DOCUMENT RESUME

ED 389 750 TM 024 383

AUTHOR Akkermans, Wies M. W.
TITLE Monte Carlo Estimation of the Conditional Rasch Model. Research Report 94-09.
INSTITUTION Twente Univ., Enschede (Netherlands). Faculty of Educational Science and Technology.
SPONS AGENCY Netherlands Organization for Scientific Research.
PUB DATE Nov 94
NOTE 41p.; Additional grant funds received from the Dr. Catharina van Tussenbroek funds.
AVAILABLE FROM Bibliotheek, Faculty of Educational Science and Technology, 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 *Estimation (Mathematics); Foreign Countries; *Markov Processes; *Maximum Likelihood Statistics; *Monte Carlo Methods; *Statistical Distributions
IDENTIFIERS *Conditionals; Item Parameters; Person Parameters; *Rasch Model

ABSTRACT

In order to obtain conditional maximum likelihood estimates, the so-called conditioning estimates have to be calculated. In this paper a method is examined that does not calculate these constants exactly, but approximates them using Monte Carlo Markov Chains. As an example, the method is applied to the conditional estimation of both item and person parameters in the Rasch model. The key idea for this approach was developed by C. J. Geyer and E. A. Thompson (1992), who showed that, in the exponential family, a quantity that is proportional to the conditioning constant can be expressed as an expectation with respect to a certain distribution. Simulating from this distribution, an estimate of the proportional quantity can be obtained as the observed sample mean. Inserting this estimate into the conditional likelihood then allows one to maximize the approximate likelihood, as the proportionality constant does not depend on the parameters to be estimated. (Contains 5 tables, 1 figure, and 11 references.) (Author/SLD)

Monte Carlo Estimation of the Conditional Rasch Model

Research Report 94-09

Wies M.W. Akkermans

faculty of
EDUCATIONAL SCIENCE AND TECHNOLOGY

Department of Educational Measurement and Data Analysis

University of Twen

Monte Carlo estimation of the conditional Rasch Model

Wies Akkermans

A Monte Carlo method for CML estimation in the Rasch Model, Wies Akkermans - Enschede: University of Twente, Faculty of Educational Science and Technology, November 1994. - 35 pages.

Abstract

In order to obtain conditional maximum likelihood estimates, the so-called conditioning constants have to be calculated. In this paper a method is examined that does not calculate these constants exactly, but approximates them using Monte Carlo Markov Chains. As an example, the method is applied to the conditional estimation of both item and person parameters in the Rasch model.

Key words: CML estimation, Monte Carlo methods, Markov Chains, Rasch Model.

Introduction

In this paper, an alternative calculation method is examined for conditional maximum likelihood estimation (CML) in item response models that belong to the exponential family. In order to obtain CML estimates, the conditioning constants have to be calculated. These constants may be difficult to compute. The method examined provides an alternative to calculating the constants exactly: they are approximated using Monte Carlo methods. The key idea for this approximation was developed by Geyer and Thompson (1992). They showed that in the exponential family a quantity which is proportional to the conditioning constant can be expressed as an expectation with respect to a certain distribution. Upon simulating from this distribution, an estimate of the proportional quantity can therefore be obtained as the observed sample mean. Inserting this estimate into the conditional likelihood then allows one to maximise the approximate likelihood, as the proportionality constant does not depend upon the parameters to be estimated. In the first section below the method will be explained in some detail. The next section will consist of a description of the simulation process: as the distributions from which to simulate may be rather complex, Monte Carlo Markov Chain methods, such as Hastings or Gibbs sampling may be necessary. The resulting estimation equations will be examined more closely in section 3.

As an example, the method discussed will be applied to the conditional estimation of both item and person parameters in the Rasch Model (Rasch 1960). In section 4 estimates obtained using the above method will be compared to exact CML estimates. The results seem very acceptable.

A Monte Carlo method for CML estimation

In this section it will be explained how a method developed by Geyer and Thompson (1992) can be applied to conditional maximum likelihood estimation in item response models. The method will be applied to the Rasch model. In order to make the theory to follow more understandable, the section will therefore start with a description of the conditional Rasch model.

The conditional Rasch Model

Consider a test consisting of M items. Let $X_j = x$, with $x = 0, 1$ be a response to item j, and let the variable $\mathbf{X} = \mathbf{x}$ be the M-vector of responses to the entire test. The total score T is defined as

$$T = \sum_{j=1}^{M} X_j.$$

Then for the Rasch model the conditional likelihood for one observation, i.e. response pattern, looks like

$$\Pr(\mathbf{X} \mid T; \theta, \boldsymbol{\delta}) = \frac{\exp\left(-\sum_j X_j \delta_j\right)}{\sum_{\mathbf{x}:t(\mathbf{x})=t} \exp\left(-\sum_j X_j \delta_j\right)}$$

where $\boldsymbol{\delta}$ is a vector of difficulty parameters for the j items, and θ denotes the latent ability. The summation in the denominator is over all possible response patterns with the same total score $T = t$. These denominators, there is one for each value of T, are known as *elementary symmetric functions* and they will be denoted in this paper by $\gamma_t(\boldsymbol{\delta})$:

$$\gamma_t(\boldsymbol{\delta}) = \sum_{\mathbf{x}\ t(\mathbf{x})=t} \exp\left(-\sum_j X_j \delta_j\right).$$

With this notation the above formula can be rewritten as

$$\Pr(\mathbf{X} \mid T; \theta, \boldsymbol{\delta}) = \frac{\exp\left(-\sum_j X_j \delta_j\right)}{\gamma_t(\boldsymbol{\delta})}.$$

The corresponding log likelihood is given by

$$\log L(\boldsymbol{\delta} \mid T) = -\textstyle\sum_j X_j \delta_j - \log \gamma_t(\boldsymbol{\delta})$$

so that the log-likelihood for the whole sample becomes

$$\log L(\boldsymbol{\delta} \mid \mathbf{T}) = -\textstyle\sum_j \delta_j S_j - \sum_t N_t \log \gamma_t(\boldsymbol{\delta})$$

where N_t denotes the number of persons in the sample with $T = t$, S_j is the item total and $\mathbf{T}$ is the column vector of observed total scores. The solution equations are obtained upon setting the partial derivatives of this equation with respect to the δ_j's equal to zero. These partial derivatives are given by

$$\frac{\partial \log L}{\partial \delta_k} = -S_k + \sum_t N_t \frac{\gamma_{t-1}^{(k)}(\boldsymbol{\delta}) \exp(-\delta_k)}{\gamma_t(\boldsymbol{\delta})}$$

in which the numerator is equal to $\partial \gamma_t(\boldsymbol{\delta}) / \partial \delta_k$, with $\gamma_{t-1}^{(k)}(\boldsymbol{\delta})$ a gamma function for the set of items not containing item k. For example, if there were 3 items, we would have

$$\begin{aligned} \gamma_2 &= \exp(-\delta_1 - \delta_2) + \exp(-\delta_1 - \delta_3) + \exp(-\delta_2 - \delta_3), \\ \gamma_2^{(1)} &= \exp(-\delta_2 - \delta_3). \end{aligned}$$

For the Rasch model, recursive formulae are available for calculating $\gamma_t(\boldsymbol{\delta})$; but for other IRT models this is not always the case.

