Quantitative and Computational Methods in Behavioral Sciences

PsychOpen GOLD

Fundamental Research Articles

Check for updates

A Note on a Computationally Efficient Implementation of the EM Algorithm in Item Response Models

Alexander Robitzsch [1,2]

[1] *IPN – Leibniz Institute for Science and Mathematics Education, Kiel, Germany.* **[2]** *Center for International Student Assessment (ZIB), Kiel, Germany.*

Quantitative and Computational Methods in Behavioral Sciences, 2021, Article e3783,
https://doi.org/10.5964/qcmb.3783

Received: 2020-05-28 • **Accepted:** 2021-03-25 • **Published (VoR):** 2021-05-11

Corresponding Author: Alexander Robitzsch, Leibniz Institute for Science and Mathematics Education (IPN), Olshausenstr. 62, 24118 Kiel, Germany. E-mail: robitzsch@leibniz-ipn.de

Abstract

This note sketches two computational shortcuts for estimating unidimensional item response models and multidimensional item response models with between-item dimensionality utilizing an expectation-maximization (EM) algorithm that relies on numerical integration with fixed quadrature points. It is shown that the number of operations required in the E-step can be reduced in situations of many cases and many items by appropriate shortcuts. Consequently, software implementations of a modified E-step in the EM algorithm could benefit from gains in computation time.

Keywords

computation time, statistical software, item response model, EM algorithm

This note sketches two computational shortcuts for estimating multidimensional item response theory (IRT) models (van der Linden & Hambleton, 1997). We focus on the estimation problem using the expectation-maximization (EM) algorithm using numerical integration. In particular, we consider a more efficient implementation of the expectation step (E-step) in the EM algorithm. The computational amount is estimated by calculating the number of required operations. Additionally, computational gains in a pilot study are reported. Our primary focus is to propose two algorithmic shortcuts that avoid redundant operations. Such approaches can also be found for structural equation models (Pritikin, Hunter, von Oertzen, Brick, & Boker, 2017; von Oertzen & Brick, 2014).

This paper is structured as follows. First, the EM algorithm in IRT models is introduced, and its computational amount is analyzed. Second, we show how computational gains are realized by using so-called pseudo-items instead of original items. Third, we suggest a computational shortcut for multidimensional IRT models whose items possess a simple loading structure (i.e., between item dimensionality). Finally, we conclude with a discussion.

EM Algorithm in IRT Models

For simplicity, we focus on dichotomous item responses x_{ni} in this paper, where $n = 1, \dots, N$ denote persons, and i denote items. Let $\boldsymbol{x}_n = (x_{n1}, \dots, x_{nI})$ denote the vector of observed data for person n. In the IRT model, it is assumed that the item response function (IRF) of item i depends on item parameters $\boldsymbol{\gamma}_i$, and the multidimensional distribution of the latent variable $\boldsymbol{\theta}$ depends on some parameter $\boldsymbol{\sigma}$. Let $p_i(x|\boldsymbol{\theta}; \boldsymbol{\gamma}_i)$ denote the item response probability $P(X_i = x|\boldsymbol{\theta})$ of item i ($i = 1, \dots, I$). The total log-likelihood is then given as

$$l(\boldsymbol{\gamma}, \boldsymbol{\sigma}) = \sum_{n=1}^{N} \log\left(\int \left[\prod_{i=1}^{I} p_i(x_{ni}|\boldsymbol{\theta}; \boldsymbol{\gamma}_i)\right] f_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \boldsymbol{\sigma}) \mathrm{d}\boldsymbol{\theta}\right), \tag{1}$$

where $f_{\boldsymbol{\theta}}$ denotes the density of the latent variable $\boldsymbol{\theta}$. In the following, we approximate the integral in Equation 1 by a finite summation and use a set of fixed quadrature points $\boldsymbol{\theta}_t$ ($t = 1, \dots, T$). The numerically approximated log-likelihood is then given as

$$l(\boldsymbol{\gamma}, \boldsymbol{\sigma}) = \sum_{n=1}^{N} \log\left(\sum_{t=1}^{T} \left[\prod_{i=1}^{I} p_i(x_{ni}|\boldsymbol{\theta}_t; \boldsymbol{\gamma}_i)\right] w_{\boldsymbol{\theta}}(\boldsymbol{\theta}_t; \boldsymbol{\sigma})\right), \tag{2}$$

where $w_{\boldsymbol{\theta}}$ is a discrete density distribution for $\boldsymbol{\theta}$. The latent variable $\boldsymbol{\theta}$ cannot be directly observed. For this reason, the EM algorithm (see Aitkin, 2016; Fu, 2019; Glas, 2016, for overviews) is often used for maximizing l in Equation 2. The EM algorithm alternates between two steps (see Baker & Kim, 2004, for details). In the E-step, the expected value of l in Equation 2 is computed by integrating $\boldsymbol{\theta}$ out using individual posterior distributions. In the M-step, the expected log-likelihood is maximized based on computed expected counts from the E-step. Note that the update of item parameters $\boldsymbol{\gamma}_i$ in the M-step is a logistic regression model using case weights.

