Mining Implications From Data

Ahcène Boubekki and Daniel Bengs

Deutsches Institut für Internationale Pädagogische Forschung
Frankfurt am Main, Germany
{boubekki, bengs}@dipf.de

Abstract. Item Tree Analysis (ITA) can be used to mine deterministic relationships from noisy data. In the educational domain, it has been used to infer descriptions of student knowledge from test responses in order to discover the implications between test items, allowing researchers to gain insight into the structure of the respective knowledge space. Existing approaches to ITA are computationally intense and yield results of limited accuracy, constraining the use of ITA to small datasets. We present work in progress towards an improved method that allows for efficient approximate ITA, enabling the use of ITA on larger data sets. Experimental results show that our method performs comparably to or better than existing approaches.

1 Introduction

Systematic implications between variables in datasets arise whenever the generating variables are correlated and are at the heart of almost any data analysis procedure. For instance, in the analysis of sales data, knowledge about which products are usually bought together is beneficial in deducing marketing strategies, in the social sciences hierarchical relations in questionnaire data can uncover structures of underlying traits, and in the field of educational data mining knowledge requirements for solving test items can be revealed. We consider the case where the underlying variables form a strict hierarchy, that is, there are deterministic implications between variables. For instance, in educational testing, a testee who solves a difficult test item is very likely to solve all easier items as well. Similarly, in the case of questionnaires in the social sciences, items are often formulated as statements that relate to a latent trait. Here it is natural to expect that agreement with a strong statement implies agreement with all weaker statements.

When responses to test or questionnaire items are observed in a realistic setting, due to random errors in the measurement process, guessing and careless

 In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at http://ceur-ws.org

mistakes by testees, response data invariably exhibits answers that are inconsistent with the item hierarchy, making the implications impossible to observe directly. The challenge is then to reconstruct implications from noisy data.

Formally, the problem statement is the following: Consider a finite set I of n binary items, taking values in $\{0, 1\}$. If it holds that variable $j = 1$ whenever variable $i = 1$, we say that i implies j and write $i \sqsubseteq j$. We require the implications to be logically consistent, that is, the assertion of transitivity holds:

$$i \sqsubseteq j \text{ and } j \sqsubseteq k \implies i \sqsubseteq k \tag{1}$$

As each item implies itself, $i \sqsubseteq i$ for all items i, the relation $\sqsubseteq$ is a quasi-order on I. During the measurement process, patterns that obey $\sqsubseteq$ are perturbed by random noise. The implication mining algorithm aims to reconstruct the original quasi-order as closely as possible.

The problem has first been considered by Van Leeuwe [10], who introduced an algorithm called Item Tree Analysis (ITA). Schrepp [6, 8] suggested to construct implications inductively, showing that his method was more accurate than the original algorithm. Sargin and Ünlü [5] also proposed improvements to the inductive ITA algorithm. Still, the state-of-the-art algorithms for item tree analysis are limited in terms of accuracy and computational feasibility for large datasets. In this paper, we present modifications to the inductive ITA algorithm that lead to significantly reduced execution times and increased accuracy.

Association rule mining (e.g. [1], [2]) is related to ITA, as both methods seek to uncover asymmetric relations between items. Association rule mining aims at finding local hierarchies in the data, while ITA builds a global one, meaning that implications need to hold for all cases in the data set, with exceptions being attributed to random noise. In contrast, association rules can be acceptable if they hold for a minimum fraction of cases. Consequently, the difference is what criterion is used to evaluate the relations. The reliability of an ITA implication $i \Rightarrow j$ is given by the probability $P(\neg i \vee j)$, while for an association rule $i \to j$ the confidence criterion is related to the probability $P(i \wedge j \mid i)$, which can be expressed as $P(i \wedge j \,|\, i) = P(\neg i \vee j \,|\, i)$. We compare the results of our algorithm to association rules mined using the `apriori` algorithm [2] of the `arules` package for R [4].

2 Inductive Item Tree Analysis

We will proceed by explaining the current approach as used by [5] based on the algorithm described by [6]. In section 3 we show how both steps can be improved and introduce our algorithm. First, let us set some notation that will be used in the rest of the paper.

2.1 Preliminaries

A binary matrix, Q, is a matrix with coefficients equal to 0 or 1. For example, the incidence matrix of a relation is a binary matrix. A pattern p from such a matrix is defined by :

$$p = \sum_{i=1}^{n} a_i q_i$$

where q_i are the lines of the matrix and $(a_i) \in \{0,1\}^n$. The set of all possible patterns, $pattern(Q)$, is obtained by considering all the possible values of the binary vector (a_i). However the duplicates are considered only once.

Let us consider a set I of n properties and $\mathcal{D} = \{d_1, ..., d_m\}$ a data set of m observations of these properties. It can be seen as a $n \times m$ binary matrix. From an educational point of view, the columns are the items of a test and the rows represent answers of the students. We define $p_i = |\{s|d_s[i] = 1\}|$ as the number of observations having the property i, and $b_{i,j} := |\{s|d_s[i] = 1 \wedge d_s[j] = 0\}|$ as the number of observations contradicting $\neg i \vee j$. Denote by $(\beta_L)_{L=1}^{n^2}$ the sequence of $b_{i,j}$ in ascending order.

The data generation process is the following: Starting with a quasi-order Q on a fixed number of items, first the exhaustive set of possible pattern, $pattern(Q)$, is constructed. As Q is reflexive, $pattern(Q)$ contains the n-vectors $\mathbb{1}_n$ and 0_n. The data set is then generated from a collection of patterns by adding noise, that is, flipping coefficients with a prescribed probability τ.

An implication between two properties i and j can be written as a disjunctive logic expression : $i \implies j$ is equivalent to $\neg i \vee j$ and its negation is $i \wedge \neg j$. The more the relation is satisfied in the data set, the more the implication is likely ; or by duality : the more the negation is contradicted, the more it is likely. As the dual formulation is a conjunctive expression, it is easier to extract, which is why it is commonly used to evaluate the confidence in the relation. For two items i, j the number of times the implication $i \implies j$ is contradicted is given by $b_{i,j}$. So the smaller is $b_{i,j}$, the more likely is the relation $i \implies j$.

2.2 Inductive ITA Algorithm

The inductive approach to ITA due to [8] is a two-step procedure:

1. Generate candidate set $\mathcal{C}$
2. Select the best fitting quasi-order

The first step given by Schrepp [6] is a recursive algorithm, as it uses the relation $\sqsubseteq_L$ in the generation of $\sqsubseteq_{L+1}$, finally returning the exhaustive set of up to $n(n-1)+1$ candidate quasi-orders.

Original Candidate Set Generation

- Initialisation : $\sqsubseteq_0 = \{(i,j) \mid b_{i,j} = 0\}$ which is a quasi-order.
- Suppose $\sqsubseteq_L$ is a candidate quasi-order.
  - Build $A_{L+1} = \{(i,j) \mid b_{i,j} \leq \beta_{L+1}$ and $(i,j) \notin \sqsubseteq_L$.
  - Remove all elements of A_{L+1} causing intransitivity in $\sqsubseteq_L \cup A_{L+1}$.
- Set $\sqsubseteq_{L+1} = \sqsubseteq_L \cup A_{L+1}$.

The second step is the selection of a relation according to a measure of goodness of fit. In [5] the authors suggested the following method based on a supposed expected numbers of contradictions $b^*_{i,j}$, then the quasi-order fitting best to the observations is selected as follows:

Original Fit

For each quasi-order $\sqsubseteq$ in the candidate set.

- Compute $\gamma = \dfrac{\sum_{\substack{i \sqsubseteq j \\ i \neq j}} \frac{b_{i,j}}{p_j}}{|\sqsubseteq| - n}$.
- For each pair (i,j), determine $b^*_{i,j}$:
  - if $i \sqsubseteq j$, then $b^*_{i,j} = \gamma p_j$
  - if $i \not\sqsubseteq j$ and $j \sqsubseteq i$, then $b^*_{i,j} = p_j - p_i + p_i\gamma$
  - if $i \not\sqsubseteq j$ and $j \not\sqsubseteq i$, then $b^*_{i,j} = (1 - p_i/m)p_j$
- Evaluate $\mathit{diff}(\sqsubseteq) = \dfrac{\sum_{i \neq j}(b_{i,j} - b^*_{i,j})^2}{n(n-1)}$.
- Return $\operatorname{argmin} \mathit{diff}(\sqsubseteq)$

3 Critique and Refinements

Up to now, only the step 2 has been criticized and improved by [5], although at least two points of the first step also need consideration. There are three points that we will address: First, the number of candidates can be reduced by only selecting the most salient quasi-orders, for which we propose a principled way. Second, the way transitivity is enforced in the original algorithm by removing offending pairs depends on the order of removal which is not controlled. We propose a modification to reduce the dependency on the order by reintegrating previously removed pairs. Third, concerning the asymmetry of the fitting coefficient has even been reinforced by the modifications proposed in [5]. To overcome this problem, we propose a new fitting coefficient. To support the discussion, we consider an example using a dataset of size $m = 1000$ created from a synthetic quasi-order on 9 items as described above.

3.1 Selecting the Candidates

In a naive approach, the exhaustive set of up to $n(n-1)+1$ quasi-orders are included in the candidate set. We propose to reduce the candidate set by considering only the most salient quasi-orders. These are the ones where the number of contradictions rises significantly. Looking at the sequence β_L in the example, there are pronounced steps followed by almost level parts. The problem is now to detect the "steps" in the curve. We do so by computing the standard deviation to the cumulative sets of differences of two consecutive terms of (β_L) and denote the resulting sequence by (σ_L), thus

$$\sigma_L = \sigma(\{\beta_{l+1} - \beta_l,\ 1 \leq l \leq L\}, \quad \text{for } 1 \leq L \leq n^2.$$

As it is evident in Figure 1, where β_L (contradictions, black curve) and σ_L (cumulative std., red curve), taking the cumulative standard deviation effectively magnifies the gap between two steps; moreover, the sequence only increases between each steps and then decreases. Therefore, the steps can easily be identified as the indices where an increase of σ_L occurs. We use the last value of each group, as the algorithm will also include all the values on the same level and those on previous levels.

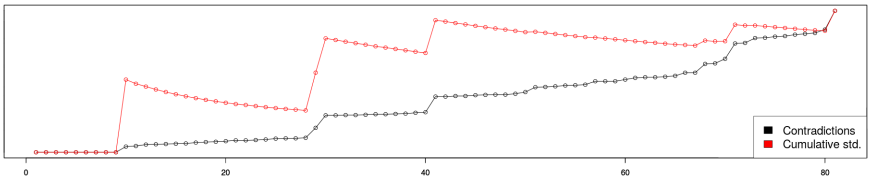

Fig. 1. Number of contradictions and the evolution of the cumulative standard deviation.

Based on this observation, we propose the following method for selecting the candidates:

Selection of Sparse Candidates

- Determine the sequence (β_L) of the ascending $b_{i,j}$.
- Compute the cumulative standard deviation sequence (σ_L).
- If $\sigma_{L+1} > \sigma_L$, the quasi-order $\sqsubseteq_L$ is built.

Using the sparse quasi-order selection algorithm, less quasi-orders are generated, consequently, for step 2, less computation time is needed. As we will show in the experiments, still the good quasi-orders are captured as long as noise levels are reasonable.

We now address the issue of transforming a relation to a transitive one. In Schrepp's algorithm the couples leading to intransitivity are simply removed. We propose to reintegrate these *rejected* pairs by the following procedure:

Reintegration

- Define $A_L = \{(i,j) \mid b_{i,j} \leq \beta_L\}$.
- The set A_L is sorted by increasing value of corresponding $b_{i,j}$.
- As long as A_L is not transitive, the last element of A_L is removed and stored in R_L.
- Repeat:
  - For each element r in R_L.
    If $A_L \cup \{r\}$ is transitive, remove r from R_L and reintegrate it into A_L.
  - If no change of R_L occurs, break.

We conjecture that the result of this algorithm is in fact the biggest quasi-order included in the set $\{(i,j)|b_{i,j} \leq \beta_L\}$. As β_L is strictly increasing, producing the same relation occurs less frequent as in Schrepp's algorithm.

3.2 Fit coefficient

As said before, the fit criterion has been the center of attention in the evolution of the method. The state-of-the-art fit coefficient (see "Original fit" in the previous section) has been proposed by Sargin and Ünlü [5] to improve the one given by Schrepp [8] with regard to quasi-orders with fewer relations. However, there are two problematic aspects to be considered:
Firstly, the formula is not symmetric : for the equivalent cases $i \not\preceq_L j$ and $j \not\preceq_L i$, the coefficient $b^*_{i,j}$ takes completely different forms. Also, the formulae for $b^*_{i,j}$ for the three cases do not allow for an intuitive interpretation.
The second point arises from the later. The fit function *diff* is not consistent, meaning that given the set of quasi-orders resulting from the first step, there are cases where even if the correct quasi-order has been computed, it is not the one that will minimize the *diff* coefficient. Consequently, the wrong quasi-order will be returned.
An example is presented in Figure 2. The red curve represents the *diff* coefficient for each quasi-order produced by the first step, while the black curve is the number of coefficient that differs from the original relation. The correct and original quasi-order is the 29^{th}. It is indicated with a black vertical line. The one minimizing the *diff* is the 26^{th} and is indicated with a red vertical line.

To get rid of this asymmetry, we will build another coefficient based on simple considerations. The probability $P(i \implies j|D)$ that an implication $i \implies j$ is latently included in the data is related to $b_{i,j}$ by

$$P(i \implies j|\mathcal{D}) = 1 - \frac{b_{i,j}}{m}.$$

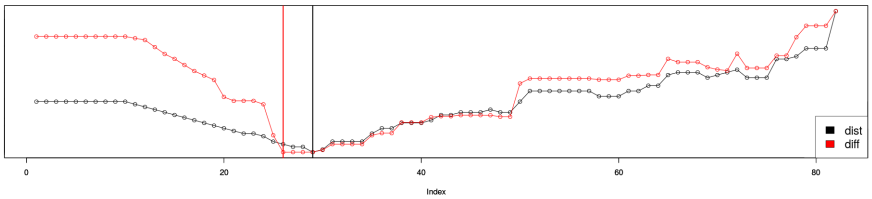

Fig. 2. The correct quasi-order (black vertical line) does not always minimize (red vertical line) the fitting coefficient.

Now the issue is determining which of the candidates fits the data best. Let us call $\mathcal{M}$ the incidence matrix of a retained relation $\sqsubseteq$. From this, the set of possible patterns, $\mathcal{P} = \mathit{pattern}(\mathcal{M})$ is determined. If no noise was involved and $\sqsubseteq$ was the correct quasi-order, every observation would be included in $\mathit{pattern}(\mathcal{M})$. And if every possible pattern had the same probability to happen, then $P(i \implies j|\mathcal{D}) = P(i \implies j|\mathcal{P})$. The closer the relation $\sqsubseteq$ is to the data, the closer are $P(i \implies j|\mathcal{P})$ and $P(i \implies j|\mathcal{D})$. This observation motivates the coefficient *diff* which is computed as follows:

$$\begin{aligned} \mathit{diff}_{new}(\sqsubseteq) &= \sqrt{\sum_{i \neq j} \left(P(i \implies j|\mathcal{D}) - P(i \implies j|\mathcal{P})\right)^2} \\ &= \sqrt{\sum_{i \neq j} \left(\frac{b_{i,j}}{m} - \frac{b^*_{i,j}}{m_{\mathcal{P}}}\right)^2} \end{aligned}$$

where $m_{\mathcal{P}} = |\mathcal{P}|$ and $b^*_{i,j}$ is the number of patterns of $\mathcal{P}$ where the implication $i \implies j$ is contradicted :

$$b^*_{i,j} := |\{p \in \mathcal{P} \mid p[i] = 1 \wedge p[j] = 0\}|$$

Corrected Fit

For each quasi-order in the candidates set.
- Build the set of possible patterns $\mathcal{P}$.
- Compute the numbers $b^*_{i,j}$ of patterns contradicting the implication $i \implies j$.
- Evaluate the fit coefficient $\mathit{diff}_{new}(\sqsubseteq)$.

Return argmin$\mathit{diff}(\sqsubseteq)$

4 Experimental setup and results

We test combinations of our proposed modifications to fit coefficient and generation of candidate set against the original versions using synthetic data. The same setting is used for all three comparisons : 100 of different quasi-orders on 9 elements are built, for each 1000 data sets with 1000 observations lines are created with varying noise rate τ.

4.1 Experiment 1

This first set of experiments focuses on the modification of the first step of the procedure. For each data set, the original quasi-order is searched in the list resulting from the first step. The problem of finding it, is not an issue yet. For three different error rates $\tau \in \{0.05, 0.1, 0.15\}$, we compare the following three algorithms: *Original* , *Original* with *Reintegration*, *Selection* with *Reintegration*. The minimum distance to the correct quasi-order is then computed. If it is equal to 0, it means the correct relation is included in the candidates set. The results are reported in Table 1. These are means over the $100 \times 1\ 000 = 100\ 000$ loops.

$\tau = 5\%$

	Original	Original Reintegration	Selection Reintegration
Minimum	0	0	0
Mean	0.02	0.02	0.05
Maximum	0.45	0.46	0.74
Standard Dev.	0.06	0.06	0.11
Contains Correct	98.4%	98.7%	96.7%

$\tau = 10\%$

	Original	Original Reintegration	Selection Reintegration
Minimum	0	0	0.08
Mean	0.56	0.54	1.14
Maximum	2	2	5
Standard Dev.	0.40	0.41	0.74
Contains Correct	69.8%	70.5%	59.2%

$\tau = 15\%$

	Original	Original Reintegration	Selection Reintegration
Minimum	0.11	0.09	0.40
Mean	1.38	1.44	4.21
Maximum	3.75	4.21	15.51
Standard Dev.	0.68	0.78	2.68
Contains Correct	41.4%	42.0%	27.0%

Table 1. Comparison of three first step algorithms for different noise levels.

When noise increases all the algorithms behave badly. The algorithms *Original with Reintegration* and *Original* tolerate higher noise levels, even though the mixed algorithm performs better across all noise levels. It is important to point out that the mean distances stay around 1. The algorithm *Selection Reintegration* quality drops rapidly. Particularly the maximum distance goes up to

around 15, but remarkably, the standard deviation and the mean stay quite low : if *Selection* does not contain the correct quasi-order, it is still close. This shows that the selection of the sparse candidate set works in most cases.

4.2 Experiment 2

Here the interest is put on the second step, which means to compare the different fit coefficients. Again three combinations of algorithms are compared. The original *diff* coefficient is combined with the original first step algorithm to reproduce the original procedure, and with the *Selection Reintegration*. Finally the corrected $diff_{new}$ is combined with the *Selection Reintegration* to show the performance of both combined. The settings of the experience is the same as previously. The results are reproduced in Table 2. The row *Found Correct* is the percentage of times the algorithm has found the correct quasi-order, and *Found Closest* is the percentage of times it has found the one in the possible set that is the closest (or equal) to the original one.

The results are clearly in favor of the improvements proposed in the article. The inductive ITA as described by Schrepp [8] and with the fit function proposed by Sargin & Ünlü [5] is able to detect the correct relation hardly 1 time over 3. This is not related to the *Original* first step, because as Table 1 shows, the correct quasi-order has a probability to be in the candidate set varying between 98% and 41% depending on the noise. The combination of the original second step and the proposed first step supports our critique of the original fit coefficient. Indeed, as there is less choice for the fit coefficient, the percentage of correct is bigger than the *Original-Original* combination. Moreover, the *Selection Reintegration* contains the correct quasi-order less frequently . On the other hand, the mix *Corrected* and *Selection Reintegration* produces the best results. It finds the closest relation in the candidates set more than 82.5% of time. This is interesting, because this algorithm for the first step often does not contain the correct relation for higher noise but it is still close to it.

4.3 Experiment 3

In this final set of experiments, we explore to what extent association rule mining can be used to mine implications. Albeit association rule mining targets n-ary antecedents and obviously will not produce transitive relations, we test whether implications are recovered as first order rules, i.e. rules of the form $i \Rightarrow j$. For this purpose we mine association rules with the R-package `arules` developed by M. Hahsler et al., and only extract first order rules. As the package does not allow for easy computation of the incidence matrix, only the number of pairs included in a relation will be considered.

The comparison is done with the proposed combination *Corrected diff - Selection Reintegration*. For the `apriori` method we select rules with confidence greater than 75% and a support greater than 1%, as this gave the best results. The noise level is set to $\tau = .5$ and .1. Results are presented in Table 3. The comparison is quite rough, as we do not check whether the correct relations are

$\tau = 5\%$

diff	Original	Original	Corrected
First Step	Original	Selection Reintegration	Selection Reintegration
Minimum	0.03	0.01	0.00
Mean	1.48	0.71	0.07
Maximum	5.41	5.35	1.58
Standard Dev.	1.19	1.07	0.19
Found Correct	35.3%	66.4%	96.1%
Found Closest	36.3%	69.1%	98.3%

$\tau = 10\%$

diff	Original	Original	Corrected
First Step	Original	Selection Reintegration	Selection Reintegration
Minimum	0.01	0.08	0.08
Mean	2.23	1.77	1.43
Maximum	7	10	12
Standard Dev.	1.38	1.51	1.39
Found Correct	19.3%	43.7%	56.6%
Found Closest	31.2%	76.6%	90.7%

$\tau = 15\%$

diff	Original	Original	Corrected
First Step	Original	Selection Reintegration	Selection Reintegration
Minimum	0.18	0.40	0.40
Mean	3.77	6.21	5.89
Maximum	11.74	23.65	26.63
Standard Dev.	1.98	4.74	4.54
Found Correct	8.4%	21.9%	23.4%
Found Closest	28.0%	79.1%	82.5%

Table 2. Comparison of three *diff* coefficients for different noise level.

recovered, but only if their number is correct. The results show that `arules` is outperformed by our ITA algorithm. While ITA gives 98% of good answers, the `arules` only reaches 18%.

5 Conclusion and directions of future work

We proposed three modifications on Item Tree Analysis as presented by [8] and [5]. The first affects the way the transitivity is obtained. This leads to better candidates sets but worsens the computation time. To improve this, we proposed an algorithm that generates a sparse set of candidates containing only the most

$\tau = 5\%$

Algorithm	Selection Reintegration Corrected	Association Rules `arules`
Minimum	0	0
Mean	0.12	3.68
Maximum	6	15
Standard Dev.	0.58	2.72
Found Correct	93.9%	12.3%

Table 3. Comparison between ITA and Association Rules.

salient quasi-orders. Calculation are much faster but the results do not always behave correctly when noise levels are high. The last contribution is a new definition of the fit coefficient, $diff_{new}$. We showed that our improved fit coefficient outperforms the old definition. It also compensates the weakness of the *Selection* algorithm by finding the closest quasi-order.

Both *Selection* algorithm and $diff_{new}$ new coefficient need to be improved to tolerate high noise levels better. Effort should be put on the candidate set generation, as it conditions the results of the second. Theoretical work should also be done such as estimating the probability that the candidates set contains the correct quasi-order. This will surely reveal new directions for further improvement.

In our setting, association rule mining does not apply directly, but a deeper study should be done to reveal the ties between association rules mining and ITA to leverage ideas behind advanced algorithms for association rule mining for implication mining and ITA.

References

1. Agrawal, R., Imieliński, T., Swami, A. *Mining association rules between sets of items in large databases*. Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. p. 207, 1993.
2. Agrawal, R., Srikant, R. *Fast Algorithms for Mining Association Rules in Large Databases*. Proceeding VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases p. 487-499, 1994.
3. Hahsler, M., Gruen, B., Hornik, K. *Computational Environment for Mining Association Rules and Frequent Item Sets*. Journal of Statistical Software 14/15 2005.
4. Hahsler, M., Buchta, C., Gruen, B., Hornik, K. *arules: Mining Association Rules and Frequent Itemsets*. R package version 1.1-3. http://CRAN.R-project.org/package=arules 2014.
5. Sargin, A., Ünlü, A. *Inductive item tree analysis: Corrections, improvements, and comparisons*. Mathematical Social Sciences 58, 376-392, 2009.
6. Schrepp, M. *On the empirical construction of implications between bi-valued test items*. Mathematical Social Sciences 38, 361–375, 1999.

7. Schrepp, M. *Explorative analysis of empirical data by boolean analysis of questionnaires*. Zeitschrift für Psychologie 210, 99–109, 2002.
8. Schrepp, M. *A method for the analysis of hierarchical dependencies between items of a questionnaire*. Methods of Psychological Research 19, 43–79, 2003.
9. Schrepp, M. *On the evaluation of fit measures for quasi-orders*. Mathematical Social Sciences 53, 196–208, 2007.
10. Van Leeuwe, J. *Item Tree Analysis.*. Nederlands Tijdschrift voor de Psychologie, 29, 475–484. 1974.