Approximating $\gamma_t(\boldsymbol{\delta})$: the Geyer and Thompson method

Instead of calculating the gamma functions it is possible to approximate them with the help of Monte Carlo Markov Chain (MCMC) methods, using the following idea developed by Geyer and Thompson (1992). Let the probability

density function for one observation in the conditional formulation of the Rasch Model be known as $f_t(\mathbf{X};\delta)$; then the corresponding conditional likelihood of δ as a function of $\mathbf{X}$ can be written as $f_t(\delta;\mathbf{X})$. Note that in this notation the conditioning variable T has moved from behind the bar to a subscript on f. Omitting the dependence of the likelihood on $\mathbf{X}$ we therefore have

$$f_t(\delta) = \Pr(\mathbf{X} \mid T;\theta,\delta) = \frac{\exp(-\sum_j X_j\delta_j)}{\gamma_t(\delta)}.$$

Now if ψ were another set of parameters, then $f_t(\psi)$ would be equal to

$$f_t(\psi) = \Pr(\mathbf{X} \mid T;\theta,\psi) = \frac{\exp(-\sum_j X_j\psi_j)}{\gamma_t(\psi)}.$$

Trivially, using the definition of $\gamma_t(\delta)$ and multiplying by one,

$$\gamma_t(\delta) = \sum_{\mathbf{x}:t(\mathbf{x})=t} \exp(-\textstyle\sum_j X_j\delta_j)\dfrac{\gamma_t(\psi)}{\exp(-\sum_j X_j\psi_j)}\dfrac{\exp(-\sum_j X_j\psi_j)}{\gamma_t(\psi)}$$

so that, moving one $\gamma_t(\psi)$ from the right to the left hand side of the equals sign and using the definition formula of $f_t(\psi)$

$$\begin{aligned}\frac{\gamma_t(\delta)}{\gamma_t(\psi)} &= \sum_{\mathbf{x}:t(\mathbf{x})=t} \exp\left\{-\textstyle\sum_j X_j(\delta_j-\psi_j)\right\}\dfrac{\exp(-\sum_j X_j\psi_j)}{\gamma_t(\psi)} \\ &= \sum_{\mathbf{x}\ t(\mathbf{x})=t} \left[\exp\left\{-\textstyle\sum_j X_j(\delta_j-\psi_j)\right\}\right] f_t(\psi) \\ &= E_\psi\left[\exp\left\{-\textstyle\sum_j X_j(\delta_j-\psi_j)\right\}\right].\end{aligned}$$

In other words, if we define

$$d_t(\delta) = \frac{\gamma_t(\delta)}{\gamma_t(\psi)}$$

then, if we were able to simulate a random sample of B 'observations' from $f_t(\psi) = f_t(\mathbf{X};\psi)$ we could estimate all $d_t(\delta)$'s by the sample means:

$$\hat{d}_t(\delta) = \frac{1}{B}\sum_{b=1}^{B} \exp\left\{-\textstyle\sum_j X_{tbj}(\delta_j-\psi_j)\right\}$$

for any value of δ. Note that the first subscript on X, the t, is there to indicate that every simulated response vector $\mathbf{X}$ belongs to a set having common total score T. Defining $\log L^*$ to be

$$\log L^*(\delta \mid \mathbf{T}; \psi) = -\sum_j \delta_j S_j - \sum_t N_t \log d_t(\delta),$$

note that

$$\begin{aligned}
\log L^*(\delta \mid \mathbf{T}; \psi) &= -\sum_j \delta_j S_j - \sum_t N_t \log d_t(\delta) \\
&= -\sum_j \delta_j S_j - \sum_t N_t \log \frac{\gamma_t(\delta)}{\gamma_t(\psi)} \\
&= -\sum_j \delta_j S_j - \sum_t N_t \log \gamma_t(\delta) + \sum_t N_t \gamma_t(\psi) \\
&= \log L(\delta \mid \mathbf{T}) + \sum_t N_t \gamma_t(\psi)
\end{aligned}$$

attains its maximum for the same value of δ as does $\log L(\delta \mid \mathbf{T})$. So we now can substitute $\hat{d}_t(\delta)$ for $d_t(\delta)$ and maximise the resulting expression

$$\begin{aligned}
\log L^*(\delta \mid \mathbf{T}; \psi) &\approx -\sum_j \delta_j S_j - \sum_t N_t \log \hat{d}_t(\delta) \\
&= -\sum_j \delta_j S_j - \sum_t N_t \log \tfrac{1}{B} \sum_{b=1}^{B} \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}
\end{aligned}$$

with respect to δ to get an approximate solution to the original likelihood equation.

However, the density $f_t(\mathbf{X}; \psi)$ depends on a similar constant as does $f_t(\mathbf{X}; \delta)$, i.e. on $\gamma_t(\psi)$, and this constant is still unknown, or at least difficult to calculate; therefore it is not possible to simulate from $f_t(\mathbf{X}; \psi)$ in any direct way. The solution to this can be found in the use of Markov Chains and the Hastings algorithm, which will be described below.

Simulation of response patterns: the Hastings algorithm

A Markov Chain is a sequence of realisations of a random variable X with the property that

$$\Pr(X_k = x_k \mid X_1 = x_1, \ldots, X_{k-1} = x_{k-1}) = \Pr(X_k = x_k \mid X_{k-1} = x_{k-1}).$$

Here the subscript k is used to denote the ordering of the sequence in time; and X either may or may not be a vector valued variable; in this section I will not use boldface to distinguish between the two. In the Markov Chain context the value of X is often called the 'state' of X. The probabilities of going from one state to another in a Markov Chain can be represented in a matrix P, having as entries $p_{ij} = \Pr(X_i = i \mid X_{i-1} = j)$; hence the rows of P add up to one. The Markov Chain is said to be irreducible if it is possible to get from any state to any other state in a finite number of transitions. The states of irreducible Markov Chains on finite sets of values can be shown to follow a unique limiting or ergodic distribution; denoting this (discrete) distribution by π, it is given as the solution to

$$\pi P = \pi$$

i.e. if the transitions are made according to P, then for large N, $\Pr(X_{k+N} = i \mid X_k = j) = \pi_i$, independent of the value of X_k (see e.g. Proth & Hillion 1990).

For our calculations we need simulations from $f_t(\mathbf{X}; \psi)$. Now imagine it would be possible to find a transition matrix P that has its limiting distribution π equal to $f_t(\mathbf{X}; \psi)$. Then we could start from any initial response pattern $\mathbf{X}_k$, and using the proper row with conditional probabilities from matrix P we could then simulate a new state for the response pattern, say $\mathbf{X}_{k+1}$ (the subscript still indicates ordering in time), and so on; until, after N successive state changes, we would have $\mathbf{X}_{k+N} \sim \pi = f_t(\mathbf{X}; \psi)$. Using this Monte Carlo Markov Chain sampling scheme gives us an estimator $\hat{d}_t(\delta)$ that is asymptotically normally

distributed and approaches $d_t(\delta)$ in mean square as the number of Monte Carlo samples $B \rightarrow \infty$ (Hastings 1970).