In the E-step, individual posterior distributions $h_n(\boldsymbol{\theta}_t|\boldsymbol{x}_n)$ are computed as

$$h_n(\boldsymbol{\theta}_t|\boldsymbol{x}_n) = \frac{\left[\prod_{i=1}^{I} p_i(x_{ni}|\boldsymbol{\theta}_t; \boldsymbol{\gamma}_i)\right] w_{\boldsymbol{\theta}}(\boldsymbol{\theta}_t; \boldsymbol{\sigma})}{\sum_{u=1}^{T} \left[\prod_{i=1}^{I} p_i(x_{ni}|\boldsymbol{\theta}_u; \boldsymbol{\gamma}_i)\right] w_{\boldsymbol{\theta}}(\boldsymbol{\theta}_u; \boldsymbol{\sigma})}. \tag{3}$$

For all items, expected counts are computed as

$$e_i(x|\theta_t) = \sum_{n=1}^{N} h_n(\theta_t|\boldsymbol{x}_n)\mathbf{1}_{\{x_{ni}=x\}} \qquad x = 0, 1, \tag{4}$$

where **1** denotes the indicator function.

We can now compute the total number of operations (multiplications or additions) needed in the E-step. The multiplications of probabilities in the numerator in the fraction in Equation 3 are typically substituted by additions of logarithmized probabilities. The following identity is used:

$$\left[\prod_{i=1}^{I} p_i(x_{ni}|\theta_t;\boldsymbol{\gamma}_i)\right] w_\theta(\theta_t;\boldsymbol{\sigma}) = \exp\left\{\left[\sum_{i=1}^{I} \log p_i(x_{ni}|\theta_t;\boldsymbol{\gamma}_i)\right] + \log w_\theta(\theta_t;\boldsymbol{\sigma})\right\}. \tag{5}$$

Notably, additions are typically faster than multiplications. Moreover, calculations on the logarithmic metric prevent numerical underflow (see, e.g., Vermunt, 2003, for an application in multilevel latent class models). First, for the computation of individual posterior distributions (see Equation 3), NTI operations are needed. In addition, NT operations are needed for normalizing the individual posterior distributions. However, we ignore this amount of operations in our calculations because we are particularly interested in situations in which I is large, say larger than 50.

Second, NTI operations are required for computing expected counts in Equation 4. Note that in software implementations, the multiplication with the indicator function **1** in Equation 4 must not be carried out in the summation for the expected counts $e_i(x|\theta_t)$ because one has only to add posterior probabilities for subjects n with $x_{ni} = x$. In practice, one uses an index vector for storing the necessary information. In total, the number of operations O_1 needed in the E-step can be determined as

$$O_1 = 2NTI\,. \tag{6}$$

Computationally Efficient EM Algorithms

One strategy for improving computational efficiency is to use a parallelized version of an EM algorithm (López-de Teruel, García, & Acacio, 1999). This approach distributes the computational work in the E-step and the M-step among multiple cores. There are several applications of this strategy in mixture modeling (Chen, Ostrouchov, Pugmire, Prabhat, & Wehner, 2013; Kwedlo, 2014; Lee, Leemaqz, & McLachlan, 2016; Lee, McLachlan, & Leemaqz, 2020; Yin, Zhang, & Gao, 2018). von Davier (2016) proposed a parallel EM algorithm for multidimensional IRT models and found considerable performance gains, in particular, for large datasets (see also Khorramdel, Shin, & von Davier, 2019).

A different strand of literature relies on simplifying the model structure in multidimensional IRT models (see Rijmen, Jeon, von Davier, & Rabe-Hesketh, 2014, for an overview). In this case, the computation of posterior distributions can be simplified if some trait distributions are (conditionally) independent (Rijmen, 2009). Examples include the bi-factor model (a.k.a. testlet models, Gibbons & Hedeker, 1992) and the two-tier model (Cai, 2010).

Usage of Pseudo-Items

Pseudo-Items

In the following, the concept of pseudo-items is introduced. The main idea is to perform the computationally intensive likelihood computation on pseudo-items instead of original items to reduce computation time. A pseudo-item is composed of k original polytomous items. Assume that $I = I_k k$ for an integer k, and there are now I_k pseudo-items. The number k shall also be noted as the size of pseudo-items. More formally, let $\left(x_1, \ldots, x_k\right)$ denote a vector of k polytomous items, each possessing H categories, a pseudo-item y_j is defined as

$$y_j = \sum_{i=1}^{k} H^{i-1} x_{i+I_k(j-1)}, \quad j = 1, \ldots, I_k \,. \tag{7}$$

The pseudo-item y_j takes H^k integer values $0, 1, \ldots, H^k - 1$.

In the sequel, we only consider the case of dichotomous items (i.e., $H = 2$). For example, $k = 3$ original dichotomous items are recoded as a pseudo-item y that takes integer values $0, 1, \ldots, 7$. The original data consisting of I dichotomous item responses $\boldsymbol{x}_n = \left(x_{n1}, \ldots, x_{nI}\right)$ for subjects $n = 1, \ldots, N$ is recoded into data $\mathbf{y}_n = \left(y_{n1}, \ldots, y_{nI_k}\right)$ consisting of polytomous pseudo-items.

Using Pseudo-Items in the EM Algorithm

A computationally more efficient EM algorithm is obtained by evaluating the posterior distributions based on these pseudo-items. Instead of using I original items in the posterior, now only I_k pseudo-items are involved, and the number of multiplications needed in the posterior is reduced. However, the number of probabilities that have to be additionally precomputed is given by $I_k 2^k k T = 2^k I T$. Due to local independence, the item response probabilities of pseudo-item y_j are given as

$$\tilde{p}_j\left(y \middle| \boldsymbol{\theta}; \boldsymbol{\gamma}\right) = \prod_{i=1}^{I_k} p_{i+I_k(h-1)}\left(x_i \middle| \boldsymbol{\theta}\right) \quad \text{with} \quad y = \sum_{i=1}^{k} 2^{i-1} x_i \,. \tag{8}$$

The computation based on individual posterior distributions is now given as (insert Equation 8 into Equation 3)

$$h_n(\theta_t|\boldsymbol{x}_n) = h_n(\theta_t|\mathbf{y}_n) = \frac{\left[\prod_{j=1}^{I_k} \tilde{p}_j\left(y_{nj}|\theta_t;\boldsymbol{\gamma}\right)\right] f_\theta(\theta_t;\boldsymbol{\sigma})}{\sum_{u=1}^{T}\left[\prod_{j=1}^{I_k} \tilde{p}_j\left(y_{nj}|\theta_u;\boldsymbol{\gamma}\right)\right] f_\theta(\theta_u;\boldsymbol{\sigma})} . \tag{9}$$

The number of operations needed in the posterior computation in Equation 9 is $NTI_k = NTI/k$.

Expected counts can now be computed for pseudo-items y_j:

$$\tilde{e}_j(y|\theta_t) = \sum_{n=1}^{N} h_n(\theta_t|\mathbf{y}_n)\mathbf{1}_{\{y_{nj} = y\}}, \quad y = 0, 1, \ldots, 2^k - 1 . \tag{10}$$

Therefore, for each pseudo-item j and each trait grid point θ_t, 2^k expected counts are computed. Based on these counts $\tilde{e}_j$, expected counts for item i can be computed by summing over appropriate counts of pseudo-items. The total number of operations needed for computing the expected counts is then given as $NTI/k + 2^kIT$. Hence, the total number of operations $O_2(k)$ for computing individual posterior distributions and expected counts for items based on the approach using I_k pseudo-items is given as

$$O_2(k) = 2NTI/k + 2^kIT = IT\left(2N/k + 2^k\right) . \tag{11}$$

We can now define the computational gain $G(N, k)$ that describes the factor of decrease in computation time by using pseudo-items:

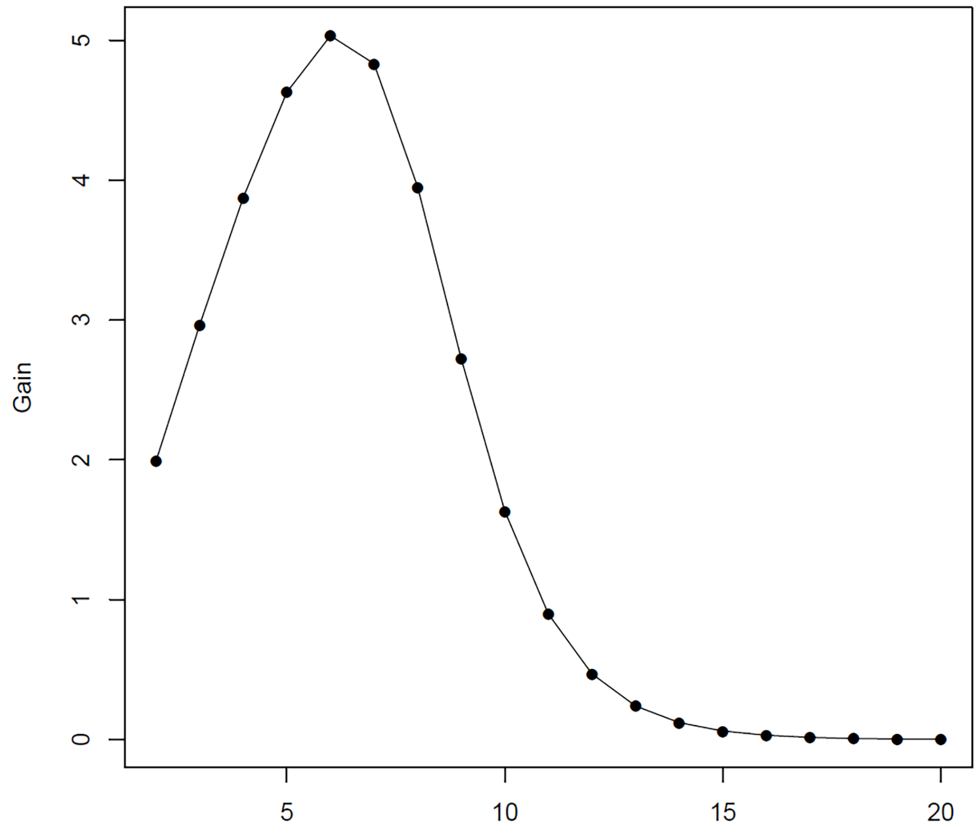
$$G(N,k) = \frac{O_1}{O_2(k)} = \frac{N}{N/k + 2^{k-1}} = k\frac{1}{1 + k2^{k-1}/N} . \tag{12}$$

Hence, for a fixed pseudo-item size k, the maximum gain $G(N, k)$ is k. It should be emphasized that the gain in the E-step does neither depend on the number of items I nor the number of integration points T.

In Figure 1, the computational gain is shown as a function of pseudo-item size k for a sample size of $N = 1000$. It can be seen that there is a maximum at $k = 6$ with a corresponding gain of about 5. This means that using pseudo-items that are defined by grouping 6 original items results in the highest efficiency gain. Notably, using a too large value of k results in a very inefficient computation.

Figure 1

Computational Gain as a Function of Pseudo-Item size k for a Sample Size of N = 1000



Due to $\lim_{k\to\infty}\left(N/k+2^k\right)=\infty$, it holds that $\lim_{k\to\infty} G(N,k)=0$. Hence, the size of pseudo-items has to be optimally chosen, and using a too large value of k can lead to an inefficient algorithm. Also, note that for a fixed k, it holds that $\lim_{N\to\infty} G(N,k)=k$.

Computation of an Optimal Pseudo-Item Size

We now determine the optimal pseudo-item size k_0 that maximizes the gain $G(N,k)$ defined in Equation 12. For simplicity, the pseudo-item size k is now regarded as a continuous variable. Define $g(k)=Nk^{-1}+2^{k-1}$. For a fixed sample size N, a maximum of $G(N,k)$ is given as a minimum $g(k)$. The derivative of g is given by $g'(k)=-Nk^{-2}+\log(2)2^{k-1}$. Hence, the optimal value k_0 fulfills

$$k^2 2^{k-1} = \frac{N}{\log(2)} \qquad \text{or} \qquad k2^{k-1} = \frac{N}{\log(2)k}. \tag{13}$$

Inserting Equation 13 into Equation 12 provides the maximum gain G_0 that can be achieved

$$G_0 = G(N, k_0) = k_0 \frac{N}{N + k_0 2^{k_0 - 1}} = k_0 \frac{N}{N + \frac{N}{\log(2)k_0}} = k_0 \frac{1}{1 + 1/(\log(2)k_0)}. \tag{14}$$

By taking the logarithm in Equation 13, it can be seen that the optimal k_0 fulfills

$$2 \log k + \log(2)k = \log N + \log(2) - \log(\log(2)). \tag{15}$$

Because the term k is faster increasing than the term $\log k$, the optimal k_0 is primarily a function of $\log N$, i.e., $k_0 = O(\log(N))$.

Analytical Illustration

We now illustrate the computational gains of the more efficient E-step implementation for sample sizes ranging between $N = 100$ and $N = 10\ 000\ 000$ subjects. In Table 1, the computational gain of using pseudo-items in the E-step is shown as a function of sample size N using Equation 12. It can be seen that the maximum computational gain G_0 increases with increasing sample sizes. Increases in computational efficiency are indeed non-negligible, at least for applications of international large-scale assessment studies like PISA (OECD, 2017) that involve student sample sizes of about half a million.

Table 1

Analytically Calculated Computational Gain From Equation 12 Using Pseudo-Items in the E-Step as a Function of Sample Size

Sample size	k_{max}	G_0
100	4	3.0
1 000	6	5.0
10 000	9	7.3
100 000	11	9.9
1 000 000	14	12.6
10 000 000	17	15.3

Note. k_{max} = pseudo-item size that results in maximum computational gain; G_0 = maximum computational gain.

Computational Illustration

We now demonstrate how the proposed shortcut using pseudo-items performs in an experimental implementation using the Rcpp package (Eddelbuettel, 2013; Eddelbuettel & François, 2011) in the R software (R Core Team, 2020). The E-step consists of three parts in the computation. First, the logarithms of the item response probabilities are added to the logarithm of the prior distribution. In this first part, an incomplete version of the unnormalized likelihood is computed. Afterward, the exp operation has to be applied to obtain the complete unnormalized likelihood. Second, the unnormalized likelihood on the logarithmic scale is further processed by taking the exp operation and normalizing the likelihood contributions for each subject for obtaining individual posterior distributions. Third, expected counts are computed based on individual posterior distributions.

The computational gains were investigated using sample sizes $N = 100$, 1000, 10000, and 100000. The number of items was fixed to 50, 100, and 200. In the EM algorithm, the number of quadrature points was fixed to 21. The R package microbenchmark (Mersmann, Beleites, Hurling, Friedman, & Ulrich, 2019) was used for timing comparisons. In each condition, 500 replications were utilized in the comparison.

In Table 2, computational gains are displayed for different pseudo-item sizes. It can be seen that efficiency gains were only about 2 or 3 if the complete E-step consisting of all three parts is used in the comparison. Notably, the exponentiation step and the normalization of the likelihood was not quantified in our analytically derived computational gains. Interestingly, it turned out that the predictions of computational gains were fulfilled for the computation of the unnormalized likelihood (without taking exp, first part) and the computation of expected counts (third part). The second part (exponentiation and normalization of the likelihood terms) is unrelated to using pseudo-items. By considering only the first part of the E-step, computational gains range between 3 and 10, meaning that this part is now three times to ten times faster than a previous implementation. For the third step, similar gains were obtained. Very likely, this finding demonstrates that applying the exponentiatial function is much more computationally demanding than an addition or multiplication.

By comparing the computational gains of the computation of the unnormalized likelihood with the prediction of Equation 12 in Table 2, we observe a moderate agreement of maximum computation gains for sample sizes of at least 1 000. Differences between timings obtained from implementations and analytical predictions could also emerge from the computation time required for memory allocation or accessing appropriate matrix entries in functions.

Table 2

Computational Gain in an Rcpp Implementation Using Pseudo-Items in the E-step as a Function of Pseudo-Item Size, Sample Size, and Number of Items

Sample size: No. of items	**Pseudo-item size k**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
					Complete E-step					
100:										
50	**1**	0.5	0.3	0.3	0.4	0.3	0.2	0.2	0.1	0.1
100	**1**	0.7	0.5	0.6	0.5	0.3	0.2	0.2	0.1	0.1
200	**1**	0.9	0.7	0.8	0.6	0.4	0.3	0.2	0.1	0.0
1 000:										
50	1	1.3	1.2	1.2	**1.4**	1.1	0.9	0.8	0.6	0.5
100	1	1.6	1.8	1.9	**2.0**	1.6	1.4	1.1	0.7	0.4
200	1	1.6	1.9	**2.3**	1.9	2.0	1.7	1.2	0.8	0.4
10 000:										
50	1	1.6	1.9	2.0	**2.1**	2.0	1.6	1.6	1.5	1.4
100	1	1.8	2.4	2.7	2.8	**2.8**	2.7	2.3	1.8	1.6
200	1	1.7	2.3	2.8	2.5	3.0	**3.1**	2.6	2.1	1.4
100 000:										
50	1	1.6	1.9	2.1	**2.2**	2.2	1.7	1.9	1.7	1.8
100	1	1.8	2.3	2.7	2.8	**2.8**	2.8	2.5	2.0	2.1
200	1	1.7	2.4	2.8	2.5	3.2	**3.3**	2.9	2.5	2.0
					Unnormalized likelihood without exp *operation*					
100:										
50	**1**	0.6	0.4	0.4	0.5	0.3	0.3	0.2	0.2	0.1
100	**1**	0.9	0.6	0.8	0.7	0.4	0.3	0.2	0.2	0.1
200	1	**1.3**	0.9	1.1	1.0	0.5	0.4	0.3	0.2	0.1
1 000:										
50	1	2.7	2.3	2.0	**3.0**	2.1	1.9	1.6	1.2	0.9
100	1	3.2	3.5	4.3	**4.3**	3.0	2.5	2.1	1.5	0.8
200	1	3.0	3.7	4.4	**4.9**	3.8	3.0	2.7	1.6	0.7
10 000:										
50	1	4.3	5.5	6.5	**7.3**	7.1	6.3	6.2	5.3	5.2
100	1	3.9	6.1	7.6	**9.1**	8.0	8.1	7.4	6.6	4.9
200	1	3.6	5.3	6.9	**8.4**	7.9	8.1	8.1	6.9	3.7
100 000:										
50	1	4.3	5.9	7.1	8.3	**8.8**	8.0	7.8	8.1	8.4
100	1	3.8	5.9	7.8	9.1	**9.5**	8.7	9.2	8.7	9.2
200	1	3.6	5.2	7.0	8.4	8.6	9.0	**9.8**	9.5	6.0
					Computational gain predicted from Equation 12					
100	1	1.9	2.7	**3.0**	2.8	2.1	1.3	0.7	0.4	0.2
1 000	1	2.0	3.0	3.9	4.6	**5.0**	4.8	4.0	2.7	1.6
10 000	1	2.0	3.0	4.0	5.0	5.9	6.7	7.3	**7.3**	6.6
100 000	1	2.0	3.0	4.0	5.0	6.0	7.0	7.9	8.8	**9.5**

Note. Optimal computational gains are printed in bold. Computational gains computed by Equation 12 do not involve the number of items.

We also investigated computation gains using 61 quadrature points which is the default for unidimensional IRT model in the popular R package mirt (Chalmers, 2012). The findings for computational gains were similar albeit not identical (i.e., some variation in gains of at most 1 in each condition).

It has to be noted that the optimal pseudo-item size from Equation 12 does not coincide with the empirically obtained optimal pseudo-item size. While the computational gains in the computation of the unnormalized likelihood approximately match the analytical predictions (see Table 2), the optimal pseudo-item became smaller in practical implementations. Depending on specific software implementation, it is recommended to always check potential computational gains depending on the sample size, the number of items, and the number of quadrature points for determining the optimal pseudo-item size.

Overall, it has to be admitted that the proposed shortcut might not result in considerable gains in the E-step of the EM algorithm. However, similar gains would also be obtained in parallel computing if only three or four cores were used.

Dimension-Wise Likelihood Computation in Between Item Dimensionality Multidimensional IRT Models

Assume now that in a multidimensional IRT model, each item loads on exactly one $\boldsymbol{\theta}$ dimension. This property is labeled as between item dimensionality (Adams, Wilson, & Wang, 1997). It is assumed that for each dimension d $(d = 1, \ldots, D)$, there are I_d items. The item response probability in the case of between item dimensionality now only depends on one dimension. Hence, the total likelihood contribution for subject n can be factored as

$$\prod_{d=1}^{D}\prod_{i=1}^{I_d} p_i\left(x_{ndi}\middle|\boldsymbol{\theta};\boldsymbol{\gamma}_{di}\right) = \prod_{d=1}^{D}\prod_{i=1}^{I_d} p_i\left(x_{ndi}\middle|\theta_d;\boldsymbol{\gamma}_{di}\right) = \prod_{d=1}^{D} L_{nd}(\boldsymbol{\theta};\boldsymbol{x}_{nd}), \tag{16}$$

where x_{ndi} denotes the item response of subject n for the i th on dimension d. From Equation 16, it is evident that the likelihood part in individual posterior distributions can be independently computed for all dimensions. In a multidimensional IRT model, one often uses the same $\boldsymbol{\theta}$ grid values across dimensions. For example, let $\mathcal{Q} = \{-4.0, -3.6, \ldots, 4.0\}$ be again the grid of fixed quadrature points for $\boldsymbol{\theta}$, the multidimensional $\boldsymbol{\theta}$ grid is obtained by defining $\mathcal{Q}^D$. With $Q = |\mathcal{Q}|$, $T = Q^D$ grid values $\boldsymbol{\theta}_t$ are defined, and each component of this vector possesses values in $\mathcal{Q}$. Using Equation 6,

$$O_1 = 2NQ^D I \tag{17}$$

operations are needed for computing the posterior distributions. It turns out that in the computation of the posterior $h_n(\boldsymbol{\theta}_t; \boldsymbol{x}_n) \propto L_n(\boldsymbol{\theta}_t; \boldsymbol{x}_n)$, the product term does not have to be calculated for all dimensions. In contrast, one can rely on the factorization

$$L_n(\boldsymbol{\theta}_t; \boldsymbol{x}_n) = \prod_{d=1}^{D} L_{nd}(\theta_{td}; \boldsymbol{x}_{nd}) \,. \tag{18}$$

The expected counts for an item i have only to be evaluated based on a dimension d due to between item dimensionality. Hence, one can integrate over all dimensions except the d th dimension. Using the notation $\boldsymbol{\theta}_t = \left(\theta_{td}, \boldsymbol{\theta}_{t(-d)}\right)$, one can compute dimension-wise individual marginal posterior distributions h^*_{nd}:

$$h^*_{nd}(\theta_{td} | \boldsymbol{x}_n) = \textstyle\sum_{(-d)} h_n\left(\theta_{td}, \boldsymbol{\theta}_{t(-d)} | \boldsymbol{x}_n\right), \tag{19}$$

where $\sum_{(-d)}$ denotes a summation over all dimensions except d. Based on these marginal distributions, the computational demand of expected counts is substantially reduced as one has to only sum over individual probabilities related on exactly one dimension for each item. By employing these computational shortcuts, the number of required operations can be reduced to

$$O_2 = 2NQ \sum_{d=1}^{D} I_d + 2NQ^D = 2NQI + (D+1)NQ^D \,. \tag{20}$$

This less demanding computation provides a computational gain of

$$G = \frac{O_1}{O_2} = \frac{2NIQ^D}{2NIQ + (D+1)NQ^D} = Q^{D-1} \frac{1}{1 + (D+1)Q^{D-1}/(2I)} \,. \tag{21}$$

For a large number of items I, the computational gain approaches Q^{D-1}. Note that the computational gain is independent of the sample size N.

Combination of Usage of Pseudo-Items and Dimension-Wise Likelihood Computation

Next, we combine the usage of pseudo-items and dimension-wise likelihood computation. For simplicity, we assume an equal number of items per dimension, i.e., $I_d = I/d$. Then, we obtain by using the same computations as in Equation 11

$$O_2(k) \;=\; 2NQI/k + (D+1)NQ^D + 2 \cdot 2^k IQ \,. \tag{22}$$

The computational gain can be computed as

$$G(N,k) = \frac{O_1}{O_2(k)} = \frac{2NQ^D I}{2NQI/k + (D+1)NQ^D + 2^k IQ} = kQ^{D-1}\frac{1}{1 + k(D+1)Q^{D-1}/(2I) + k2^{k-1}/N}. \tag{23}$$

In the minimization of O_2 with respect to k, the same optimal value k_0 is obtained as in Equation 13. Inserting Equation 13 into Equation 23 leads to an optimal gain of

$$G_0 = G(N,k_0) = k_0 Q^{D-1}\frac{1}{1 + k_0(D+1)Q^{D-1}/(2I) + 1/(\log(2)k_0)}. \tag{24}$$

Analytical Illustration

We now illustrate the computational gains by applying our proposed computational shortcuts. We consider a multidimensional IRT model with between item dimensionality with $I = 50$ items and $Q = 15$ fixed quadrature points per dimension. Sample sizes are varied between $N = 100$ and $N = 10\ 000\ 000$, and the number of dimensions D is chosen as 2, 3, or 4. Two computational shortcuts are compared. First, we use a strategy that only relies on the dimension-wise computation of the likelihood (Method DW). Second, we combine the dimension-wise computation with the pseudo-item approach that selects the optimal pseudo-item size.

In Table 3, computational gains based on Equation 23 of the two methods are shown as a function of sample sizes and dimensions. It can be seen that the dimension-wise method (Method DW) is independent of sample sizes, but especially results in substantial computational gains for larger numbers of dimensions. For $D = 2$ dimensions, additional usage of pseudo-items (Method DP) further provides relevant computational gains, while using pseudo-items for more dimensions ($D = 3$ or $D = 4$) is not as effective. Given that the E-step in numerical integration for multidimensional IRT models is particularly computationally demanding, we think that our proposed computational shortcuts could be beneficial for multidimensional models with between-item dimensionality. However, it has to be shown that the proposed shortcuts for multidimensional IRT models also show comparable computational gains in software implementations.

Table 3

Analytically Calculated Computational Gain Based on Equation 23 Using Dimension-Wise Computation and Pseudo-Items in the E-Step as a Function of Sample Size N and Dimensions D for Q = 15 Quadrature Points per Dimension, and I = 50 Items

	D **= 2**		***D*** **= 3**		***D*** **= 4**	
Sample size	**DW**	**DP**	**DW**	**DP**	**DW**	**DP**
100	10.3	19.2	22.5	24.1	19.9	20.0
1 000	10.3	23.1	22.5	24.5	19.9	20.0
10 000	10.3	25.6	22.5	24.6	19.9	20.0
100 000	10.3	27.2	22.5	24.7	19.9	20.0
1 000 000	10.3	28.3	22.5	24.8	19.9	20.0
10 000 000	10.3	29.1	22.5	24.8	19.9	20.0

Note. D = number of dimensions; DW = computational gain by using dimension-wise computation; DP = computational gain by using dimension-wise computation and maximum pseudo-item size.

Discussion

This note shows that by using two computational shortcuts, the number of operations required in an E-step in unidimensional or multidimensional models can be substantially reduced. First, the evaluation of individual posterior distributions can be more efficiently implemented by using pseudo-items that are composed of a set of original items. By pursuing this strategy, additional item response probabilities—evaluated at the $\boldsymbol{\theta}$ grid—for these pseudo-items have to be computed. However, as a consequence, fewer multiplications have to be conducted in the computation of individual posterior distributions. It was shown that the optimal number of original items that are grouped into pseudo-items is a function of the sample size. Second, for multidimensional IRT models with between item dimensionality, EM algorithms that rely on numerical integration can be more efficiently implemented if the joint individual likelihood function is factored as a product of dimension-wise likelihood functions. This strategy avoids the evaluation of multiplications of likelihood contributions on the full multidimensional grid of trait values. The usage of pseudo-items can additionally be applied for multidimensional IRT models, which results in further computational gains.

In our evaluation, we only considered the case of dichotomous item responses. In principle, the same strategy can be applied for datasets containing polytomous item responses. However, using pseudo-items for polytomous items with many categories will typically not result in such substantial computational gains as for dichotomous items because a larger number of probabilities have to be computed. The handling of ignorable missing item responses is relatively simple because one could treat them as a special case of polytomous item responses and define a corresponding probability of a missing category as one. In this strategy, missing item responses are essentially omitted from the computation. For planned missingness designs (Frey, Hartig, & Rupp, 2009; Rhemtulla & Little, 2012), more thoughtful approaches seem reasonable that skip not administered items in the computations. Moreover, computational gains by using pseudo-items directly transfer to latent class or mixture models.

Our proposed computational shortcuts can be combined with parallel computing (von Davier, 2016) or acceleration techniques (Beisemann, Wartlick, & Doebler, 2020) in the EM algorithm. This would lead to additional computational gains, which are particularly important for applications involving large numbers of items. This contribution is primarily meant as a theoretical contribution analyzing the amount of required computational operations. The preliminary evaluation of obtained performance gains of implementations in statistical software led to slightly smaller computation gains in practice. However, these gains are nevertheless noticeable.

Funding: The author has no funding to report.

Acknowledgments: I thank the associate editor and two anonymous reviewers for their helpful comments which substantially improved the manuscript.

Competing Interests: The author has declared that no competing interests exist.

References

Adams, R. J., Wilson, M., & Wang, W. C. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement, 21*(1), 1-23. https://doi.org/10.1177/0146621697211001

Aitkin, M. (2016). Expectation maximization algorithm and extensions. In W. J. van der Linden (Ed.), *Handbook of item response theory: Vol. 2. Statistical tools* (pp. 217-236). Boca Raton, FL, USA: CRC Press. https://doi.org/10.1201/b19166-12

Baker, F. B., & Kim, S. H. (2004). *Item response theory: Parameter estimation techniques* (2nd ed.). Boca Raton, FL, USA: CRC Press. https://doi.org/10.1201/9781482276725

Beisemann, M., Wartlick, O., & Doebler, P. (2020). Comparison of recent acceleration techniques for the EM algorithm in one-and two-parameter logistic IRT models. *Psych, 2*(4), 209-252. https://doi.org/10.3390/psych2040018

Cai, L. (2010). A two-tier full-information item factor analysis model with applications. *Psychometrika, 75*(4), 581-612. https://doi.org/10.1007/s11336-010-9178-0

Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software, 48*(6), 1-29. https://doi.org/10.18637/jss.v048.i06

Chen, W. C., Ostrouchov, G., Pugmire, D., Prabhat, & Wehner, M. (2013). A parallel EM algorithm for model-based clustering applied to the exploration of large spatio-temporal data. *Technometrics, 55*(4), 513-523. https://doi.org/10.1080/00401706.2013.826146

Eddelbuettel, D. (2013). *Seamless R and C++ integration with Rcpp*. New York, NY, USA: Springer. https://doi.org/10.1007/978-1-4614-6868-4

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software, 40*(8), 1-18. https://doi.org/10.18637/jss.v040.i08

Frey, A., Hartig, J., & Rupp, A. A. (2009). An NCME instructional module on booklet designs in large-scale assessments of student achievement: Theory and practice. *Educational Measurement: Issues and Practice, 28*(3), 39-53. https://doi.org/10.1111/j.1745-3992.2009.00154.x

Fu, J. (2019). *Maximum marginal likelihood estimation with an expectation-maximization algorithm for multigroup/mixture multidimensional item response theory models* (Research Report No. RR-19-35). Educational Testing Service. https://doi.org/10.1002/ets2.12272

Gibbons, R. D., & Hedeker, D. R. (1992). Full-information item bi-factor analysis. *Psychometrika, 57*, 423-436. https://doi.org/10.1007/BF02295430

Glas, C. A. W. (2016). Maximum-likelihood estimation. In W. J. van der Linden (Ed.), *Handbook of item response theory* (Vol. 2, pp. 197-216). Boca Raton, FL, USA: CRC Press. https://doi.org/10.1201/b19166-11

Khorramdel, L., Shin, H. J., & von Davier, M. (2019). GDM software mdltm including parallel EM algorithm. In M. von Davier & Y. S. Lee (Eds.), *Handbook of diagnostic classification models* (pp. 603-628). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-030-05584-4_30

Kwedlo, W. (2014). A parallel EM algorithm for Gaussian mixture models implemented on a NUMA system using OpenMP. In In M. Aldinucci, D. D'Agostino, & P. Kilpatrick (Eds.), *22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP 2014): Proceedings* (pp. 292-298). Los Alamitos, CA, USA: IEEE Computer Society. https://doi.org/10.1109/PDP.2014.77

Lee, S. X., Leemaqz, K. L., & McLachlan, G. J. (2016). A simple parallel EM algorithm for statistical learning via mixture models. In A. W.-C. Liew, B. Lovell, C. Fookes, J. Zhou, Y. Gao, M.l Blumenstein, & Z. Wang (Eds.), *International Conference on Digital Image Computing: Techniques and Applications (DICTA 2016): Proceedings* (pp. 1-8). Los Alamitos, CA, USA: IEEE Computer Society. https://doi.org/10.1109/DICTA.2016.7796997

Lee, S. X., McLachlan, G. J., & Leemaqz, K. L. (2020). Multi-node EM algorithm for finite mixture models. arXiv. https://arxiv.org/abs/2005.06848

López-de Teruel, P. E., García, J. M., & Acacio, M. E. (1999). The parallel EM algorithm and its applications in computer vision. In H. R. Arabnia & I. Ahmad (Eds.), *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 1999): Proceedings* (pp. 571-578). Las Vegas, NV, USA: CSREA Press.

Mersmann, O., Beleites, C., Hurling, R., Friedman, A., & Ulrich, J. M. (2019). *microbenchmark: Accurate timing functions* (R package version 1.4-7). Retrieved from https://CRAN.R-project.org/package=microbenchmark

OECD. (2017). *PISA 2015 technical report.* Retreived from https://www.oecd.org/pisa/data/2015-technical-report/

Pritikin, J. N., Hunter, M. D., von Oertzen, T., Brick, T. R., & Boker, S. M. (2017). Many-level multilevel structural equation modeling: An efficient evaluation strategy. *Structural Equation Modeling, 24*(5), 684-698. https://doi.org/10.1080/10705511.2017.1293542

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: Author. https://www.R-project.org/

Rhemtulla, M., & Little, T. D. (2012). Planned missing data designs for research in cognitive development. *Journal of Cognition and Development, 13*(4), 425-438. https://doi.org/10.1080/15248372.2012.717340

Rijmen, F. (2009). *Efficient full information maximum likelihood estimation for multidimensional IRT models* (Research Report No. RR-09-03). Educational Testing Service. https://doi.org/10.1002/j.2333-8504.2009.tb02160.x

Rijmen, F., Jeon, M., von Davier, M., & Rabe-Hesketh, S. (2014). A general psychometric approach for educational survey assessments: Flexible statistical models and efficient estimation methods. In L. Rutkowski, M. von Davier, & D. Rutkowski (Eds.), *A handbook of international large-scale assessment: Background, technical issues, and methods of data analysis* (pp. 583-606). Boca Raton, FL, USA: CRC Press. https://doi.org/10.1201/b16061-30

van der Linden, W. J., & Hambleton, R. K. (Eds.). (1997). *Handbook of modern item response theory.* New York, NY, USA: Springer. https://doi.org/10.1007/978-1-4757-2691-6

Vermunt, J. K. (2003). Multilevel latent class models. *Sociological Methodology, 33*(1), 213-239. https://doi.org/10.1111/j.0081-1750.2003.t01-1-00131.x

von Davier, M. (2016). *High-performance psychometrics: The parallel-E parallel-M algorithm for generalized latent variable models* (Research Report No. RR-16-34). Educational Testing Service. https://doi.org/10.1002/ets2.12120

von Oertzen, T., & Brick, T. R. (2014). Efficient Hessian computation using sparse matrix derivatives in RAM notation. *Behavior Research Methods, 46*(2), 385-395. https://doi.org/10.3758/s13428-013-0384-4

Yin, J., Zhang, Y., & Gao, L. (2018). Accelerating distributed Expectation–Maximization algorithms with frequent updates. *Journal of Parallel and Distributed Computing, 111*, 65-75. https://doi.org/10.1016/j.jpdc.2017.07.005