.
- p_i is the relative frequency of subjects who answer positive (1) to item i, i.e.

$$p_i = \mid \{d_s \in D \mid d_s(i) = 1\} \mid /n$$

- b_{ij} is the number of response patterns which violate the dependency $i \leq j$, i.e.

$$b_{ij} = \mid \{d_s \in D \mid d_s(i) = 0 \wedge d_s(j) = 1\} \mid$$

- For a quasi-order $\leq$ on I the set of all consistent patterns $\mathrm{Cons}(\leq)$ is given by

$$\mathrm{Cons}(\leq) = \{r : I \rightarrow \{0,1\} \mid \forall i,j \in I (i \leq j \wedge r(j) = 1 \Rightarrow r(i) = 1)\}$$

  Thus, $\mathrm{Cons}(\leq)$ contains all 0-1-patterns of length m which are consistent with all the dependencies $i \leq j$ in the quasi-order $\leq$.

- Define for $d \in D$, $k \in \mathrm{Cons}(\leq)$ the distance

$$\mathrm{dist}(d,k) = \mid \{i \in I \mid d(i) \neq k(i)\} \mid$$

  between d and k. Let

$$\mathrm{mdist}(d, \mathrm{Cons}(\leq)) = \min\{\mathrm{dist}(d,k) \mid k \in \mathrm{Cons}(\leq)\}$$

be the minimal distance of d to a consistent 0-1-pattern. Then the reproducibility coefficient $\text{Repro}(\leq, D)$ is defined by:

$$\text{Repro}(\leq, D) = 1 - \frac{\sum_{d \in D} \text{mdist}(d, \text{Cons}(\leq))}{m\ n}$$

The reproducibility coefficient can be interpreted as the percentage of cells in the data matrix which can be reproduced by the quasi-order $\leq$.

2.1. The algorithm of classical ITA

To describe the algorithm of classical ITA we need in addition to the definitions given above the following conventions:

- r_{ij} is the Pearson-Correlation of items i and j.
- $C(\leq, D)$ is the number of response patterns which are not consistent with $\leq$, i.e.

$$C(\leq, D) = \mid \{d_s \in D \mid d_s \notin \text{Cons}(\leq)\} \mid$$

- The *partial order reproducibility coefficient* $\text{REP-PO}(\leq, D)$ is defined by

$$\text{REP-PO}(\leq, D) = 1 - (C(\leq, D)/n)$$

- The expected correlation r^*_{ij} under the assumption that $\leq$ is the correct quasi-order underlying the data (see Van Leeuwe (1974) for a description on how r^*_{ij} is derived from this assumption) is defined by:

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

- The *correlational agreement coefficient* $\text{CA}(\leq, D)$ is now defined by:

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

The algorithm of ITA consists accordingly to Van Leeuwe (1974) of the following 5 steps:

1. A limit τ for $\text{REP-PO}(\leq, D)$ is defined. τ is interpreted as the lowest acceptable $\text{REP-PO}(\leq, D)$ value for the best-fitting quasi-order $\leq$ resulting from the analysis.
2. A set $\text{PQO}(D) = \{\leq_L \mid L = 1, \ldots, n\}$ of relations on I is constructed where $\leq_L$ is defined by $i \leq_L j \Leftrightarrow b_{ij} \leq L$ for all $i, j \in I$. Van Leeuwe (1974) showed that $\leq_0$ is transitive but that this must not be the case for relations $\leq_L$ with $L > 0$.
3. All relations $\leq_L$ with $\text{REP-PO}(\leq_L, D) < \tau$ are eliminated from $\text{PQO}(D)$.

4. All intransitive relations $\leq_L$ are eliminated from PQO(D).
5. For all remaining relations in PQO(D) the CA($\leq_L, D$)-value is computed. Note that PQO(D) can not be empty, since $\leq_0$ is transitive and we have REP-PO($\leq_0, D) = 1$. The relation with the highest CA($\leq_L, D$)-value is the best-fitting quasi-order accordingly to classical ITA.

Currently there is a debate concerning the use of CA($\leq_L, D$) as a valid selection criterion between different quasi-orders. Ünlü and Albert (2004) list a number of properties of the coefficient CA($\leq_L, D$) which show in their opinion that CA($\leq_L, D$) should not be used for that purpose.

In a direct reply to this paper Schrepp (2006) showed that the critique raised by Ünlü and Albert (2004) is not justified. In fact this paper shows that some of the problems of CA($\leq_L, D$) described by Ünlü and Albert (2004) are properties which a good measure of fit for a quasi-order should have.

Classical ITA can be generalized to the case that some subjects have not answered all items (missing data). This is done simply by ignoring such rows in the calculation of the values r_{ij} and r^*_{ij}.

2.2. The algorithm of inductive ITA

Inductive ITA is a two step procedure. In the first step a set of quasi-orders is constructed inductively on I. In the second step a best-fitting quasi-order is chosen from this set.

Step 1: Inductive construction of a set of quasi-orders

We start with the quasi-order $\leq_0$ defined by $i \leq_0 j \Leftrightarrow b_{ij} = 0$ for all $i, j \in I$. Assume that we have constructed a quasi-order $\leq_L$. In step $L + 1$ of the process we construct a quasi-order $\leq_{L+1}$ by adding all item pairs (i, j) to $\leq_L$ for which the following two conditions hold:

- $b_{ij} \leq L + 1$
- $i \leq_{L+1} j$ does not cause an intransitivity to other dependencies which are already contained in $\leq_L$ or added in this step to $\leq_L$.

The algorithm to construct $\leq_{L+1}$ from $\leq_L$ consists of three sub-steps.

1. Define $A_{L+1} = \{(i, j) \mid b_{ij} \leq L + 1 \wedge i \not\leq_L j\}$.
2. The following steps are repeated until the stopping criterion is valid:
   - Elements of A_{L+1} which cause intransitivity to elements in $\leq_L \cup A_{L+1}$ are added to a set B_{L+1}. If no such elements exists in A_{L+1} the process stops.
   - All elements of B_{L+1} are removed from A_{L+1}.
   - All elements are removed from B_{L+1} and we proceed with the first step. When this process ends none of the remaining elements in A_{L+1} causes an intransitivity with other elements in $\leq_L \cup A_{L+1}$.
3. We define $\leq_{L+1} = \leq_L \cup A_{L+1}$. The relation $\leq_{L+1}$ is by construction transitive.

This construction method results in a set $\mathrm{IPQO}(D) = \{\leq_L \mid L = 0, 1, 2, \ldots, n\}$ of quasi-orders with $\leq_0 \subseteq \leq_1 \subseteq \leq_2 \subseteq \ \ldots \subseteq \leq_n$. Note that some of these quasi-orders may be identical. If, for example, $A_{L+1} = \emptyset$, then $\leq_L = \leq_{L+1}$.

Step 2: Determine the best fitting quasi-order

Assume that $\leq_L$ is the correct quasi-order underlying the data set D. A violation of $i \leq_L j$ is then only possible by the influence of random errors. We estimate the probability γ that a true dependency $i \leq_L j$ is violated due to random errors by:

$$\gamma = \frac{\sum \{b_{ij}/(p_j\ n) \mid i \leq_L j \wedge i \neq j\}}{(|\leq_L| - m)}$$

We distinguish two cases to calculate the expected number of violations t_{ij} to $i \leq_L j$:

1. $i \not\leq_L j$: In this case we assume that the items i and j are independent. Thus, t_{ij} equals the expected number of response patterns d with $d(i) = 0$ and $d(j) = 1$, so we have $t_{ij} = (1 - p_i)\ p_j\ n\ (1 - \gamma)$.

2. $i \leq_L j$ and $i \neq j$: In this case all violations to $i \leq_L j$ must result from random errors. Thus, $t_{ij} = \gamma\ p_j\ n$.

The fit between $\leq_L$ and the data set D is measured by the $\mathrm{diff}(\leq_L, D)$ coefficient:

$$\mathrm{diff}(\leq_L, D) = \frac{\sum_{i \neq j}(b_{ij} - t_{ij})^2}{(m^2 - m)}$$

To determine the optimal tolerance level L we calculate the $\mathrm{diff}(\leq_L, D)$ value for all quasi-orders $\leq_L$ in $\mathrm{IPQO}(D)$ and chose the quasi-order for which this value is minimal as best-fitting quasi-order accordingly to inductive ITA.

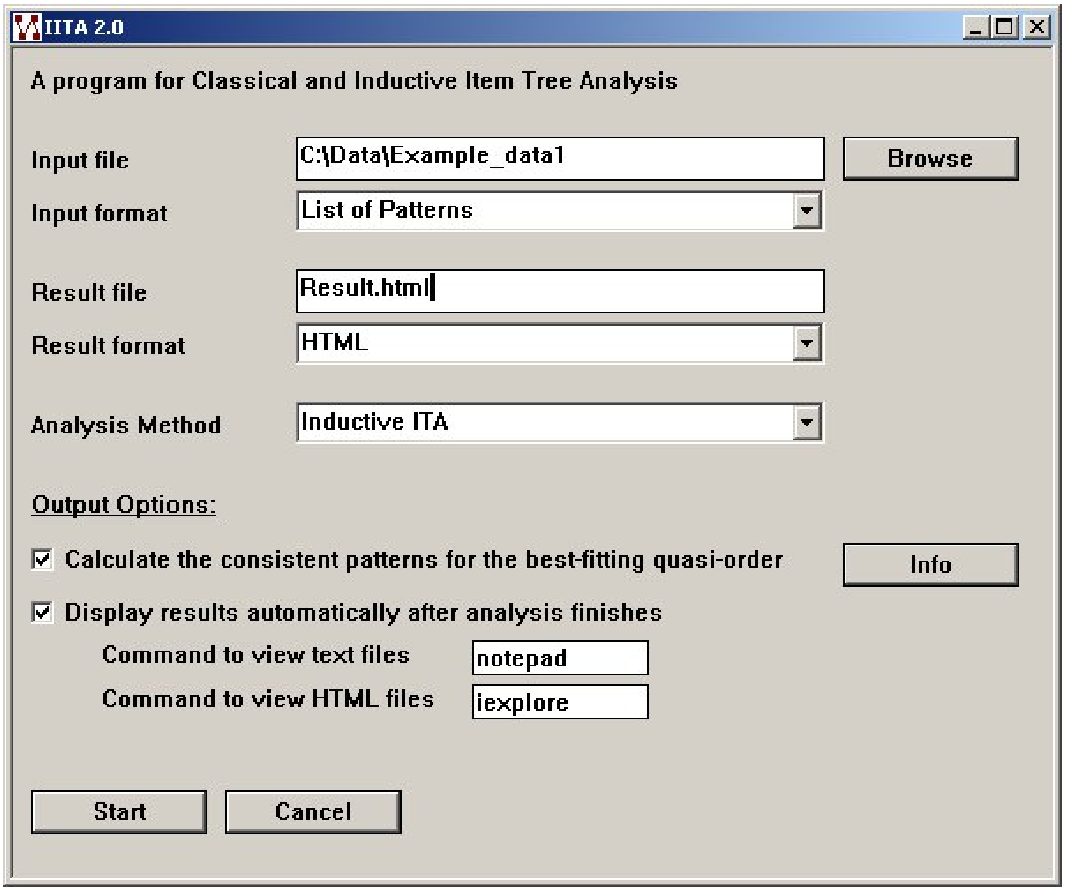
Again this algorithm can easily be extended to the case of missing data in some response patterns. Response patterns where one of the items i or j is not answered by the subject are ignored for the calculation of the b_{ij} values and in the calculation of the error parameter γ. The rest of the algorithm stays more or less the same. For an exact mathematical description of the generalized algorithm see Schrepp (2003).

3. The analysis program ITA 2.0

ITA 2.0[1] is a Windows based program which allows analysing a binary data set D by classical or inductive ITA. The program was tested under Windows XP and Windows 2000, but should also run on other Windows versions. It has a simple user interface (see Figure 1) which allows the user to set the available options for the analysis.

The data are read from the ASCII file specified in the field *Input file*. Currently two input formats (*Patterns with frequencies*, *List of Patterns*) are supported.

[1] **ITA** 2.0 is the successor of **ITA** 1.0 which was a simple DOS program to perform an inductive ITA. This version is available under `http://www.mpr-online.de/issue19/`. The main progress in **ITA** 2.0 is that it implements both methods of ITA, has a graphical user interface and supports HTML output, especially a graphical representation of the constructed best-fitting quasi-order.



IITA 2.0

A program for Classical and Inductive Item Tree Analysis

Input file: C:\Data\Example_data1

Input format: List of Patterns

Result file: Result.html

Result format: HTML

Analysis Method: Inductive ITA

Output Options:

☑ Calculate the consistent patterns for the best-fitting quasi-order

☑ Display results automatically after analysis finishes

Command to view text files: notepad

Command to view HTML files: iexplore

Figure 1: User interface of **ITA** 2.0.

The format *List of Patterns* assumes that the input file contains per subject one row of data which describes the answers of the subject to the items in the item set. Each row can contain the signs 1, 0, - (for missing data), and white space. The spaces are ignored in the input file, thus it does not matter if the entries for two items are separated by space or not.

Examples:

```
10101¶        1 0 1 0 1¶
-1101¶        - 1 1 0 1¶
-00-1¶        - 0 0 - 1¶
10111¶        1 0 1 1 1¶
00100¶        0 0 1 0 0¶
  ...            ...
```

The format *Patterns with frequencies* assumes that the input file contains for each observed response pattern a string which describes the response pattern and in addition the frequency of the response pattern in the data. Both parts of the row must be separated by a space. Inside the part of the row which describes the response pattern only the signs 1, 0, - (for missing data) are allowed.

Example:

```
10101 5¶
01-01 12¶
1-0-1 18¶
10111 2¶
00100 7¶
...
```

In both data formats the number of the items must not be specified explicitly. This number is determined from the first row of the input file. The number of items is restricted to a maximum of 30 and the number of response patterns in the data file is restricted to a maximum of 10000. Please make sure that each row of the data file ends with a new line (¶)!

The result of the analysis is written to a file (enter the name of the file into the field *Result file*) which can be a simple ASCII file (choose *Result format* Text) or an HTML file (choose *Result format* HTML).

Both formats contain the same information. The only exception is that the HTML file contains also a graphical visualization of the best fitting quasi-order as a Hasse-Diagram. This Hasse-Diagram is displayed in a Java-Applet inside the HTML file. For drawing this Hasse-Diagram the program uses the *Interactive Poset and Lattice Drawing Java Applet* from Peter Jipsen (freely available under http://math.vanderbilt.edu/~pjipsen/gap/posets.html).

3.1. Result file for classical ITA

The header of the result file contains some descriptive data, like for example the number of items in the data file.

- *Distribution of values per column:* This table provides an overview about the distribution of the values 0, 1, and - in each of the columns of the data matrix.
- *Table of the a_{ij}:* The value a_{ij} is the number of rows in the data matrix which contain a 0 in column j and a 0 in column i.
- *Table of the b_{ij}:* The value b_{ij} is the number of rows in the data matrix which contain a 1 in column j and a 0 in column i.
- *Table of the c_{ij}:* The value c_{ij} is the number of rows in the data matrix which contain a 0 in column j and a 1 in column i.
- *Table of the d_{ij}:* The value d_{ij} is the number of rows in the data matrix which contain a 1 in column j and a 1 in column i.
- *Empirical correlations r_{ij}:* This table contains the Pearson-Correlations of all item pairs.
- *CA values:* For each constructed relation $\leq_L$ the number of non-reflexive implications in $\leq_L$ and the fit indices CA($\leq_L, D$) and REP-PO($\leq_L, D$) are listed. The relatively best-fitting relation $\leq_L$ is the transitive relation which shows the maximal CA-value. Since the constructed relations $\leq_L$ are not always transitive the information about the transitivity of the relation is displayed in the last column.

- *Constructed quasi-order for level x:* This table lists all non-reflexive implications from the best fitting quasi-order. Together with each implication the the b_{ij}-value of the implication is listed.

- *Fit indices:* The mean violation of an implication (mean over all b_{ij}-values for all $i \leq j$ in the best-fitting quasi-order $\leq$), reproducibility coefficient, correlational agreement coefficient, and the REP-PO value of the best fitting quasi-order are listed.

- *Compatible states:* This is the list of consistent patterns Cons($\leq$) for the best-fitting quasi-order $\leq$.

In the HTML format in addition to this information the best-fitting quasi-order is displayed as a Hasse-Diagram.

3.2. Result file for inductive ITA

The header of the result file contains some descriptive data, like for example the number of items in the data file.

- *Distribution of values per column:* This table provides an overview about the distribution of the values 0, 1, and - in each of the columns of the data matrix.

- *Table of the b_{ij}:* The observed b_{ij}-values.

- *Quasi-orders (lowest level with $i \leq j$):* The algorithm uses an inductive method to construct a set $\{\leq_L | \ L \in \{0, \ldots, n\}\}$ of quasi-orders (n is the number of rows in the data matrix). This table gives an overview about these quasi-orders. An implication $i \leq_L j$ is true, if the value in cell (i, j) of this table is $\leq L$.

- *Diff-values:* For each constructed quasi-order $\leq_L$ the number of non-reflexive implications in $\leq_L$ and the fit index diff$(\leq_L, D)$ are listed. The relatively best fitting quasi-order is the quasi-order which shows the minimal diff-value.

- *Constructed quasi-order for level x:* This table lists all non-reflexive implications from the best fitting quasi-order. Together with each implication $i \leq j$ the *Support*, *Confidence*, and the b_{ij}-value of the implication are listed. Support and confidence are often used to evaluate the quality of an implication for prediction in data mining. They are not used in inductive ITA, but since they are easy to compute we list them together with the implication. Assume $i \leq j$. We define:

  - $q(i, j) =$ Number of rows in the data with $d(j) = 1 \wedge d(i) \neq -$
  - $x(i, j) =$ Number of rows in the data with $d(j) \neq - \wedge d(i) \neq -$
  - $y(i, j) =$ Number of rows in the data with $d(j) = 1 \wedge d(i) = 1$

  Then support and confidence of the implication $i \leq j$ are given by:

  - Support$(i \leq j) = y(i, j)/x(i, j)$
  - Confidence$(i \leq j) = y(i, j)/q(i, j)$

- *Fit indices:* The mean violation of an implication and the reproducibility coefficient are listed.

In the HTML format in addition to this information the best-fitting quasi-order is displayed as a Hasse-Diagram.

3.3. Options

The user is able to decide if the set of consistent patterns for the best-fitting quasi-order should be computed and listed in the output file (mark the checkbox *Calculate the consistent patterns for the best-fitting quasi-order*). Please use that feature carefully if your input file contains many items! The time necessary to compute the compatible states increases exponentially with the number of items. Thus, for higher numbers of items you must expect a considerable time until the program finishes. Note also that the output file can be in this case quite large.

With the checkbox *Display results automatically after analysis finishes* you can force the program to display the result file directly after the analysis is finished. If this checkbox is checked then the two additional fields *Command to view text files* and *Command to view HTML files* are visible (otherwise these files are hidden).

If you choose this option you must specify the program which should be used to display the result files. If the output format is *Text* then you have to enter the name of an editor which is able to display *.txt* files in the field *Command to view text files* (simply choose *notepad* which is usually available on each machine). If the output format is HTML, then you have to enter the program name for your browser in the field *Command to view HTML files* (for example *iexplore* for MS Internet Explorer or *firefox* for the Mozilla Firefox Browser).

3.4. Installation

ITA 2.0 consists of the following executables:

- `IITA_UI.exe`: The graphical user interface of **ITA** 2.0. Click on this executable to start the program.
- `Classic_ITA_UI.exe`: Analysis module for classical ITA (can not be started standalone).
- `Inductive_ITA_UI.exe`: Analysis module for inductive ITA (can not be started standalone).
- `Node.class`, `Poset.class`, `Edge.class`, `GraphPanel.class`: These are the Java classes for the *Interactive Poset and Lattice Drawing* Java *Applet* from Peter Jipsen, which is used to paint the Hasse-Diagram of the best-fitting quasi-order in the HTML output.

Please make sure that these files and the two files `ita_mslg` and `ita_settings` are located in the same directory. Otherwise the program will not work correctly. The mentioned executables and the corresponding source files required to compile them are available together with this article. Details concerning the installation can be found in the file `README.txt`.

4. Example of an analysis by ITA

We will now give an example for the possibilities of an analysis of a data set by ITA. Therefore we analyse the statements of question 4 of the *International Social Science Survey Programme (ISSSP)* for the year 1995 by inductive and classical ITA.

The ISSSP is a continuing annual program of cross-national collaboration on surveys covering important topics for social science research. The program conducts each year one survey with comparable questions in each of the participating nations. The theme of the 1995 survey was national identity. We analyze the results for question 4 for the data set of Western Germany. The statement for question 4 was:

Some people say the following things are important for being truly German. Others say they are not important. How important do you think each of the following is:

1. to have been born in Germany
2. to have German citizenship
3. to have lived in Germany for most of one's life
4. to be able to speak German
5. to be a Christian
6. to respect Germany's political institutions
7. to feel German

The subjects had the response possibilities *Very important*, *Important*, *Not very important*, *Not important at all*, and *Can't choose* to answer the statements. To apply ITA to this data set we changed the answer categories. *Very importantt* and *Important* are coded as 1. *Not very important* and *Not important at all* are coded as 0. *Can't choose* was handled as missing data. Figure 2 shows the resulting quasi-orders $\leq_{IITA}$ from inductive ITA and $\leq_{CITA}$ from classical ITA.

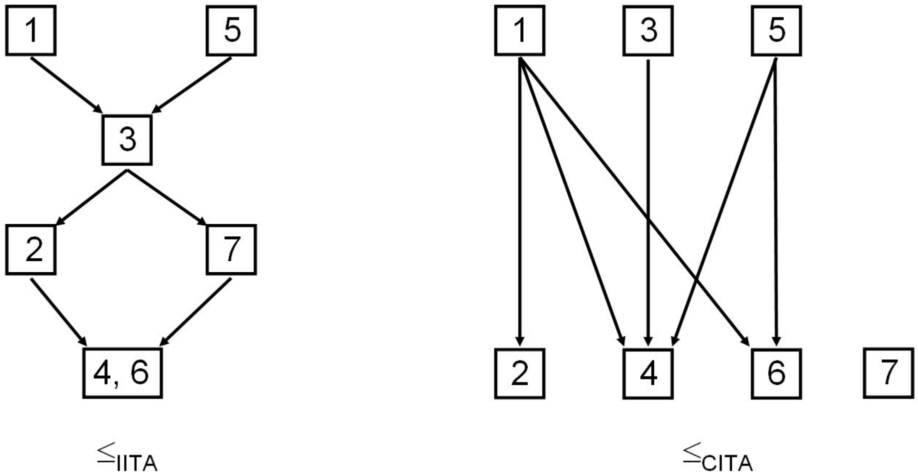


Figure 2: Hasse-Diagrams of the best-fitting quasi-orders accordingly to classical and inductive ITA.

As we can see the quasi-order $\leq_{IITA}$ is more restrictive than $\leq_{CITA}$. The set $Cons(\leq_{IITA})$ contains 9 consistent patterns while $Cons(\leq_{CITA})$ contain 36 consistent patterns.

Another remarkable result is that despite the fact that $\leq_{IITA}$ is much stricter than $\leq_{CITA}$ the reproducibility coefficients of both solutions are similar. We have $\mathrm{Repro}(\leq_{IITA}, D) = 0.94$ and $\mathrm{Repro}(\leq_{CITA}, D) = 0.956$. So the quasi-order $\leq_{IITA}$ still explains with 9 possible response patterns 94% of the cells in the data matrix. Thus, the additional restrictions contained in $\leq_{IITA}$ seem to be well supported by the data.

References

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

Bart WM, Airasian P (1974). "Determination of the Ordering Among Seven Piagetian Tasks by an Ordering-Theoretic Method." *Journal of Educational Psychology*, **66**(2), 277–284.

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

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

Doignon JP, Falmagne JC (1998). *Knowledge Spaces.* Springer-Verlag, Berlin.

Duquenne V (1987). "Conceptual Implications Between Attributes and some Representation Properties for Finite Lattices." In B Ganter, R Wille, K Wolfe (eds.), "Beiträge zur Begriffsanalyse: Vorträge der Arbeitstagung Begriffsanalyse, Darmstadt 1986," pp. 313–339. Wissenschafts-Verlag, Mannheim.

Flament C (1976). *L'Analyse Booléenne de Questionnaire.* Mouton, Paris.

Gagné RM (1968). "Learning Hierarchies." *Educational Psychology*, **6**, 1–9.

Hájek P, Havel I, Chytil M (1966). "The GUHA-Method of Automatic Hypotheses Determination." *Computing*, **1**, 293–308.

Hájek P, Havránek T (1977). "On Generation of Inductive Hypotheses." *International journal of Man-Machine Studies*, **9**, 415–438.

Härtner R, Mattes K, Wottawa H (1980). "Computerunterstützte Hypothesenagglutination zur Erfassung Komplexer Zusammenhänge." *EDV in Medizin und Biologie*, **2**, 53–59.

Held T, Korossy K (1998). "Data-Analysis as Heuristic for Establishing Theoretically Founded Item Structures." *Zeitschrift für Psychologie*, **206**, 169–188.

Janssens R (1999). "A Boolean Approach to the Measurement of Group Processes and Attitudes. The Concept of Integration as an Example." *Mathematical Social Sciences*, **38**, 275–293.

Klemettinen M, Mannila H, Ronkainen P, Toivonen H, Verkamo I (1994). "Finding Interesting Rules from Large Sets of Discovered Association Rules." In "Proceedings of the Third International Conference on Information and Knowledge Management (CIKM94)," pp. 401–407. ACM Press.

Scandura J (1991). "Deterministic Theorizing in Structural Learning: Three Levels of Empiricism." *Journal of Structural Learning*, **3**, 21–53.

Schrepp M (1999). "On the Empirical Construction of Implications on Bi-valued Test Items." *Mathematical Social Sciences*, **38**(3), 361–375.

Schrepp M (2002). "Explorative Analysis of Empirical Data by Boolean Analysis of Questionaires." *Zeitschrift für Psychologie*, **210**(2), 99–109.

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

Schrepp M (2006). "Properties of the Correlational Agreement Coefficient: A Comment to Ünlü & Albert (2004)." *Mathematical Social Sciences*, **51**, 117–123.

Theuns P (1994). "A Dichotomization Method for Boolean Analysis of Quantifiable Cooccurence Data." In G Fischer, D Laming (eds.), "Contributions to Mathematical Psychology, Psychometrics and Methodology," Scientific Psychology Series, pp. 173–194. Springer-Verlag, New York.

Theuns P (1998). "Building a Knowledge Space via Boolean Analysis of Co–occurrence Data." In C Dowling, F Roberts, P Theuns (eds.), "Recent Progress in Mathematical Psychology," Scientific Psychology Series, pp. 173–194. Lawrence Erlbaum Associates Ltd, Mahwah, NJ.

Toivonen H (1996). "Sampling Large Databases for Association Rules." In "22th International Conference on Very Large Databases (VLDB96)," pp. 134–145.

Ünlü A, Albert D (2004). "The Correlational Agreement Coefficient CA($\leq$,D): A Mathematical Analysis of a Descriptive Goodness-of-Fit Measure." *Mathematical Social Sciences*, **48**, 281–314.

Van Buggenhaut J, Degreef E (1987). "On Dichotomization Methods in Boolean Analysis of Questionnaires." In E Roskam, R Suck (eds.), "Mathematical Psychology in Progress," Elsevier Science Publishers B.V., North Holland.

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

Wottawa H, Echterhoff K (1982). "Formalisierung der Diagnostischen Urteilsfindung: Ein Vergleich von Linearen und auf Psychologenaussagen gestützte Konfigurale Ansätze." *Zeitschrift für Differentielle und Diagnostische Psychologie*, **3**(4), 301–309.

Affiliation:

Dr. Martin Schrepp
Schwetzinger Strasse 86
68766 Hockenheim, Germany
E-mail: martin.schrepp@sap.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume 16, Issue 10 *Submitted:* 2005-12-14
September 2006 *Accepted:* 2006-09-06