Of course, the problem is to find P from $\pi P = \pi$. In general, this is impossible. However, Metropolis et al. (1953) and Hastings (1970) found a way to sample from π without actually knowing P. Let X be the present state of the sequence, and X' an alternative state. Then their algorithm is as follows:

1. Define any convenient transition matrix Q
2. Propose a new state, say X', for the variable X, in our case the response pattern $\mathbf{X}$, according to the probabilities in the relevant row of Q
3. Define
$$\alpha(X', X) = \min\left\{1, \frac{\pi(X')Q(X', X)}{\pi(X)Q(X, X')}\right\}$$
4. Accept the proposed state with probability α.

The algorithm can be proved to work, i.e. to generate a sequence with the desired ergodic distribution, using the 'detailed balance' lemma. This lemma states the following: if for irreducible P it holds that $\pi_i p_{ij} = \pi_j p_{ji}$ then π is the limiting distribution for P. Substituting q_{ij} for p_{ij}, this condition can be easily checked to hold in the above algorithm.

Recalling that in our case $\pi(x)$ represents $f_t(\mathbf{X}; \psi) = \exp(-\sum_j X_j \psi_j)/\gamma_t(\psi)$ the denominators of $\pi(X)$ and of $\pi(X')$ in step 3 are equal. Thus the cleverness of the algorithm lies in the fact that in calculating $\frac{\pi(X')Q(X', X)}{\pi(X)Q(X, X')}$ there is no need to calculate these denominators as they will cancel.

The algorithm as presented above is known as the Metropolis algorithm; when a symmetric matrix Q is used it is called the Hastings algorithm. In that case also the terms involving Q will cancel and α reduces to $\alpha(X, X') = \min\left\{1, \frac{\pi(X')}{\pi(X)}\right\}$. Another improvement, in the case of a vector valued variable X, would be not to propose a new state X' ad random, but to consider changing

only one element of X at a time. If the variables in X are independent of each other, or depend only on a few other variables, many additional factors in $\frac{\pi(X')}{\pi(X)}$ will cancel. In this latter case, the generation of *one* new simulation consists of sequentially considering a new value for each of the variables in X in turn. This is also called a 'full scan'; therefore, after one full scan the next simulated value for X has been obtained.

Recapitulation:

1. We want to do conditional maximum likelihood estimation for the parameters in the Rasch model. This means we need to know the values of the conditioning constants $\gamma_t(\cdot)$.

2. We want to estimate these constants by the method proposed by Geyer and Thompson. That means we need artificial data drawn from $f_t(\mathbf{X};\psi)$, where ψ is an arbitrary point in the parameter space.

3. To draw the samples from $f_t(\mathbf{X};\psi)$ we want to use the Hastings algorithm.

Assuming we have actually drawn the sample, we then proceed to maximise the following equation, in which $d_t(\delta)$ has been expanded in full:

$$\log L^*(\delta \mid \mathbf{T};\psi) = -\sum_j \delta_j S_j - \sum_t N_t \log\left[\frac{1}{B}\sum_{b=1}^{B} \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}\right]$$

with partial derivatives

$$\frac{\partial \log L^*}{\partial \delta_k} = -S_k + \sum_t N_t \frac{\sum_b X_{tbk} \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}{\sum_b \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}$$

In the next section the resulting estimation equations will be examined more closely.

Estimation

The two main tasks to be performed for the implementation of the estimation method proposed in section 1 are the simulation of the Monte Carlo data, and the actual estimation. In this section I will comment on several aspects of the estimation equations. In particular I will compare the equations for the Monte Carlo estimation to those for ordinary CML estimation. But I will start with another description of the algorithm for generating response vectors.

Generation of Monte Carlo response patterns

Starting from the current realisation of a response vector $\mathbf{X}$, the next realisation will be obtained after one 'full scan'. Recall that we are simulating from a distribution conditional on total score. Therefore we cannot change only one value X_j at a time: a new proposal state $\mathbf{X}'$ has to be obtained by interchanging the position of two different values in the response vector. A full scan then consists of the following steps (note that the subscript on X again denotes items instead of temporal ordering):

1. $i = 1$
2. if $X_i = x$, with $x = 0$ or 1, then randomly choose one of the items, say item j, with $X_j = 1 - x$
3. the proposed state $\mathbf{X}'$ is the response vector with $X_i = 1 - x$ and $X_j = x$
4. accept this proposed state with probability $\alpha = min\left\{1, \frac{f_t(\mathbf{X}';\psi)}{f_t(\mathbf{X};\psi)}\right\}$
5. if $i \leq M$, with M the number of items, then $i = i + 1$, and go back to 2; else stop: the next simulated response vector has been obtained.

The newly arrived at simulated response vector will in turn serve as input for the next full scan etc. There may be need for a 'burn-in' period in the very

beginning, in order to allow the algorithm to move away from a possibly badly chosen starting configuration **X**.

Of course, the response vectors simulated in this way will not be completely independent. The autocorrelation could be reduced by inserting 2 or more scans between successive Monte Carlo simulations. But as its only influence will be on the variance of $\dot{d}_t(\delta)$, it is equally well possible to use all the generated response patterns and to go on generating them until the variance is acceptable.

Having obtained a 'fairly large number', B, of simulated response vectors it is possible to start estimating $\hat{\delta}$.

Starting values

As starting values, i.e. first choice of ψ, the well known Gustafsson (1979) starting values are used:

$$\delta_j^{(0)} = - \frac{S_j - \bar{S}_j}{\sum_{t=1}^{M-1} N_t \frac{t(M-t)}{M(M-1)}}$$

in which S_j is the item total $\sum_{i=1}^{N} X_{ij}$ and $\bar{S}_j$ the average of the item totals: $\sum_j S_j/M$.

If $\dot{\delta}$ is far away from ψ, the approximation of log L by $\log L^*$ might not be too well at $\delta = \dot{\delta}$. Therefore it is probably wise to use the estimate $\dot{\delta}$ as a new ψ and repeat this preliminary procedure a couple of times before going on to simulate a truly large sample that will be used for estimation. Geyer and Thompson also advise to use a restriction on the maximum steplength per iteration, but in view of the good starting values available for CML estimation in the Rasch model, I found there was no need for such a restriction.

Identifiability

To begin with, the usual constraint $\sum_j \delta_j = 0$ or $\delta_1 = 0$ has to be imposed.

Next, it is well known (Ford 1957) that in order for a solution to the ordinary CML equations to exist, all persons and items with perfect scores ($T = 0$ or $T = M$, and $S_j = 0$ or $S_j = N$) have to be deleted from the sample. Meaningful estimates for these persons and or items cannot be obtained, and deleting them will not influence the estimation of the remaining parameters.

For the present estimation procedure this latter assertion remains yet to be investigated. As to the influence of perfect persons on item parameter estimation, we had

$$\log L^* = -\sum_j \delta_j S_j - \sum_t N_t \log \left[\frac{1}{B} \sum_{b=1}^{B} \exp \left\{ -\sum_j X_{tbj}(\delta_j - \psi_j) \right\} \right].$$

It is evident that the term with $T = 0$ does not contribute to the function value, as all the X_{tbj}'s are equal to zero, so we get $\log(1/B \times B) = \log 1 = 0$. Likewise, the term with $T = M$ would have no contribution. Let there be k persons with total score $T = M$, then if we would delete these persons we would have $\log L^*$ equal to

$$-\sum_j \delta_j (S_j - k) - \sum_{t, t \neq M} N_t \log \left[\frac{1}{B} \sum_{b=1}^{B} \exp \left\{ -\sum_j X_{tbj}(\delta_j - \psi_j) \right\} \right] - k \sum_j (\delta_j - \psi_j).$$

The terms with $k \sum_j \delta_j$ cancel, and although the resulting formula is not exactly equal to the original log-likelihood, the difference does not depend on the parameters to be estimated, so it will result in the same estimates. For these reasons, persons with perfect response patterns can be safely omitted from the estimation of item parameters.

Next, to the influence of perfect items on the estimation of other item parameters. It seems plausible to use $\psi_p = \infty$ or $\psi_p = -\infty$ for items with $S_p = 0$ or $S_p = N$ respectively. Thus they will generate only perfect response patterns. It is easy to show that deleting those items from the loglikelihood will not influence the estimation equations for the other items.

Rank of the system of equations

One topic clearly needs some more attention. As S_j is sufficient for δ_j, in the context of ordinary CML there will only be as many estimation equations as there are different values of S_j. The situation is different for the Monte Carlo CML equations. Here we have

$$\frac{\partial \log L}{\partial \delta_k} = -S_k + \sum_t N_t \frac{\sum_b X_{tbk} \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}{\sum_b \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}$$

and because of the Monte Carlo processes there is no guarantee that if $S_k = S_l$, then also X_{tbk} will be equal to X_{tbl}, not even if you take $\psi_k = \psi_l$. So without taking any precautions one would in this case end up with different estimators for items with the same value of the statistic. This is clearly an undesirable state of affairs. The easiest way out would seem to retain only one of the items with equal values on S_j in the analysis, but that would cause problems for the conditional Monte Carlo sampling scheme. Instead, I decided to average the estimation equations for items with equal values for S_j. This has implications for the way the equations can be written. If $S_k = S_l$, then $\hat{\delta}_k$ will have to be equal to $\hat{\delta}_l$. To begin with, I therefore take $\psi_k = \psi_l$. This will prove to be a convenient choice. Now, with $S_k = S_l$, we have

$$S_k = \sum_t N_t \frac{\sum_b X_{tbk} \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}{\sum_b \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}, \quad j = 1 \ldots M,$$

and

$$S_l = \sum_t N_t \frac{\sum_b X_{tbl} \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}{\sum_b \exp\left\{-\sum_j X_{tbj}(\delta_j - \psi_j)\right\}}, \quad j = 1 \ldots M$$

so that $S_k + S_l$ is equal to

$$\sum_t N_t \frac{\sum_b (X_{tbk} + X_{tbl}) \exp\left\{-X_{tbk}(\delta_k - \psi_k) - X_{tbl}(\delta_l - \psi_l) - \sum_{j,j\neq k,j\neq l} X_{tbj}(\delta_j - \psi_j)\right\}}{\sum_b \exp\left\{-X_{tbk}(\delta_k - \psi_k) - X_{tbl}(\delta_l - \psi_l) - \sum_{j,j\neq k,j\neq l} X_{tbj}(\delta_j - \psi_j)\right\}}.$$

If $\psi_k = \psi_l$, and if we let $Y_{tbk'}$ be equal to $X_{tbk} + X_{tbl}$ the above equation simplifies considerably. Likewise, we can do the same for all sets of items with the same value for S, so that we get a different vector of observations, say $\mathbf{Y}$, in which Y_j is equal to a sum over several X_j's. Now we can write

$$S_h M_h = \sum_t N_t \frac{\sum_b Y_{tbh} \exp\left\{-\sum_j Y_{tbj}(\delta_j - \psi_j)\right\}}{\sum_b \exp\left\{-\sum_j Y_{tbj}(\delta_j - \psi_j)\right\}}, \quad j = 1 \ldots M_a, h = 1 \ldots M_a$$

where M_a is the number of different values actually appearing for S, and M_h the number of items with $S = s_h$. The above becomes particularly relevant when estimating abilities instead of difficulties, because usually the number of persons will be much larger than the number of items. As there only are $M - 1$ different values for T, then only $M - 1$ equations will be necessary instead of N.

Estimating abilities

All the above applies equally well to estimation of $\boldsymbol{\theta}$ as to estimation of $\boldsymbol{\delta}$. The same routines can be used to maximise both sets of equations. The only correction that has to be made is that the sign of the outcomes when estimating $\boldsymbol{\theta}$ has to be reversed. It would have been fortunate if the same Monte Carlo data could be used for both estimation procedures. However, sampling under the condition of one fixed marginal (say for T), will change the other marginal, so this is not a feasible possibility.

If both $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ are estimated using CML, they still have to be positioned on to a common scale. This can be achieved by first estimating $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ separately, and then setting $\boldsymbol{\theta}^* = \boldsymbol{\theta} + k\mathbf{1}$, the latter being a vector of ones. Then optimise

the *joint* maximum likelihood for θ^* and δ with respect to k, and finally take as your estimates $\dot{\theta} + \dot{k}1$ and $\dot{\delta}$.

All function maximisations were carried out using the Fletcher-Reeves algorithm, in a a slightly modified form proposed by Polak and Ribiere (for details, see Press, Teukolsky et al 1992).

Results

This section gives some results obtained in testing the Hastings algorithm used for the simulation of response patterns. The last part of the section will compare the Monte Carlo estimates to exact CML estimates for 3 small real data sets.

The data

For some of the tests all possible response patterns have to be considered. Therefore some small data sets, preferably with known or previously estimated parameter values, were necessary. I used data provided by Thissen (1982). He reports the results of CML estimation on a 10 item memory test. This test was taken by 40 persons, 5 of which had a zero score, so there were 35 of them left for estimation. In addition he reanalyses two 5-item sections of the law school admissions test. These two subtests, which will be denoted as lsat6 and lsat7, were analysed earlier by Andersen and Madsen (1977) and by Bock and Lieberman (1970). The data represent responses of 1000 subjects drawn from a larger sample of students applying for admission to law schools at various universities in the United States. After omitting persons with perfect scores, 699 and 680 respectively remained for analysis.

Response pattern generation

In order to test the algorithm for simulation of conditional response vectors, data were generated for a 5-item test with item difficulties equal to the starting values for the lsat6. For the four non-perfect values of the total score 500 response vectors were generated from $f_t(\mathbf{X};\boldsymbol{\psi})$ using the Hastings algorithm described in section 1. As the number of different response patterns is quite small in this case ($2^5 - 2 = 30$), it was possible to calculate the theoretical conditional probabilities, i.e. the value of $f_t(\mathbf{X};\boldsymbol{\psi}) = \exp(-\sum_j X_j\delta_j)/\gamma_t(\boldsymbol{\psi})$, for each pattern and to compare the observed frequencies for the generated response patterns to the expected ones. Next, a chi-square statistic $\chi^2 = \sum(\text{fobs} - \text{fexp})^2/\text{fexp}$ was calculated for each conditional distribution $f_t(\mathbf{X};\boldsymbol{\psi})$. This process was repeated with Monte Carlo sample sizes B equal to 1000 and 2000. The results are given in table 1.

Insert table 1 about here

None of the values in this table is significant, but two remarks apply. First, it is probably not really justifiable to perform a χ^2 goodness of fit test, because the generated response patterns are not completely independent, being subsequent realisations under the Hastings algorithm. Therefore, the values should be interpreted with some care. And second, as a result of the randomness in the data simulation process, the generation of tables like table 1 is in this case itself a random process. So ideally the analysis should be repeated a large number of times, say a 1000, and the results studied. I did not do that; I only repeated it several times. Although sometimes significant χ^2 values appeared, which is to be expected, largely the pattern was as above. Therefore, although more rigourous tests still can be done, for the time being the conclude was that the response pattern generator works satisfactorily.

The estimator $\dot{d}_t(\delta)$: sequential plots

The next thing to investigate was the performance of the estimator $\dot{d}_t(\delta)$. First, some plots were constructed depicting the relationship between $\dot{d}_t(\delta)^{(b)}$ and b, where $\dot{d}_t(\delta)^{(b)}$ is the value for $\dot{d}_t(\delta)$ as calculated from the first b (out of B) Monte Carlo simulations. Then $\dot{d}_t(\delta)^{(b)}$ is plotted against b, so that a kind of time series plot results. Now there are several factors which will influence the estimate $\dot{d}_t(\delta)^{(b)}$. Firstly, of course, there is the number of simulations b upon which it is based. Hopefully, with increasing b, the estimate will become stable, i.e. converge to a certain value. In other words, $\dot{d}_t(\delta)^{(b+j)} - \dot{d}_t(\delta)^{(b)}$ should go to zero for large b and any value of j. Next, it seems likely that the goodness of the estimate will be influenced by the shape of the distribution of $f_t(\mathbf{X};\psi)$, as the empirical pmf of a sample from a 'regularly shaped' distribution in general will more closely resemble the shape of its parent than a sample of the same size from an irregularly shaped distribution. Thirdly, recall that $\dot{d}_t(\delta)$ estimates $\gamma_t(\delta)/\gamma_t(\psi)$, and that the estimate will probably be better for δ close to ψ, and worse for δ a large distance from ψ. So the distance from δ to ψ is a third factor that might influence the 'goodness' of the estimate.

Ideally, what I should have done is use a fixed value for δ, then generate response vectors at various points ψ, and finally examine the behaviour of $\dot{d}_t(\delta)^{(b)}$ for each value of ψ. However, I decided to work the other way around: use a fixed ψ and examine the behaviour of $\dot{d}_t(\delta)^{(b)}$ for various possible values of δ. The advantage of course is that now there is need to simulate only one data set instead of several ones. I wanted to look at estimates for δ close to ψ, far away from ψ, and in between. Arbitrarily, I decided that 'close' would mean $\mid \delta_j - \psi_j \mid < .1,\ \forall j$. As it would be rather artificial to have all $\delta_j - \psi_j$ equal to each other, a δ close to ψ was generated by sampling δ_j from a uniform

distribution on $(\psi_j - .1,\ \psi_j + .1)$ for each j. In a similar way δ_j's were sampled from $U(\psi_j - 3,\ \psi_j + 3)$ to get a faraway point, and from $U(\psi_j - 1,\ \psi_j + 1)$ to get an intermediate point. Now for the faraway point not all the δ_j's will in fact be far from the respective ψ_j's, but this choice will prove interesting.

Having generated Monte Carlo data at ψ and having found an arbitrary value for δ in the required distance range, the sequential estimates for $\dot{d}_t(\delta)^{(b)}$ were then calculated for b ranging from 1 to 2000. Informally repeating this several times with different values for the seed, it appeared that sometimes the results were as expected, and sometimes they were not.

Insert Figure 1 about here

To begin with an example of a nice result, figure 1 displays the plots for three arbitrarily chosen values of t for a ten item data set using as ψ the starting values for the memory data, for the largest distance of 3. Plots for $B = 500$ and $B = 2000$ are placed next to each other; the 500 simulations are the first 500 from the 2000. After 500 simulations the estimates do not seem to have reached their equilibrium completely. After 2000 simulations the lines look smooth, but note that for example the line for $t = 7$ still seems to be increasing; also, the line for $t = 5$ still shows occasional small wiggles. However, bearing in mind that these are plots for δ a large distance away from ψ, I think these results are quite nice. It would be hard to say whether they are representative, though. Most plots for $d \approx .1$ (d indicating the distance) were better than these ones, some were similar; and also there were plots for $d \approx 3$ t hat were worse than these ones. But on the whole I am inclined to believe that these plots are fairly average for this 10 item data set. For a 5 item data set on the other hand I found that sometimes the plots were more fluctuating, even withe $B = 2000$. A closer inspection showed that this could happen because of extreme differences in the

values for $\sum_j X_j(\delta_j - \psi_j)$. The explanation is as follows. In estimating $\hat{d}_t(\delta)$ two processes are involved: each new response pattern in first *generated* and then it is *added to* $\sum_b \exp\left\{-\sum_j X_j(\delta_j - \psi_j)\right\}$ (for convenience omitting the subscripts b and t on X_{tbj}); this means each response pattern has a probability of occurring, depending only on $f_t(\mathbf{X};\boldsymbol{\psi})$, and it has a particular value for $\sum_j X_j(\delta_j - \psi_j)$, depending on the differences $\delta_j - \psi_j$. Now problems are likely to occur, for some fixed $\boldsymbol{\delta}$, when there is a (or a few) response patterns with a very small probability of occurrence, and at the same time a comparatively large negative value for $\sum_j X_j(\delta_j - \psi_j)$; large, that is, compared to the value of $\sum_j X_j(\delta_j - \psi_j)$ for *other* $\mathbf{X}$. Then $\exp\left\{-\sum_j X_j(\delta_j - \psi_j)\right\}$ can become very large indeed; so that in our example every once in a while a value of 2097 would be added to the summation, whereas many other values with higher probabilities would be equal to .5 or .11. It would take perhaps a B of 100,000 simulations to level this out. This is why I said above that the way of generating the $\boldsymbol{\delta}$'s would prove interesting. The finding can be stated differently too: there is no problem when $\delta_j - \psi_j$ is of about the same size for all j. Then all the terms in the summation over B will be of about equal size, even for different response patterns $\mathbf{X}$. Problems *can* arise if there are one or more j for which $\delta_j - \psi_j$ is very large negative, compared to the other differences. No problems will occur in the reverse case, when there is a reasonable number of items and $\delta_j - \psi_j$ is very small for some j compared to the rest.

Having understood a possible source of erratic behaviour of the sequential plots, the question becomes: do we have to worry about it? To answer this question a simulation study was conducted which will be described in the next section. One adjustment of the statement in Geyer and Thompson however has to be made: in our case it is not the overall distance from $\boldsymbol{\delta}$ to $\boldsymbol{\psi}$ which influences the goodness of the estimator $\hat{d}_t(\boldsymbol{\delta})$ most: the spread in the distances

$\delta_j - \psi_j$, combined with the probability distribution $f_t(\mathbf{X}; \psi)$ is probably more important.

The estimator $\dot{d}_t(\delta)$: Accuracy of the approximation of $d_t(\delta)$

Sequential plots, as drawn in the previous section, can be very enlightening and instructive, especially when one happens to come across one that displays unexpected or unwanted behaviour. But for finding out something about the average behaviour of the estimator alternative means are needed. I conducted a simulation study which is algorithmically given by:

1. Take a specific value for ψ
2. Choose a distance, say $d = .1$
3. Choose a value for δ within that distance from ψ
4. Calculate $\dot{d}_t(\delta)$ for $B = 500$
5. Calculate $d_t(\delta)$ for $B = 2000$
6. Repeat steps 3-5 1000 times and calculate the average and standard deviation for $\dot{d}_t(\delta)$. Compare this with the expected value
7. Repeat steps 2-6 for distances of 1 and 3.

Of course, the trends that are so nicely visible in the sequential plots will not appear in this way: looking only at $\dot{d}_t(\delta)$ for $B = 500$ and 2000 provides one with a look at a 'fixed point in the plot' only. Moreover, calculating the mean and variance of $\dot{d}_t(\delta)$ for 1000 replications might not be very instructive in itself, as the values of $\dot{d}_t(\delta)$ should in the first place be compared to $\gamma_t(\delta)/\gamma_t(\psi)$, which is different in each of the 1000 replications because they each have a different value for δ. Therefore, denoting $\gamma_t(\delta)/\gamma_t(\psi)$ by $d_t(\delta)$, for each of the 1000

replications, the relative difference $\{\hat{d}_t(\delta) - d_t(\delta)\}/d_t(\delta)$ was calculated. This relative difference was the variable of interest in the present investigation; its mean, standard deviation, minimum and maximum are displayed in table 2 for $B = 500$ and $B = 2000$.

Insert Table 2 about here

Again, the starting values for the lsat6 data were used to provide me with a point ψ. Looking at the top half of the table first, we see that for $B = 500$ the average relative error is very small: for δ near to ψ it is .001 at most. The associated standard deviations vary from .02 to .05; and the maximum values for the minimum and maximum relative errors are equal to .09 and .10 resp. The worst values occur for $t = 4$. For the other distances the estimate behaves worst for $t = 4$ as well. In the case of a distance 1, the average error for $t = 4$ is about 14%, and for $d \approx 3$ the *average* estimate for $t = 4$ is about 1.67 times as big as what it should be. Once it even was 16 times too big. Although, in my opinion, the averages for the other values of t do seem acceptable for $d \approx .1$ and $d \approx 1$, the associated standard deviations are rather large for $d \approx 1$.

The values for $B = 2000$ are, on the whole, strikingly similar to those for $B = 500$. I can only conclude that on average it does not seem to make much difference whether you use a Monte Carlo sample size of 500 or of 2000 in estimating $d_t(\delta)$: on average the estimates are very reasonable for δ within a short distance of ψ; but standard deviations of about 5% may still occur. For δ far away from ψ the estimates are unreliable; and for δ in between I am inclined to say they are not utterly reliable either. Note, however, that for item parameters in the context of IRT a difference of 1 is quite substantial, and in practice our starting values will probably be closer to the final estimates.

A consistent pattern seems to be that the estimates are better for smaller

values of t; this is probably due to the fact that for small t the summation $\sum_j X_j(\delta_j - \psi_j)$ consists of fewer terms (many of the x_j being equal to zero), so that the differences between values of $\exp\left\{-\sum_j X_j(\delta_j - \psi_j)\right\}$ for different response patterns with the same total score cannot become very large. Also, this table has been obtained using a very small data set, consisting of only 5 items. It would be interesting to see whether the results would be similar with larger numbers of items: the sequential plots shown in the previous section were in general more irregular for a 5 item test than they were for a 10 item test. As mentioned, this might be due to the fact that the pmf of a five-item response vector is probably more irregular than the pmf of a ten-item response vector.

Comparison of MC estimates to exact CML estimates

In this section results will be presented for the estimation of item and ability parameters for the lsat and memory data. The results are for one estimation only, the procedure has not been replicated to examine the behaviour of the estimators. Table 3 contains the Monte Carlo parameter estimates for the lsat data together with the exact CML estimates.

Insert Table 3 about here

The Monte Carlo estimates have been calculated under the constraint $\delta_1 = 0$, and after the estimation process was completed, the estimates were rescaled to have a mean of 0. The fourth column in the table contains the differences between both estimation procedures. The Monte Carlo estimates are within 1 standard error from the exact values. Therefore my conclusion is that they are at least acceptable.

The ability estimates are presented in the right hand side of the same table. In this case, the so-called 'exact' estimates are no CML estimates at all: the

ability estimates are calculated treating the item parameters as known. Note that, although the estimated standard errors are much larger for the ability estimates than they were for the delta's, the Monte Carlo estimates are within the same order of distance from the 'exact' values as they were for the item parameters.

Insert Table 4 about here

Table 4 contains the estimates for the memory test. The differences between Monte Carlo and exact estimates seem to be somewhat smaller than for the lsat data. Certainly in view of the larger standard errors here (a consequence of the smaller sample size) this is very nice. Turning to the ability estimates we find that there were no subjects with total scores larger than 7. Therefore I was only able to calculate Monte Carlo estimates for θ for $T = 1$ up to $T = 7$. This is in contrast to the usual estimation method: upon assuming all the item parameters are known, it is no problem to calculate ability estimates for any value of T, whether this value actually appears in the data or not. Again the differences are very small.

Computer times and storage

The storage required for this estimation procedure is huge. A tensor of approximately size $M \times M \times B$ has to be stored, in which B is the number of Monte Carlo samples, and M is the number of items. Storage is necessary because the maximisation of every $\log L^*$ needs several iterations, in each of which the Monte Carlo data appear, together with different values for the parameters. If there are items with the same value for S, the size of the array can be reduced somewhat because of the fact that Monte Carlo responses for those items will only appear in the equations added up together (i.e. Y_j's instead of X_j's), so

they might as well be stored added up together. In the case of estimating θ this is no trivial reduction, because otherwise the size of the array would have been of order $M \times N \times B$.

For the three data sets I have examined, more than 90% of the computer time, perhaps even 99%, was spent in generating the response patterns. Once the response patterns were there, it was usually a matter of seconds, or even less, to see the actual estimates appearing on the screen.

Insert Table 5 about here

This is reflected in table 5 in which the CPU-times for the estimates reported in the previous section are given. The times are for a small UNIX machine. For the lsat data, the CPU-times for calculating ability estimates are much larger than the times for calculating item difficulties. This is because there are nearly 700 persons in the sample, so that in estimating θ for each item 700 responses have to be generated; whereas in estimating δ only 5 items are involved. Clearly some work has to be done to see if these times can be reduced. One suggestion could be to reduce the number of Monte Carlo estimation cycles, or the size of the Monte Carlo samples, especially in estimating θ. At present I use an initial sample size of 500; when the squared Euclidean distance between two consecutive sets of parameter estimates becomes less than .10, I switch to a Monte Carlo sample size of 2000 and do 2 more maximisation cycles. In estimating θ however the precision reached after only 1 cycle with $B = 500$ was hardly ever increased by subsequent cycles with $B = 2000$. Considering the fact that for the estimation of ability parameters the simulated 'response vector' is of length N, it is not very surprising that these estimates are better, for the same value of B, than those of the item parameters, which are based on a response vector of length M. So for the ability parameters a substantial reduction in the amount

of CPU-time might very well be possible. For the estimation of δ I am not so sure a very large reduction is possible (apart from improvements in the style of programming). The switch to $B = 2000$ was always made after 1 or 2 cycles with $B = 500$, but sometimes the first cycle with $B = 2000$ would then again show a squared Euclidean distance of for example .14. Again, a suitable topic for further work.

Conclusion and discussion

The estimation method examined in this paper seems to work, at least for small data sets. The approximate estimates produced are not too far away from the exact values. However, many things still need to be done; some of them I have not even started with. Here they come, not in any particular order.

To begin with, the variance of the estimator $\dot{d}_t(\delta)^{(B)}$ needs some more attention. According to Hastings (1970) this variance is equal to

$$\frac{\sigma^2}{B} \sum_{j=-B+1}^{B-1} \left(1 - \frac{|j|}{B}\right) \rho_j$$

where ρ_j is the autocorrelation for lag j. As pointed out before, the autocorrelation cannot a priori be assumed to be zero. Two methods for investigating this are, first, similar sequential plots for the estimated variance of $\dot{d}_t(\delta)^{(b)}$ as the ones for $\dot{d}_t(\delta)^{(b)}$ itself. The second method would be to calculate the variance of $\dot{d}_t(\delta)^{(B)}$ for different numbers of scans between two successive simulations; this should obviously reduce the autocorrelation.

The parameter estimates produced in section 3 should be reproduced a large number of times with different values for the seed of the random number generator, to get some additional insight into the average and standard deviation of the Monte Carlo estimates.

The variance/covariance matrix of the parameter estimates has to be calculated as well. Probably it is possible to approximate the matrix of second derivatives by a method similar to the one used to get the parameter estimates, and then to invert this matrix. But possibly this will be more involved than approximating only the parameter estimates.

The conditions under which the estimator $\dot{d}_t(\delta)$ performs well need some more attention. Especially the difference that seems to exist between the findings in this paper and the ones by Geyer and Thompson, concerning the influence of the distance from δ to ψ, could be examined further.

The estimation process needs some more attention as well, especially theoretically: can it be expected that the results of unconstrained maximisation will be similar to those using one fixed parameter? Some preliminary analyses seem to suggest that unconstrained maximisation (and later rescaling to a mean of zero) gives results similar to the ones found above, the only difference being slightly larger computer times.

All the estimations done until now were on very small data sets. It is necessary to get some indication of the precision of the method for larger data sets as well. The exactness of the ability estimates are promising in this respect: in a way, these can be compared to estimates for very long tests (taken by only a few people). However, both the CPU-time and storage needed for larger item sets will increase with the number of items. But perhaps this problem will not become too serious, though: recall that for estimating the parameters of a larger test probably the number of Monte Carlo samples can be substantially reduced. This will have an impact on both CPU and storage requirements.

The above leads me to conclude that although the method works, a lot of work still needs to be done to make it suitable for CML estimation in models where exact CML estimates are impossible to obtain.

References

Andersen, E.B. & Madsen, M. (1977). Estimating the parameters of the latent population distribution. Psychometrika, 42, 357-374.

Bock, R.D. & Lieberman, M. (1970). Fitting a response model for dichotomously scored items. Psychometrika, 35, 179-197.

Ford, L.R.J. (1957). Solution of a ranking problem from binary comparisons. American Mathematical Monthly, 64, 241-252.

Geyer, C.J. & Thompson, E.A. (1992). Constrained Monte Carlo Maximum Likelihood for dependend data (with discussion). Journal of the Royal Statistical Society, series B, 54, 657-699.

Gustafsson, J.E. (1979). PML: a computer program for conditional estimation and testing in the Rasch model for dichotomous items. Gotheburg: Report nr. 85 from Institute of Education.

Hastings, W.K. (1970). Monte Carlo sampling methods using Markov Chanis, and their applications. Biometrika, 57, 97-109.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E. (1953). Equations of state calculations by fast computing machines. Journal of Chemical Physics, 21, 1087-1092.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. (1992). Numerical recipes in C. The art of scientific computing. Cambridge: Cambridge University Press.

Proth, J.M. & Hillion, H.P. (1990). Mathematical Tools in Production Management. New York: Plenum Press.

Rasch, G. (1980). Probabilistic models for some intelligence and attainment tests. Chicago: The University of Chicago Press (reprint from 1960).

Thissen, D.M. (1982). Marginal maximum likelihood estimation for the one-parameter logistic model. Psychometrika, 47, 175-186.

Table 1: **Chi-square goodness of fit values for the distribution of generated response patterns.**

		χ^2 values for		
t	df	B=500	B=1000	B=2000
1	4	3.94	2.00	2.21
2	9	1.88	12.83	11.71
3	9	11.19	8.23	7.76
4	4	0.27	11.86	8.01

t=total score, df=degrees of freedom.

B=number of generated response patterns.

Table 2: **Relative accuracy of the estimator $\hat{d}_t(\delta)$. Results for 1000 replications.**

B	dist	t	mean	sd	min	max
500	0.1	1	-0.0013	0.022	-0.043	0.040
		2	-0.0015	0.037	-0.066	0.069
		3	-0.0015	0.047	-0.079	0.[illegible]
		4	-0.0013	0.054	-0.091	0.[illegible]
	1.0	1	-0.012	0.20	-0.46	0.53
		2	0.021	0.36	-0.55	1.02
		3	0.080	0.49	-0.60	1.31
		4	0.135	0.59	-0.63	1.56
	3.0	1	-0.074	0.45	-0.94	3.55
		2	0.178	1.07	-0.94	7.79
		3	0.821	2.16	-0.95	11.41
		4	1.678	3.57	-0.95	15.98
2000	0.1	1	0.0008	0.021	-0.041	0.039
		2	0.0018	0.036	-0.065	0.067
		3	0.0028	0.046	-0.081	0.084
		4	0.0035	0.053	-0.091	0.099
	1.0	1	-0.014	0.21	-0.47	0.45
		2	0.030	0.36	-0.56	0.90
		3	0.088	0.49	-0.60	1.23
		4	0.145	0.60	-0.62	1.55
	3.0	1	-0.092	0.47	-0.92	2.51
		2	0.146	0.99	-0.95	5.42
		3	0.697	1.99	-0.95	11.31
		4	1.478	3.29	-0.95	15.91

B Number of simulations.

dist order of distance from ψ to δ, a distance of d means $\delta_j - \psi_j < d$, $\forall j$

t=total score, mean=average relative error of $\hat{d}_t(\delta)$, sd=standard deviation of average relative error

min=minimum observed value of relative error; max=maximum observed value of relative error

Table 3: **Item and ability parameter estimates for lsat data.**

data	item	mc-it	cml	se	diff	t	mc-ab	cml	se	diff
lsat6	1	-1.20	-1.26	.12	.06	1	-1.74	-1.60	1.18	-.14
	2	-.70	-.62	.10	-.08	2	-.52	-.47	.99	-.05
	3	.23	.17	.09	.06	3	.53	.48	.99	.03
	4	.43	.47	.08	-.04	4	1.73	1.60	1.18	.12
	5	1.24	1.24	.08	.00	-	-	-	-	-
lsat7	1	-.79	-.67	.10	-.12	1	-1.53	-1.44	1.14	-.09
	2	-.43	-.54	.10	.11	2	-.46	-.44	.95	-.02
	3	-.08	-.13	.09	.05	3	.45	.44	.95	.01
	4	.52	.54	.08	-.02	4	1.54	1.44	1.15	.10
	5	.78	.81	.08	-.03	-	-	-	-	-

mc-it. monte carlo estimate for item parameters, mc-ab monte carlo estimate for ability parameters.
cml exact cml estimates; se estimated standard errors for cml t. total score.
diff montecarlo-estimates - cml-estimates.

Table 4: **Item and ability parameter estimates for the memory data.**

item	mc-est	cml	se	diff	t	mc-est	'cml'	se	diff
1	-2.42	-2.49	.46	.07	1	-2.81	-2.71	1.16	-.10
2	-1.04	-1.02	.36	-.02	2	-1.68	-1.69	.90	.01
3	-.85	-.91	.36	.06	3	-.98	-1.00	.78	.02
4	.10	.05	.39	.05	4	-.39	-.43	.73	.04
5	.31	.33	.40	-.02	5	.10	.08	.70	.02
6	.44	.49	.42	-.05	6	.59	.57	.70	.02
7	.44	.49	.42	-.05	7	1.10	1.09	.74	.01
8	.60	.66	.43	-.06	8	-	1.69	.83	-
9	1.16	1.07	.48	.09	9	-	2.56	1.80	-
10	1.28	1.33	.52	-.05	-	-	-	-	-

mc-it. monte carlo estimate of item parameters; mc-ab. monte carlo estimate of ability parameters.
cml exact cml estimates, se estimated standard errors for cml t: total score
diff. montecarlo-estimate - cml-estimate

Table 5: **Time needed for computations.**

	data	cpu-time	real time
item parameters	lsat6	139	2:37
	lsat7	148	2:45
	memory	458	8:44
person parameters	lsat6	1596	28:21
	lsat7	1713	33:26
	memory	665	11:14

cpu-time = number of central processor units.

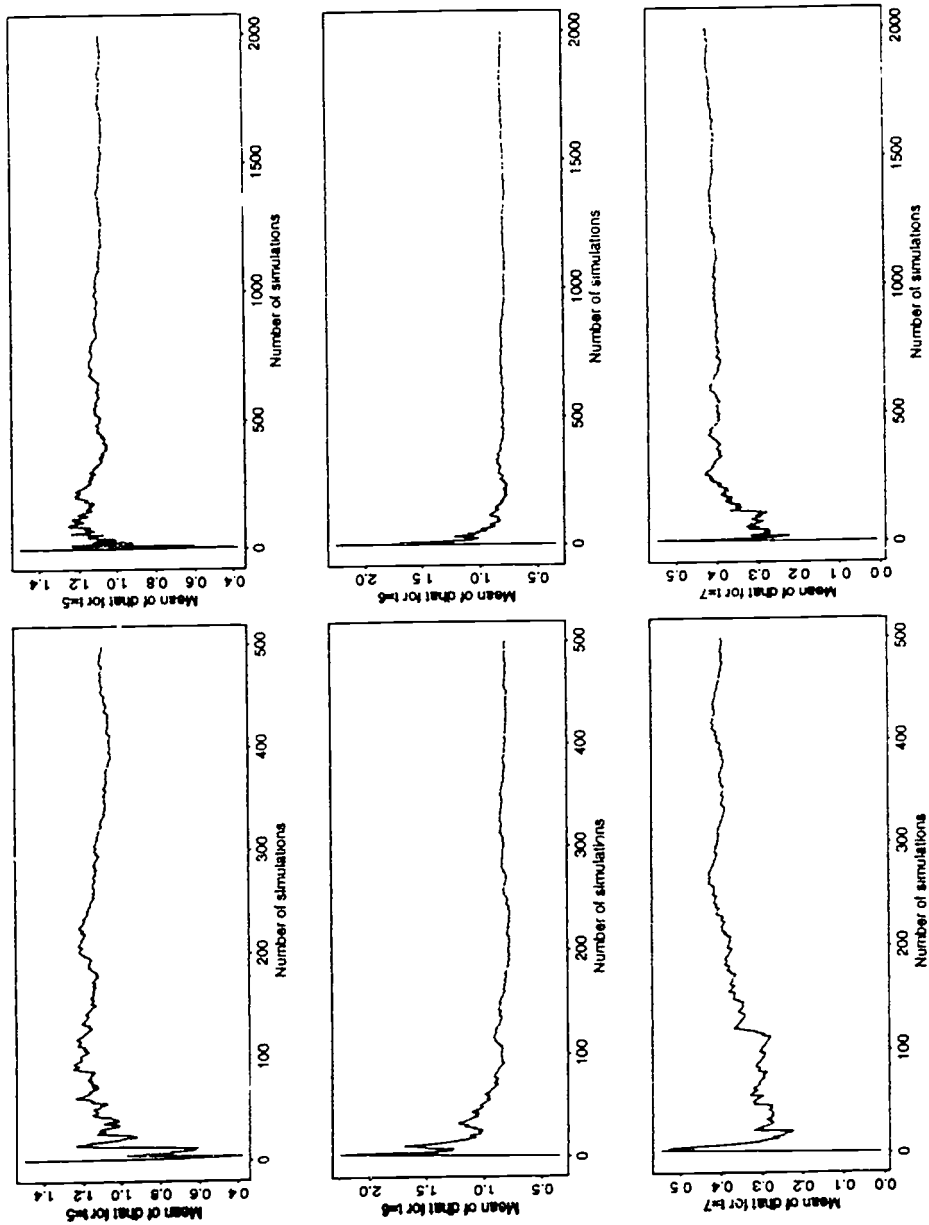
real time = elapsed time in minutes and seconds.

Authors' Note

The research for this report has been made possible in part by grants from NWO (the Dutch Organization for Scientific Research) and the Dr. Catharina van Tussenbroekfonds. Furthermore, I would like to express my thanks to Dr. Alun Thomas from the University of Bath for teaching me about Monte Carlo methods. I really enjoyed his lectures, and his comments and suggestions on the topic of this present paper have been very valuable to me.

Figure Caption

Sequential estimates $\dot{d}_t(\boldsymbol{\delta})^{(b)}$ for some values of t, for B=500 and B=2000.
Memory data; distance between δ_j and $\psi_j \leq 3$, $\forall j$



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

RR-94-9 L.M.W. Akkermans, *Monte Carlo estimation of the conditional Rasch model*

RR-94-8 R.R. Meijer & K. Sijtsma, *Detection of aberrant item score patterns: A review of recent developments*

RR-94-7 W.J. van der Linden & R.M. Luecht, *An optimization model for test assembly to match observed-score distributions*

RR-94-6 W.J.J. Veerkamp & M.P.F. Berger, *Some new item selection criteria for adaptive testing*

RR-94-5 R.R. Meijer, K. Sijtsma & I.W. Molenaar, *Reliability estimation for single dichotomous items*

RR-94-4 M.P.F. Berger & W.J.J. Veerkamp, *A review of selection methods for optimal design*

RR-94-3 W.J. van der Linden, *A conceptual analysis of standard setting in large-scale assessments*

RR-94-2 W.J. van der Linden & H.J. Vos. *A compensatory approach to optimal selection with mastery scores*

RR-94-1 R.R. Meijer, *The influence of the presence of deviant item score patterns on the power of a person-fit statistic*

RR-93-1 P. Westers & H. Kelderman, *Generalizations of the Solution-Error Response-Error Model*

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, Faculty of Educational Science and Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

faculty of
EDUCATIONAL SCIENCE AND TECHNOLOGY

A publication by
The Faculty of Educational Science and Technology of the University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands