DOCUMENT RESUME

ED 341 698 TM 017 837

AUTHOR Kelderman, Henk

TITLE Computing Maximum Likelihood Estimates of Loglinear Models from Marginal Sums with Special Attention to Loglinear Item Response Theory. Project Psychometric Aspects of Item Banking No. 53. Research Report 91-1.

INSTITUTION Twente Univ., Enschede (Netherlands). Dept. of Education.

PUB DATE Oct 91

NOTE 45p.

AVAILABLE FROM Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC02 Plus Postage.

DESCRIPTORS *Algorithms; Computer Simulation; *Educational Assessment; Equations (Mathematics); *Estimation (Mathematics); *Item Response Theory; *Mathematical Models; *Maximum Likelihood Statistics; Predictive Measurement; Psychological Testing

IDENTIFIERS Contingency Tables; Iterative Methods; LOGIMO Computer Program; *Log Linear Models

ABSTRACT

In this paper, algorithms are described for obtaining the maximum likelihood estimates of the parameters in log-linear models. Modified versions of the iterative proportional fitting and Newton-Raphson algorithms are described that work on the minimal sufficient statistics rather than on the usual counts in the full contingency table. This is desirable if the contingency table becomes too large to store. Special attention is given to log-linear Item Response Theory (IRT) models that are used for the analysis of educational and psychological test data. To calculate the necessary expected sufficient statistics and other marginal sums of the table, a method is described that avoids summing large numbers of elementary cell frequencies by writing them out in terms of multiplicative model parameters and applying the distributive law of multiplication over summation. These algorithms are used in the computer program LOGIMO, and are illustrated with simulated data for 10,000 cases. Two tables, 3 graphs, and a 34-item list of references are included.
(Author/SLD)

Computing Maximum Likelihood Estimates of Loglinear Models from Marginal Sums with Special Attention to Loglinear Item Response Theory

Research Report

91-1

Henk Kelderman

department of EDUCATION

University of Twente

Division of Educational Measurement and Data Analysis

Project Psychometric Aspects of Item Banking No. 53

Colofon:
Typing: H. Kelderman and L.A.M. Bosch-Padberg
Cover design: Audiovisuele Sectie TOLAB
Toegepaste Onderwijskunde
Printed by: Centrale Reproductie-afdeling
Oplage: 150

TM

Computing Maximum Likelihood Estimates of Loglinear Models from Marginal Sums with Special Attention to Loglinear Item Response Theory

Henk Kelderman

Computing maximum likelihood estimates of loglinear models from marginal sums with special attention to loglinear item response theory , Henk Kelderman - Enschede : University of Twente, Department of Education, October, 1991. - 35 pages

Abstract

In this paper algorithms are described for obtaining the maximum likelihood estimates of the parameters in log-linear models. Modified versions of the iterative proportional fitting and Newton-Raphson algorithms are described that work on the minimal sufficient statistics rather than on the usual counts in the full contingency table. This is desirable if the contingency table becomes too large to store. Special attention is given to log-linear IRT models that are used for the analysis of educational and psychological test data. To calculate the necessary expected sufficient statistics and other marginal sums of the table, a method is described that avoids summing large numbers of elementary cell frequencies by writing them out in terms of multiplicative model parameters and applying the distributive law of multiplication over summation. These algorithms are used in the computer program LOGIMO. The modified algorithms are illustrated with simulated data.

Computing Maximum Likelihood Estimates of Loglinear Models from Marginal Sums with Special Attention to Loglinear Item Response Theory

Purpose

Log-linear models are used increasingly to analyze psychological and educational tests (Cressie & Holland, 1983; Duncan, 1984; Kelderman, 1984, 1989; Tjur, 1982). Current computer prcgrams such as GLIM (Baker & Nelder, 1978), ECTA (Goodman & Fay, 1974) and SPSS LOGLINEAR (SPSS, 1988) for analysis of log-linear models have limited utility when used with models of the size and complexity required in some test and applications to test and item analysis. The computer program LOGIMO is especially designed for this situation. In this paper the algorithms used in LOGIMO are described. The algorithms are useful for the analysis of both ordinary log-linear models and log-linear IRT models. For a discussion of applications of log-linear IRT models the reader is referred to Duncan (1984), Duncan and Stenbeck (1987) and (Kelderman (1984, 1989a, 1989b, 1991)

In this paper three log-linear models are used to describe the algorithms, one ordinary log-linear model and two log-linear IRT models. To keep exposition simple, we assume that each test has four items. Needless to say, the results are valid also for larger numbers of items.

Let there be a sample of N subjects with responses i, j, k and l on four variables. The i, j, k and l are realizations of random variables with joint probability p_{ijkl}. Consider the following examples of parametric models for p_{ijkl}.

Example 1

The first model is an ordinary log-linear model (see e.g. Agresti, 1984) describing interactions between consecutive variables:

$$p_{ijkl} = a_{ij}b_{jk}c_{kl}, \qquad (1)$$

$i = 1, \ldots, I;\ j = 1, \ldots, J;\ k = 1, \ldots, K;\ l = 1, \ldots, L,$

where a_{ij}, b_{jk}, c_{kl} are parameters to be estimated. Even though this simple multiplicative parameterization is not identifiable, it is useful for illustrating the first algorithm described in the next section. An identifiable log-linear formulation of the model with main and interaction effect terms will be presented later.

Example 2

Let i, j, k, l = 0, 1 now be dichotomous item responses and let $m = i + j + k + l$, the simple sum of item scores, be a new variable. Several authors (e.g. Cressie & Holland, 1983; Kelderman, 1984) have shown that the model

$$p_{ijklm} = a_i b_j c_k d_l e_m \qquad (2)$$

is equivalent to the dichotomous Rasch (1960/1980) model. This is readily seen by conditioning on the sum score, which yields the familiar formulation of the conditional Rasch model (Rasch, 1980, p.177):

$$p_{ijkl|m} = a_i b_j c_k d_l \Big/ \sum_{\substack{i+j+k+l=m}} a_i b_j c_k d_l .$$

The parameters in (2) are multiplicative main effect parameters describing the effect of the variables. The usual additive Rasch-item-difficulty parameters can be obtained from them as $(\log a_0 - \log a_1)$, $(\log b_0 - \log b_1)$, etc. They are unique up to an additive constant. Let us note that the variable m in $p_{ijkl|m}$ is redundant because it depends completely on i, j, k, and l. Now consider a two-dimensional log-linear IRT model.

Example 3

The most complicated model considered here contains two variables that depend on item responses. To define these variables, two weights are assigned to each response. These weights or category coefficients are positive integers denoted by $v_1(i)$ and $w_1(i)$, $v_2(j)$ and $w_2(j)$, $v_3(k)$ and $w_3(k)$, $v_4(l)$ and $w_4(l)$ for items i, j, k, and l respectively. New variables may now be defined as the simple sums of weights

$$m \equiv v_1(i) + v_2(j) + v_3(k) + v_4(l),$$

and (3)

$$t \equiv w_1(i) + w_2(j) + w_3(k) + w_4(l),$$

for $i = 1, \ldots, I$; $j = 1, \ldots, J$; $k = 1, \ldots, K$; $l = 1, \ldots, L$. A two-dimensional log-linear IRT model can now be written as:

$$P_{ijklmt} = a_i b_j c_k d_l e_{mt}. \quad (4)$$

Kelderman (1989) showed that, for suitable choice of category coefficients, (4) defines a class of IRT models that includes the partial credit model (Masters, 1382), the multidimensional Rasch (Andersen, 1973; Rasch, 1961) model, and other interesting IRT models. It is easy to see that Model 4 can be expanded to include more items, more weight-sum variables and/or interaction terms as in Example 1.

Problems are likely to arise with the usual algorithms for maximum likelihood estimation of parameters in log-linear models if the number of items or weight-sum variables is large. Most of the currently available algorithms require the storage of the tables of observed and expected counts ($\{f_{ijkl}\}$ and $\{F_{ijkl}\} = \{N\, p_{ijkl}\}$, respectively). These tables can become extremely large if the number of items is not small. For example, if there are twelve four-response items, each table will consist of 17 million cells.

The algorithms described below avoid this problem by computing the parameters directly from certain marginal sums of the contingency table. The next section describes two such algorithms: a modified version of the iterative proportional fitting algorithm, and a version of the Newton-Raphson algorithm. Furthermore, an efficient method to calculate the expected marginal sums is described at the end of the next section. In the applications section, the computational efficiency of this method is assessed, and the modified IPF algorithm is applied to a set of simulated data.

Description

If it is assumed that the subjects respond independently of one another, the frequencies (f_{ijkl}) have a multinomial distribution with index N and probabilities (p_{ijkl}). The likelihood of the models for sample data is

$$h \prod_i \prod_j \prod_k \prod_l (p_{ijkl})^{f_{ijkl}},$$

where h is a function of the data only. The variables m and t are omitted in the above expression. Taking the derivatives of the log likelihood with respect to the parameters and setting them equal to zero, will yield the maximum likelihood equations (see Haberman, 1979, p.448). For the model in Example 1 the maximum likelihood equations become

$$
\begin{aligned}
f_{ij++} - F_{ij++} &= 0, && i = 1, \ldots, I;\ j = 1, \ldots, J. \\
f_{+jk+} - F_{+jk+} &= 0, && j = 1, \ldots, J;\ k = 1, \ldots, K. \\
f_{++kl} - F_{++kl} &= 0, && k = 1, \ldots, K;\ l = 1, \ldots, L.
\end{aligned}
\qquad (5)
$$

where a plus sign replacing an index denotes summation over that index (e.g. $F_{ij++} = \sum_K \sum_L F_{ijkl}$). The marginal sums $\{f_{ij++}\}$, $\{f_{+jk+}\}$, and $\{f_{++kl}\}$ are minimal sufficient statistics for the parameters $\{a_{ij}\}$, $\{b_{jk}\}$, and $\{c_{kl}\}$ respectively. Generally, in log-linear model analysis, the sufficient statistics associated with par ieters are the marginal sums with the same indices as the corresponding parameters. Furthermore, the likelihood equations are obtained by setting the observed sufficient statistics equal to the corresponding expected values under the model. Thus, for Model 2, the likelihood equations are obtained by setting the marginal sums $\{f_{i++++}\}$, $\{f_{+j++}\}$, $\{f_{++k+}\}$, $\{f_{+++l+}\}$ and $\{f_{++++m}\}$ equal to the corresponding expected values $\{F_{i++++}\}$, $\{F_{+j++}\}$, $\{F_{++k+}\}$, $\{F_{+++l+}\}$ and $\{F_{++++m}\}$.

Solving the Equations (5) for the parameters yields the maximum likelihood estimates of the parameters. These equations can not be solved directly, but numerical algorithms are available for their solution (e.g. Baker & Nelder, 1978, Goodman & Fay, 1974)

A Modified Iterative Proportional Fitting Algorithm

In iterative proportional fitting (IPF, Deming & Stephan, 1940), the expected cell counts (F_{ijkl}) are proportionally adjusted to fit the set of marginal sums obtained from the sample. In this section we describe a modified IPF algorithm to adjust parameter estimates rather than expected cell frequencies. This modification alleviates both storage requirements and computational complexity because test-data models usually have much less parameters than expected frequencies.

Let us consider regular IPF. Denoting the expected counts before the adjustment as ($F_{ijkl}^{(old)}$) and after adjustment as ($F_{ijkl}^{(new)}$), start the computational procedure by setting all $F_{ijkl}^{(old)} = 1$. In IPF, the maximum likelihood estimates ($\hat{F}_{ijkl}$) under Model 1 are obtained by repeated application of the adjustments

$$\hat{F}_{ijkl}^{(new)} = \hat{F}_{ijkl}^{(old)} \left(f_{ij++} / \hat{F}_{ij++}^{(old)}\right)$$

$$\hat{F}_{ijkl}^{(new)} = \hat{F}_{ijkl}^{(old)} \left(f_{+jk+} / \hat{F}_{+jk+}^{(old)}\right)$$

$$\hat{F}_{ijkl}^{(new)} = \hat{F}_{ijkl}^{(old)} \left(f_{++kl} / \hat{F}_{++kl}^{(old)}\right)$$

each for $i = 1, \ldots, I$; $j = 1, \ldots, J$; $k = 1, \ldots, K$; $l = 1, \ldots, L$, until convergence is achieved. The algorithm will always converge to a solution satisfying Equations 5. The application of IPF to other models, such as those given in Example 2 and 3, is straightforward.

To adjust parameter estimates rather than expected cell

frequencies, let us first express $\hat{F}_{ijkl}$ in terms of parameters. For the first update, this becomes

$$N\, a_{ij}^{(new)} b_{jk}^{(new)} c_{kl}^{(new)} = N\, a_{ij}^{(old)} b_{jk}^{(old)} c_{kl}^{(old)} (f_{ij++}/F_{ij++}^{(old)})$$

Because the same adjustment $(f_{ij++}/F_{ij++}^{(old)})$ is made for all values of k and l, it suffices to change the parameter a_{ij} only. The remaining parameters b_{jk} and c_{kl} can be treated as constants so that $b_{jk}^{(new)} c_{kl}^{(new)} = b_{jk}^{(old)} c_{kl}^{(old)}$. Therefore, we have

$$a_{ij}^{(new)} = a_{ij}^{(old)} (f_{ij++}/F_{ij++}^{(old)}), \quad i = 1, \ldots, I; \quad j = 1, \ldots, J,$$

Similarly for the other updates, we have

$$b_{jk}^{(new)} = b_{jk}^{(old)} (f_{+jk+}/F_{+jk+}^{(old)}), \quad j = 1, \ldots, J, \quad k = 1, \ldots, K,$$

and

$$c_{kl}^{(new)} = c_{kl}^{(old)} (f_{++kl}/F_{++kl}^{(old)}), \quad k = 1, \ldots, K, \quad l = 1, \ldots, L.$$

Within the modified IPF algorithm, only IJ + JK + KL parameters have to be adjusted in one cycle. Compared to the

3(IJKL) cell frequencies in ordinary IPF, there is a considerable reduction in computational complexity with the modified IPF. We will look at this reduction much more closely in the section on applications.

The IPF algorithm works with indeterminate parameters. A unique solution of the log-linear version of Model 1 with main and interaction effect parameters can be obtained by the following reparameterization.

$$\mu = \log a_{IJ} + \log b_{JK} + \log c_{KL}$$
$$\alpha_i = \log a_{iJ} - \log a_{IJ}$$
$$\beta_j = \log a_{Ij} - \log a_{IJ} + \log b_{jK} - \log b_{JK}$$
$$\gamma_k = \log b_{Jk} - \log b_{JK} + \log c_{kL} - \log c_{KL}$$
$$\delta_l = \log c_{Kl} - \log c_{KL}$$
$$(\alpha\beta)_{ij} = \log a_{ij} - \log a_{Ij} - \log a_{iJ} + \log a_{IJ}$$
$$(\beta\gamma)_{jk} = \log b_{jk} - \log b_{Jk} - \log b_{jK} + \log b_{JK}$$
$$(\gamma\delta)_{kl} = \log c_{kl} - \log c_{Kl} - \log c_{kL} + \log c_{KL}$$

where α_i, β_j, γ_k and δ_l are main effect parameters and $(\alpha\beta)_{ij}$, $(\beta\gamma)_{jk}$ and $(\gamma\delta)_{kl}$ interaction effect parameters.

It is easy to verify that the model (i.e. $\{p_{ijkl}\}$) would remain invariant under this reparameterization. That is:

$$\log a_{ij}b_{jk}c_{kl} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + (\alpha\beta)_{ij} + (\beta\gamma)_{jk} + (\gamma\delta)_{kl} \quad (6)$$

and that the constraints

$$\alpha_I = \beta_J = \gamma_K = \delta_L = (\alpha\beta)_{Ij} = (\alpha\beta)_{iJ}$$
$$= (\beta\gamma)_{jK} = (\beta\gamma)_{Jk} = (\gamma\delta)_{Kl} = (\gamma\delta)_{kL} = 0 \qquad (7)$$

are satisfied.

This parameterization contrasts the effect of each category with the last. Bock (1975, p.239) refers to this as the 'simple contrast'. Other parameterizations such as deviation contrasts, where the effect of each category is contrasted with the mean effect, can be obtained by similar transformations.

A Newton-Raphson Algorithm

The well-known Newton-Raphson algorithm is based on a second order Taylor expansion of the log-likelihood function (Andersen, 1980, p. 47; Adby & Dempster, 1974, p. 65). The algorithm iteratively computes the log-linear parameters using the gradient and the Hessian matrix, which can be written as functions of the marginal sums. Before discussing the Newton-Raphson (N-R) update, let us first introduce the matrix formulation of the log-linear formulation given in (6) for Model 1:

$$\log p_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_l$$
$$+ (\alpha\beta)_{ij} + (\beta\gamma)_{jk} + (\gamma\delta)_{kl}.$$

Without loss of generality, let us assume that I = J = K = L = 2. Unlike IPF, the N-R algorithm requires the parameters to be identified. Therefore we impose the constraints given in (7). Let $\mathbf{p} = (p_{1111}, p_{2111}, \ldots, p_{2222})'$ be the vector of cell probabilities, and let $\xi = (\mu, \alpha_1, \beta_1, \gamma_1, \delta_1, (\alpha\beta)_{11}, (\beta\gamma)_{11}, (\gamma\delta)_{11})'$ be the vector of parameters to be estimated. The matrix version of the model can be written as

$$\log \mathbf{p} = \mathbf{D}\,\xi,$$

where **D** is the design matrix with ones and zero's in the appropriate places and **log** means the elementwise logarithm operator. Letting $\mathbf{f} = (f_{1111}, f_{2111}, \ldots, f_{2222})$ and $\mathbf{A} = \mathrm{diag}(\mathbf{p})$, the gradient vector and the Hessian matrix can be expressed as

$$\mathbf{g} = \frac{\partial \log L}{\partial \xi} = \mathbf{D}'\mathbf{f} - \mathbf{D}'\mathbf{p}N$$

and

$$\mathbf{H} = \frac{\partial^2 \log L}{\partial \xi\, \partial \xi'} = N[\mathbf{D}'\mathbf{A}_p\mathbf{D} - (\mathbf{D}'\mathbf{p})\ (\mathbf{D}'\mathbf{p})'] ,$$

respectively.

These can also be expressed in terms of marginal sums since

$$\mathbf{D'f} = (f_{1+++},\ f_{+1++},\ f_{++1+},\ f_{+++1},\ f_{11++},\ f_{+11+},\ f_{++11}),$$

$$\mathbf{D'p} = (p_{1+++},\ p_{+1++},\ p_{++1+},\ p_{+++1},\ p_{11++},\ p_{+11+},\ p_{++11}),$$

and

$$\mathbf{D'AD} = \begin{bmatrix} p_{1+++} & p_{11++} & p_{1+1+} & p_{1++1} & p_{11++} & p_{111+} & p_{1+11} \\ p_{11++} & p_{+1++} & p_{+11+} & p_{+1+1} & p_{11++} & p_{+11+} & p_{+111} \\ p_{1+1+} & p_{+11+} & p_{++1+} & p_{++11} & p_{111+} & p_{+11+} & p_{++11} \\ p_{1++1} & p_{+1+1} & p_{++11} & p_{+++1} & p_{11+1} & p_{+111} & p_{++11} \\ p_{11++} & p_{11++} & p_{111+} & p_{11+1} & p_{11++} & p_{111+} & p_{1111} \\ p_{111+} & p_{+11+} & p_{+11+} & p_{+111} & p_{111+} & p_{+11+} & p_{+111} \\ p_{1+11} & p_{+111} & p_{++11} & p_{++11} & p_{1111} & p_{+111} & p_{++11} \end{bmatrix} \quad (8)$$

The N-R algorithm repeatedly adjusts the parameters ξ. Let $\xi^{(old)}$ and $\xi^{(new)}$ be the parameter vectors before and after adjustment and let $\mathbf{g}^{(old)}$, $\mathbf{g}^{(new)}$ and $\mathbf{H}^{(old)}$, $\mathbf{H}^{(new)}$ be the gradient and Hessian computed from them. The maximum likelihood estimates of ξ are obtained by repeated application of

$$\xi^{(new)} = \xi^{(old)} + \Delta,$$

where Δ is the solution of the linear system:

$$\mathbf{H}^{(old)}\,\Delta = \mathbf{g}^{(old)}$$

Usually the update Δ is computed by pre-multiplication of system the by the inverse of $\mathbf{H}^{(old)}$, but it is more efficient to solve the system directly for Δ (Dongarra et al, 1979; Holland & Thayer, 1987). Gill, Murray and Wright (1991) describe fast methods for solving systems of linear equations. The Newton Raphson algorithm converges much more rapidly to the maximum likelihood solution than the IPF algorithm but requires starting values that are close to the final solution. Also $\mathbf{H}$ requires the marginal sums given in (8), which are not necessary for the modified IPF algorithm.

The most important feature of the above modifications of the IPF and N-R algorithms is that in neither case is it necessary to set up the full contingency table. Marginal sums alone are sufficient. Although this reduces storage, requirements it does not relieve us of the computational burden of summing over the cells of the full table, which is probably the reason why the above N-R procedure is never used in existing programs for log-linear analysis. A novel element in the application of the N-R algorithm and modified IPF, is that the marginal sums are computed in an efficient way described in the next section.

Efficient Computation of Marginal Sums

The obvious way to compute (F_{ij++}) is to sum over the cells

$$F_{ij++} = N \sum_k \sum_l p_{ijkl} = N \sum_k \sum_l a_{ij} b_{jk} c_{kl}, \qquad (9)$$

$i = 1, \ldots, I$, $j = 1, \ldots, J$, where the last term is used to avoid storage of the full table.

Suppose that $I = J = K = L = 10$, then (9) involves $2(IJKL) + 1 = 20,001$ multiplications and $IJ(KL - 1) = 9900$ summations. This number of computations can be reduced by rewriting (9), using the distributive law of multiplication over summation, as

$$F_{ij++} = N\, a_{ij} \sum_k b_{jk} \sum_l c_{kl},$$

$i = 1, \ldots, I$, $j = 1, \ldots, J$. This requires only $1 + IJ + K = 111$ multiplications and $J(K-1) + K(L-1) = 180$ summations. This is obviously a considerable reduction in the number of computations needed.

We will refer to this method of computing the expected marginal sums as the marginalization-by-variable (MBV) method, because summations for one variable (at a time) are done only over parameters that depend on that variable. Multiplication with parameters that do not depend on that variable is postponed until after the summation.

The MBV method becomes more complicated if the model contains weight-sum variables, because they are dependent on item responses (e.g. Example 3). In that example, the values that a summation in the MBV method can take, may depend on the value of other summation variables. For example, the computation of F_{++++m} in Model 2, can be written as

$$F_{++++m} = N \sum_{\substack{i\ j\ k\ l \\ i+j+k+l=m}} a_i b_j c_k d_l e_m$$

The summations over i, j, k and l may only be performed for those patterns for which i + j + k + l = m. To see what this means for each separate summation let us rewrite i + j + k + l ≡ m into the equivalent form

$$m_1 \equiv i + j,$$

$$m_2 \equiv m_1 + k,$$

$$m \equiv m_2 + l,$$

where m_1 and m_2 are partial sum scores.

Let $\sum_{\substack{x,y \\ x+y=z}}$ mean the summation over the values of x and y for which x + y = z; the MBV method for computing P_{++++m} then becomes

$$F_{++++m} = N\, e_m \sum_{\substack{m_2,l; \\ m_2+l=m}} d_l \left(\sum_{\substack{m_1,k; \\ m_1+k=m_2}} c_k \left(\sum_{\substack{i,j; \\ i+j=m_1}} b_j a_i \right) \right) \qquad (10)$$

In the above equation, a_i and b_j are first multiplied for all i = 0, 1 and j = 0, 1. The products for which i + j = m_1 are summed, which gives a separate sum for each m_1(= 0, 1, 2). Each sum is then multiplied with each of the c_k (k = 0, 1) parameters. Again, these products are summed if m_1 + k = m_2. This yields a sum for each m_2(=0, 1, 2, 3). Finally, this

process of multiplication and summation is repeated one more time to obtain F_{++++m}. In this way, the marginal sums are computed efficiently while, at the same time, avoiding summation over logically impossible combinations of variable values.

In a similar manner, the marginal sums for the model in Example 3 can also be computed. First, rewrite the weight-sums given in (3) as

$$m_1=v_1(i)+v_2(j), \quad m_2=m_1+v_3(k), \quad m=m_2+v_4(l)$$
$$t_1=w_1(i)+w_2(j), \quad t_2=t_1+w_3(k); \quad t=t_2+w_4(l) \tag{11}$$

Under these constraints, the marginal sum p_{++++mt} can be computed as

$$F_{++++mt} = N\, e_{mt} \sum_{l,m_2,t_2} d_l \left(\sum_{k,m_1,t_1} c_k \left(\sum_{i,j} b_j a_i\right)\right).$$

Again each summation can be performed separately if the constraints in (11) are respected. Obviously, the same method can be applied to calculate the other expected marginal sums such as $\{F_{i+++++}\}$, $\{F_{+j++++}\}$, etc. Consequently, the MBV method can supply all marginal sums needed in the modified IPF or N-R algorithm.

The modified IPF algorithm using the MBV method to compute expected marginal sums, is implemented in the computer program called LOGIMO (LOGlinear IRT MOdeling,

Kelderman & Steen, 1988). LOGIMO is a Pascal program that estimates log-linear models with main and interaction effect parameters of item response, background variables and one or more weight-sum variables as shown in Example 3. The weights are integer valued and must be specified by the user. In the next section we present the application of the modified IPF and N-R algorithm.

Application

The complexity of computing the parameters of log-linear models is substantially reduced by using modified IPF and N-R algorithms based on marginal sums that can be computed efficiently by the MBV method. In this section we will examine the computational complexity as a function of the number of variables in the model. We will first look at the increase in computational complexity with the MBV algorithm and then at the full algorithm.

In this application, we restrict our attention to the IPF algorithm and to the simplest model with sum scores as given in (2). This model is chosen because the number of MBV computations is tractable and because it is equivalent to the dichotomous Rasch model. Consequently the parameter estimates can be compared to those of an existing algorithm for computing Rasch parameters and to verify the correctness of the algorithm.

Insert Table 1 here

In Table 1, the numbers of summations and multiplications in the computation of $F_{+...+m}$ of the simple sum-score model (2) are given for five to 20 items. It can be seen that for the MBV algorithm these numbers remain within reasonable limits, whereas, for the case of summing over all cells (9), these numbers increase very rapidly.

To evaluate the full IPF algorithm, test data conforming to the Rasch model were generated for 20 items. The item difficulties where randomly chosen from the uniform distribution over the interval [-2,2]. Latent trait values for 10,000 cases were drawn from a uniform distribution over the [-3,3] interval. Log-linear Rasch models given in (2) were then fitted to these data. Nine computer runs were made for different subsets of items, where the first subset contained the first four items, the second subset contained the first six items etc. In Figures 1, 2, and 3, different statistics of these runs are plotted against the number of items in the model.

Insert Figures 1, 2 and 3 about here

In Figure 1 the number of IPF iterations needed to arrive at the maximum likelihood solution is plotted against the number of items. Iterations were performed on a VAX 8750 computer until none of the parameter estimates could be improved by more than .005. It is seen that the relationship between the number of items and the number of iterations needed for convergence is approximately linear.

As the number of items increases, the CPU time needed for each of these iterations will also increase. In Figure 2, the mean CPU time per iteration is plotted against the number of items. It can be seen that the CPU time increases steeply with the number of items but stays within reasonable limits for moderate numbers of items. In Figure 3, the total CPU time for IPF iterations and for initializing the algorithm is plotted against the number of items. Initialization time includes data input, computing marginal sums and creating data structures for storage. According to Figure 3, the CPU times for initialization increases almost linearly with the number of items and the iteration time does not increase dramatically with the number of items in the test.

In Table 2 the real item difficulties and the estimated item difficulties values of all 20 items are given. The item parameter estimates were obtained by the LOGIMO program and by the PML (Gustafsson, 1977, 1980) program. The PML program calculates the CML estimates of the item parameters with Andersen's (1972) method. In both cases the first item

difficulty parameter was set equal to its real value. Furthermore, the iterations were stopped until none of the parameter estimates could be improved by more than .0001. It can be seen from Table 2 that both solutions are identical up to the second decimal place, indicating that the IPF/MBV algorithm correctly calculates maximum likelihood estimates.

Insert Table 2 here

Finally a note on the usefulness and availability of LOGIMO. For ordinary log-linear models, provided they are not too complicated, LOGIMO makes it possible to analyze larger numbers of variables than with other programs. For certain special Rasch models such as (2), dedicated programs such as RIDA (1989), and PML will generally be faster. If, however, the user wants to define his of her own IRT model with several dimensions and/or user specified category coefficients, LOGIMO is the way to go. LOGIMO is a Pascal program that runs VAX system running under VMS. For smaller problems there is a PC version (386, with extended memory). LOGIMO will be distributed starting somewhere in the summer of 1992 by iec *ProGAMMA*, P.O.Box 841, 9700 AV Groningen, The Netherlands (E-mail: GAMMA@RUG.NL).

Discussion

In this paper an efficient algorithm is described that calculates the parameter estimates of log-linear models including log-linear IRT models. The algorithm avoids setting up the full Item 1 x ... x Item k table by computing the parameter estimates from the marginal sums of the table by a modified version of the iterative proportional fitting algorithm or the Newton-Raphson algorithm. The computation of expected marginal sums is done efficiently using the MBV method.

The methods modified IPF and MBV methods can be seen as generalizations of older methods for the estimation of unidimensional Rasch models. For this case, the modified IPF algorithm turns out to be equivalent to an algorithm proposed by Scheiblechner (1971, see Fischer, 1974, p.247) and the MBV method can be shown to be identical to the so called summation algorithm for the computation of elementary symmetric functions (Andersen, 1972). To see the latter, normalize the parameters in the Rasch model (2) as $a_0 = b_0 = c_0 = d_0 = 1$. Elementary symmetric functions can then be computed recursively using the following type of relations

$$\gamma_m(a_1,b_1,c_1,d_1) = \gamma_m(a_1,b_1,c_1) + d_1\,\gamma_{m-1}(a_1,b_1,c_1),$$

and similar relations for $\gamma_m(a_1,b_1,c_1)$, $\gamma_m(a_1,b_1)$, etc.

It is easy to see that this summation is equivalent to the left-most summation in (10), and $\gamma_m(a_1,b_1,c_1)$ and $\gamma_{m-1}(a_1,b_1,c_1)$ are equivalent to the second summation in (10). Thus, the MBV method for computing marginal sums in the Rasch model is equivalent to the summation algorithm for computing elementary symmetric functions. Despite this for unidimensional Rasch models LOGIMO is genererally slower than programs using the sum algorithm that are dedicated to those models. As remarked before its strength lies in ordinary log-linear models and more complicated log-linear IRT models.

LOGIMO is capable of dealing with models with interaction terms and multiple weight-sum variables with arbitrary weights defined by the user. In these models the nice symmetries of the Rasch model are lost. It is an open question whether improved methods for computing elementary symmetric functions, such as those of Formann (1986) and Verhelst, Glas and van der Sluis (1984), depend on these symmetries or and/or can be generalized for use with general log-linear models.

References

Adby, P.R. & Dempster, M.A.H. (1974). Introduction to optimization methods. London: Chapman and Hall.

Agresti, A. (1984). Analysis of ordinal categorical data. New York: Wiley.

Andersen, E.B. (1972). The numerical solution of a set of conditional estimation equations. Journal of the Royal Statistical Society B, 34, 42-54.

Andersen, E.B. (1973). Conditional inference and multiple choice questionnaires. British Journal of Mathematical and Statistical Psychology, 26, 31-44.

Andersen, E.B. (1980). Discrete statistical models with social science applications. Amsterdam: North Holland.

Baker, R.J., & Nelder, J.A. (1978). The GLIM system: Generalized linear interactive modeling. Oxford: The Numerical Algorithms Group.

Bock, R.D. (1975). Multivariate statistical methods in behavioral research. New York: McGraw Hill.

Cressie, N., & Holland, P.W. (1983). Characterizing the manifest probabilities of latent trait models. Psychometrika, 48, 129-142.

Deming, W.E., & Stephan, F.F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. Annals of Mathematical Statistics, 11, 427-444.

Dongarra, J.J., Bunch, J.R., Moler, C.B., and Stewart, G.W. (1979) LINPACK users' guide. Philadelphia, PA: SIAM.

Duncan, O.D. (1984). Rasch measurement: Further examples and discussion. In C.F. Turner & E. Martin (Eds.), Surveying subjective phenomena, Vol. 2 (pp. 367-403). New York: Russell Sage Foundation.

Duncan, O.D., & Stenbeck, M. (1987). Are likert scales unidimensional? Social Science Research, 16, 245-259.

Fischer, G.H. (1974). Einführung in die Theorie psychologischer Tests (Introduction to the theory of psychological tests). Bern: Huber (In German).

Formann, A.K. (1986). A note on the computation of the second-order derivatives of the elementary symmetric functions in Rasch models. Psychometrika, 51, 335-339.

Gill, P.E., Murray, W., & Wright, M.H. (1991). Numerical linear algebra and optimization. Redwood City: Addison Wesley.

Glas, C.A.W., (1989). Estimation and testing Rasch models. Unpublished Doctoral Dissertation, University of Twente. Enschede the Netherlands. (ISBN 90-9003078-6)

Goodman, L.A., & Fay, R. (1974). ECTA program, description for users. Chicago: Department of Statistics University of Chicago.

Gustafsson, J.E. (1977). The Rasch model for dichotomous items: Theory, applications and a computer program. Reports from the Institute of Education, University of Göteborg, nr. 85.

Gustafsson, J.E. (1980). A solution of the conditional estimation problem for long tests in the Rasch model for dichotomous items. Educational and Psychological Measurement, 40, 377-385.

Haberman, S.J. (1979). Analysis of qualitative data: New developments, Vol. 2. New York: Academic Press.

Holland, P.W. & Thayer, D.T. (1987). Notes on the use of log-linear models for fitting discrete probability distributions. (Technical Report No. 87-79). Princeton, NJ: Educational Testing Service.

Kelderman, H. (1984). Loglinear Rasch model tests. Psychometrika, 49, 223-245.

Kelderman, H. (1989a). Item bias detection using loglinear IRT. Psychometrika, 54, 681-698.

Kelderman, H. (1989b). Loglinear multidimensional IRT models for polytomously scored items. paper read at the Fifth International Objective Measurement Workshop, Berkeley, March 26.(ERIC document Reproduction Service No. ED 308 238))

Kelderman, H. (1991). Estimating and testing a multidimensional Rasch model for partial credit scoring. paper read at the Annual Meeting of the American Educational Reseach Association, Chicago Ill, April 5. (In review)

Kelderman, H., & Steen, R. (1988). LOGIMO: Loglinear IRT Modeling [Program Manual]. Enschede, The Netherlands: University of Twente.

Masters, G.N. (1982). A Rasch model for partial credit scoring. Psychometrika, 47, 149-174.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests. Copenhagen: Paedagogiske Institut.

Rasch, G. (1961). On general laws and the meaning of measurement in psychology. Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, 5, 321-333.

Rasch, G. (1980). Probabilistic models for some intelligence and attainment tests. Chicago: The University of Chicago Press.

Scheiblechner, H. (1971). A simple algorithm for CML-parameter estimation in Rasch' probabilistic measurement model with two or more categories of answers. Research Bulletin Nr. 5/71, Psychological Institute, University of Vienna, Austria.

SPSS, Inc. (1988). SPSS User's Guide (2rd Edition). Chicago, IL: SPSS, Inc.

Tjur, T. (1982). A connection between Rasch's item analysis model and a multiplicative Poisson model. Scandinavian Journal of Statistics, 9, 23-30.

Verhelst, N.D., Glas, C.A.W., Sluis, A. van der. (1984). Estimation problems in the Rasch model: The basic symmetric functions. Computational Statistics Quarterly, 1, 245-262.

Author Notes

The author thanks Wim J. van der Linden, Gideon J. Mellenbergh and Namburi S. Raju for their valuable comments and suggestions. Requests for reprints should be sent to: University of Twente, Bibliotheek, P.O. Box 217, 7500 AE Enschede, The Netherlands.

Table 1

Number of Multiplications and Summations Required by Summing over all Cells and the MBV Method to Calculate the Sumscore Marginal

Number of Items	Summing all cells		MBV Method	
	x	+	x	+
5	192	26	40	10
6	448	57	54	15
7	1024	120	70	21
8	2304	247	88	28
9	5120	502	108	36
10	11264	1013	130	45
11	24576	2036	154	55
12	53248	4083	180	66
13	114688	8178	208	78
14	245760	16369	238	91
15	524288	32752	270	105
16	114112	65519	304	120
17	2359296	131054	340	136
18	4980736	262125	378	153
19	10485760	524268	418	171
20	22020096	1048555	460	190

Table 2

Real and Estimated item Difficulties for Simulated Data (N=10,000)

	Item				
	1	2	3	4	5
Real	.858	-1.512	-0.173	-1.040	1.137
LOGIMO	.858*	-1.517	-0.214	-1.069	1.161
PML	.858*	-1.517	-0.215	-1.069	1.161
	6	7	8	9	10
Real	1.354	1.690	0.577	-1.270	-0.155
LOGIMO	1.318	1.636	0.618	-1.350	-0.154
PML	1.318	1.636	0.618	-1.349	-0.153
	11	12	13	14	15
Real	1.302	1.352	-0.823	-0.883	-1.754
LOGIMO	1.243	1.282	-0.858	0.871	-1.801
PML	1.244	1.284	-0.857	0.871	-1.801
	16	17	18	19	20
Real	-0.026	0.221	0.517	-0.460	1.658
LOGIMO	-0.038	0.183	0.502	-0.506	1.654
PML	-0.038	0.183	0.502	-0.507	1.653

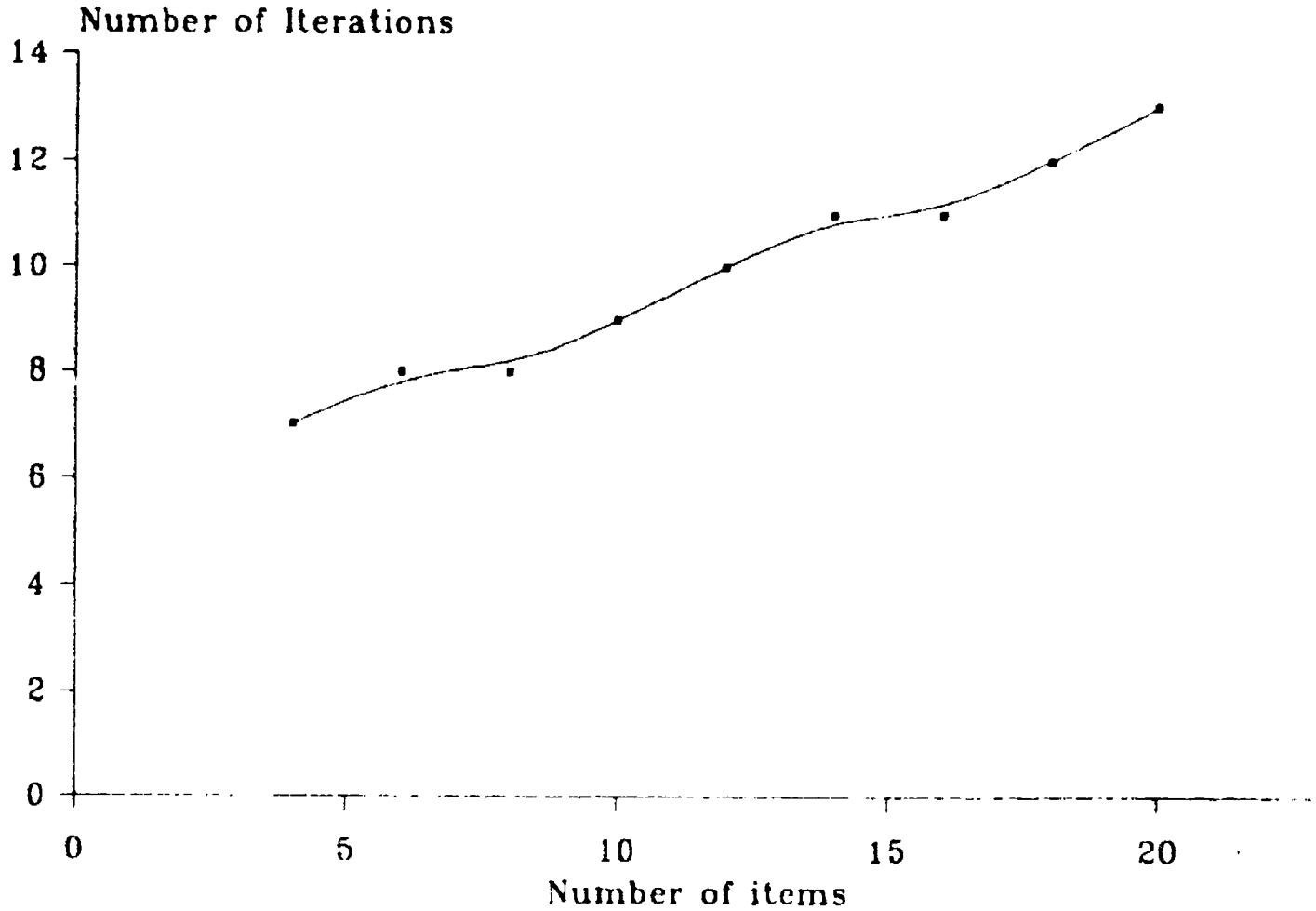
*) The estimated parameter of the first item was set equal to the real parameter value to fix the scale

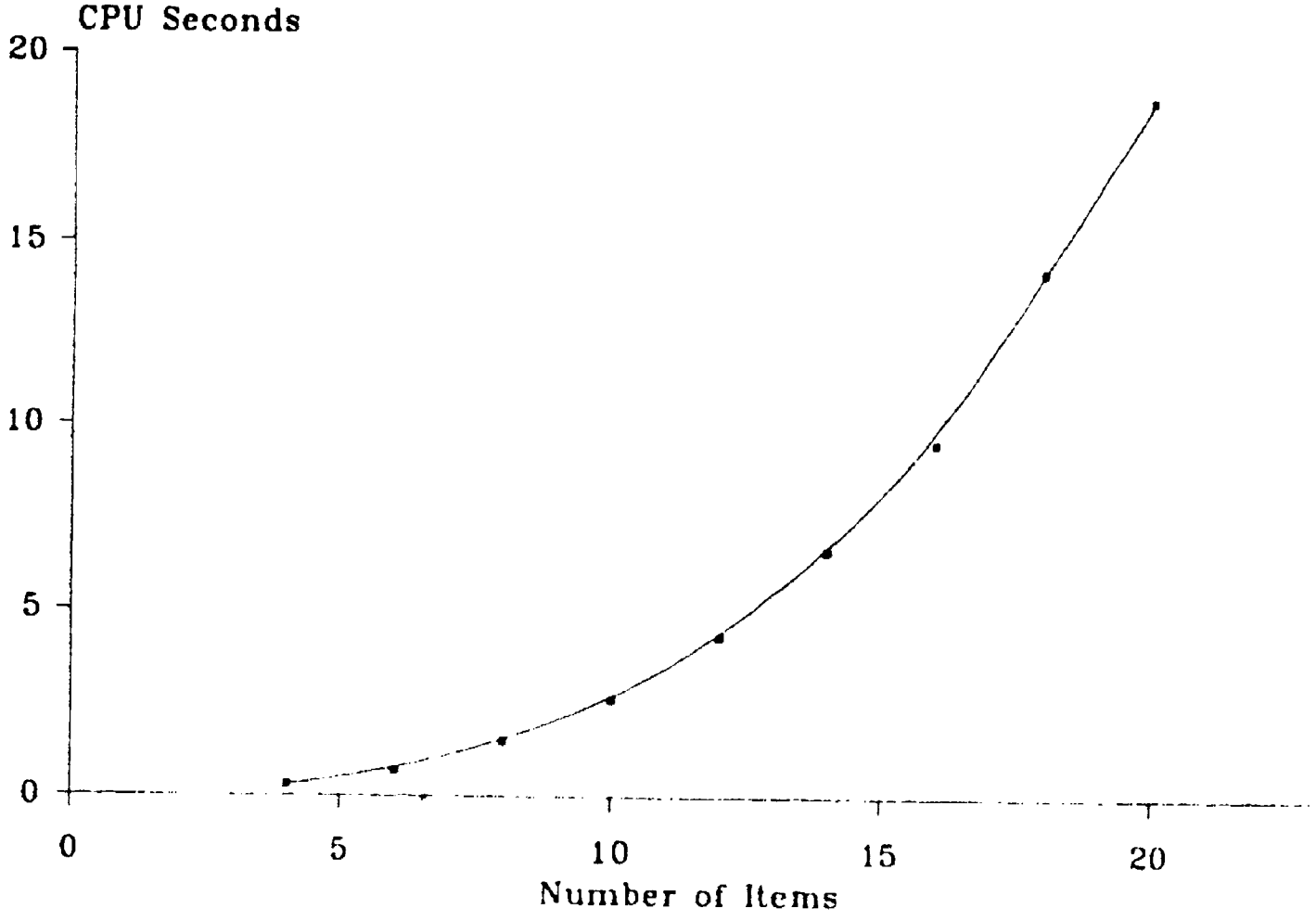
Figure Captions

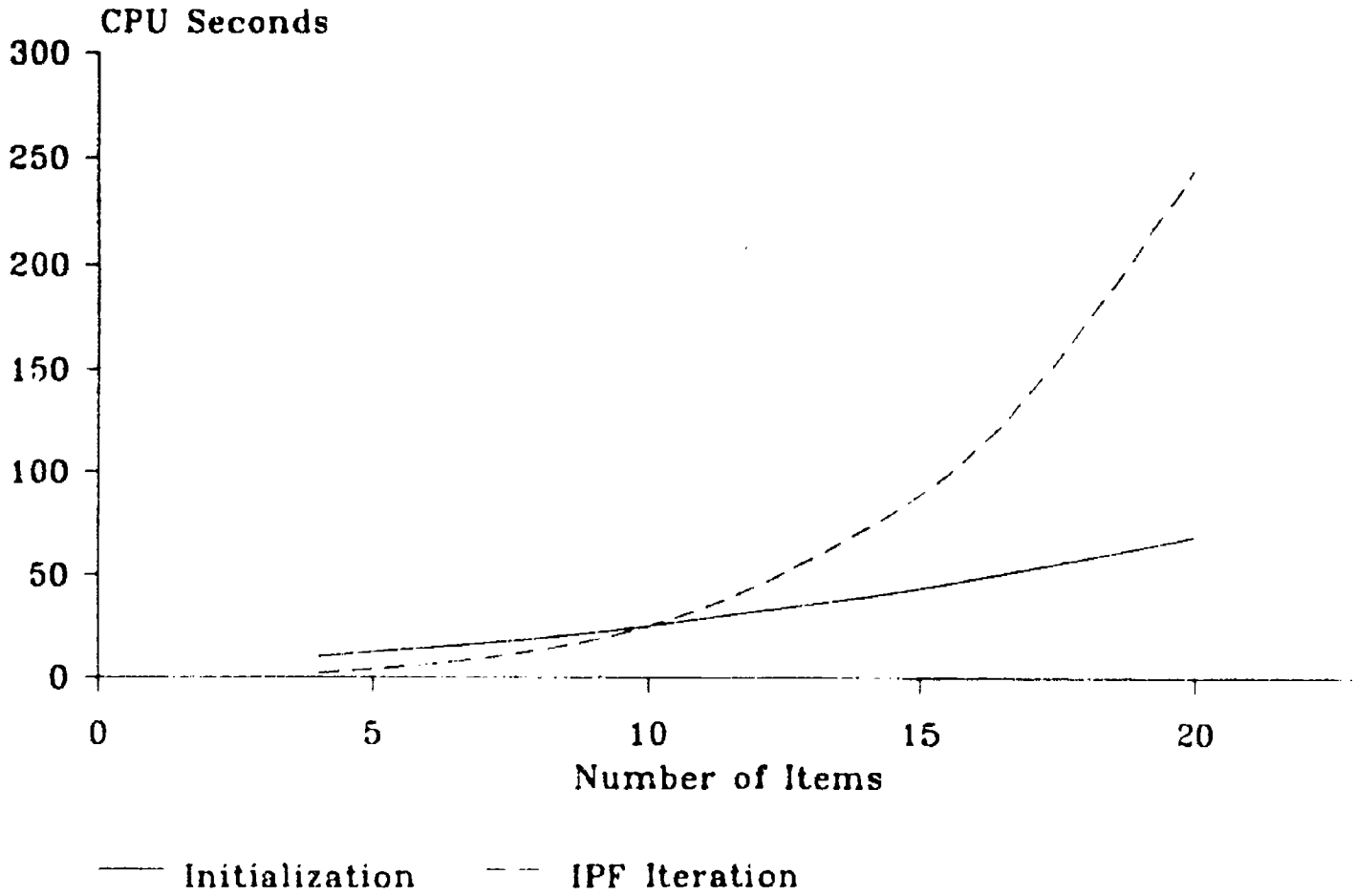
Figure 1. Growth of the Number of IPF Iterations with the Number of Items in Model 2.

Figure 2. Growth of CPU Time per Iteration with the Number of Items in Model 2.

Figure 3. Growth of CPU Time for Initialization and IPF Iterations with the Number of Items in Model 2.


Number of Iterations
14
12
10
8
6
4
2
0
0
5
10
15
20
Number of items


CPU Seconds
20
15
10
5
0
0
5
10
15
20
Number of Items


CPU Seconds
300
250
200
150
100
50
0
0
5
10
15
20
Number of Items
Initialization
IPF Iteration

Symbol List

i	a_{ij}	$\mathbf{p}$
I	b_{jk}	$\mathbf{D}\xi$
j	c_{kl}	$\mathbf{f}$
J	μ	$\mathbf{g}$
k	α_i	$\mathbf{D}^T$
K	β_j	
l	γ_k	
L	δ_l	
m	$(\alpha\beta)_{ij}$	
m_1	$(\beta\gamma)_{jk}$	
m_2	$(\gamma\delta)_{kl}$	
t	ξ	
T	$\xi^{(old)}$	
t_1	$\xi^{(new)}$	
t_2	$v_1(i)$	
P_{ijkl}	$v_2(j)$	
$P_{ijkl\|m}$	$v_3(k)$	
P_{ijklmo}	$v_4(l)$	
f_{ijkl}	$w_1(i)$	
F_{ijkl}	$w_2(j)$	
f_{ij++}	$w_3(k)$	
F_{ij++}	$w_4(l)$	
N	$\mathbf{g}^{(old)}$	
a_i	$\mathbf{g}^{(new)}$	
b_j	$\mathbf{H}^{(old)}$	
c_k	$\mathbf{H}^{(new)}$	
d_l	log	

Titles of recent Research Reports from the Division of Educational Measurement and Data Analysis. University of Twente, Enschede, The Netherlands.

RR-91-1 H. Kelderman, *Computing Maximum Likelihood Estimates of Loglinear Models from Marginal Sums with Special Attention to Loglinear Item Response Theory*

RR-90-8 M.P.F. Berger & D.L. Knol, *On the Assessment of Dimensionality in Multidimensional Item Response Theory Models*

RR-90-7 E. Boekkooi-Timminga, *A Method for Designing IRT-based Item Banks*

RR-90-6 J.J. Adema, *The Construction of Weakly Parallel Tests by Mathematical Programming*

RR-90-5 J.J. Adema, *A Revised Simplex Method for Test Construction Problems*

RR-90-4 J.J. Adema, *Methods and Models for the Construction of Weakly Parallel Tests*

RR-90-2 H. Tobi, *Item Response Theory at subject- and group-level*

RR-90-1 P. Westers & H. Kelderman, *Differential item functioning in multiple choice items*

Research Reports can be obtained at costs from Bibliotheek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

department of
EDUCATION

A publication by
the Department of Education
of the University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands