Title

A Modified Expectation-Maximization Algorithm for Accelerating Item Response Theory Model Estimation with Large Datasets

Author

Feng, Tianying

Publication Date

2025

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

A Modified Expectation-Maximization Algorithm for Accelerating Item Response Theory Model Estimation with Large Datasets

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Tianying Feng

2025

ABSTRACT OF THE THESIS

A Modified Expectation-Maximization Algorithm for Accelerating Item Response Theory Model Estimation with Large Datasets

by

Tianying Feng
Master of Science in Statistics
University of California, Los Angeles, 2025
Professor Yingnian Wu, Chair

The Expectation-Maximization (EM) algorithm is widely used for parameter estimation in item response theory (IRT) modeling. Despite its utility, when applied to datasets with large numbers of individuals and items, the standard EM algorithm can be slow to converge, with computationally expensive E-steps. This study proposes a modified EM algorithm with a two-stage structure that capitalizes on data subsets to accelerate estimation for unidimensional two-parameter logistic (2PL) IRT models. Two simulation studies evaluated the reduction in convergence time, bias and root mean squared error (RMSE) of parameter recovery, and bias and RMSE of standard error (SE) estimation under the proposed algorithm. The proposed algorithm was also compared to standard EM across varying subset sizes and item set lengths. Results showed that (a) parameter recovery differed between types of IRT parameters and between moderate and extreme parameter values; (b) recovery improved with larger subset sizes; (c) reduction in time was more pronounced with a larger item set; and (d) a small upward bias and RMSE in SE estimates were observed compared to standard EM, though both metrics remained modest in all conditions. Overall, the proposed algorithm improved computational efficiency while maintaining comparable estimation accuracy and precision under larger item sets and moderate-to-large subset sizes. The study concludes with future applica-

tions in large-scale operational contexts with large volumes of examinees and moderate to large item pools, and potential extensions to multidimensional IRT settings.

The thesis of Tianying Feng is approved.

Mark S. Handcock

Hongquan Xu

Yingnian Wu, Committee Chair

University of California, Los Angeles

2025

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

The Expectation-Maximization (EM) algorithm (Dempster et al., 1977) is an iterative procedure for finding maximum likelihood estimates when some data or variables are unobserved. It has been widely used for marginal maximum likelihood estimation in item response theory (IRT) modeling (Baker & Kim, 2004; Bock & Aitkin, 1981; von Davier, 2016).

Despite its utility, the EM algorithm can be slow to converge, with diminishing improvements as the estimates approach the solution (Baker & Kim, 2004; Bock & Aitkin, 1981; also see discussant comments in Dempster et al., 1977). Moreover, computing the E-step becomes computationally expensive as the sample size grows or the number of dimensions requiring integration increases.

1.1 Existing Extensions on Accelerating the EM Algorithm

1.1.1 Partial E-Steps

Variants of EM that modify the E-step to accelerate convergence have a long history in parallel computing and machine learning (Cappé & Moulines, 2009; Liang & Klein, 2009; Neal & Hinton, 1998; Thiesson et al., 2001; Yin et al., 2018). These algorithms partition the data into disjoint subsets but differ in how they store and update key statistics after each partial E-step.

One strategy is to parallelize the partial E- and/or M-steps (e.g., parallel-E parallel-M; von Davier, 2016). When parallelizing only the E-step, the algorithm computes subset-

specific expected statistics in parallel and aggregates them. The M-step is performed as in the standard EM algorithm, executed once after aggregating subset-specific results.

Another strategy is to incrementally update the key statistics as the algorithm processes each subset. For example, the incremental EM algorithm (Neal & Hinton, 1998) tracks both the overall and subset-level expected sufficient statistics. With each partial E-step, it updates the overall statistics by subtracting the old subset values and adding the newly computed ones. The stochastic approximation EM algorithm (SAEM; Cappé & Moulines, 2009; Delyon et al., 1999) uses a weighted interpolation between the current and subset-level statistics, with the weighting controlled by a step size parameter.

1.1.2 Partial M-Steps

The M-step of the EM algorithm can be modified to improve computational efficiency. The generalized EM algorithm (GEM; Dempster et al., 1977) relaxes the standard EM's requirement of maximizing the log-likelihood in the M-step by allowing for a computationally more efficient update that improves the log-likelihood. A subclass of GEM, the expectation-conditional maximization algorithm (Meng & Rubin, 1993), replaces the M-step of standard EM with multiple conditional maximization steps.

1.2 Features of Partial-Step Schemes

With partial steps, parameters are updated more frequently with reduced computation per pass through the data. This frequent updating allows subsequent steps to leverage more up-to-date parameter estimates, leading to faster convergence. The difference between partial EM and standard EM can be likened to hill climbing. In a partial-step scheme, each iteration of the algorithm takes several noisier but faster and computationally cheaper steps toward a local peak, with each step pointing in the right direction. In contrast, the standard EM algorithm computes the expected complete data log-likelihood using the whole dataset and update the parameters once per iteration, making one more

accurate but slower step toward the peak.

While noise from partial updates can help algorithms make rapid progress and escape poor stationary points (e.g., saddle points), excessive noise from stochastic updates can cause the iterates to converge to a ”noise ball,” wherein they fluctuate around the solution rather than converging to it (Gower et al., 2020). Several techniques can mitigate this issue, including using a sequence of decreasing step sizes that satisfy the Robbins-Monro conditions (Robbins & Monro, 1951), increasing the subset size, computing weighted averages, or incorporating infrequent full-data updates to reduce the variance of subset-driven updates (Chen et al., 2018; Johnson & Zhang, 2013).

1.3 This Thesis

In this thesis, I propose a modified EM algorithm for computationally efficient estimation of unidimensional IRT models, an area that remains underexplored. The modifications are straightforward to implement on top of an existing EM routine and offer computational benefits when applied to large datasets.

The remainder of this thesis is organized as follows. Chapter 2 introduces the two-parameter logistic IRT model, the standard EM algorithm for unidimensional IRT parameter estimation, and specific components of EM estimation that motivate the proposed modifications. Chapter 3 describes the proposed algorithm that uses a two-stage structure. Chapter 4 presents two simulation studies that evaluate the proposed algorithm against standard EM in terms of convergence time, parameter recovery, and standard error estimation across a range of data conditions. Chapter 5 discusses the simulation findings and concludes with key considerations for implementing and improving the proposed algorithm.

CHAPTER 2

The Standard EM Algorithm for IRT Estimation

2.1 Unidimensional Two-Parameter Logistic IRT Models

We focus our discussion on unidimensional two-parameter logistic (2PL) IRT models that are not members of the exponential family. Let η denote a standard normally distributed latent variable. Let i index the items ($i = 1, \ldots, I$), and let y_{ip}, coded as 0 or 1, denote Person p's dichotomously scored response to Item i. Using $\boldsymbol{\phi}_i$ to collect the two item-specific parameters a_i and c_i, the conditional probabilities are

$$P_i(\eta_p) := P(y_{ip} = 1 \mid \eta_p, \boldsymbol{\phi}_i) = \frac{1}{1 + \exp\left[-z_{ip}\right]} = \frac{1}{1 + \exp\left[-(a_i\eta_p + c_i)\right]} \tag{2.1}$$

$$Q_i(\eta_p) := P(y_{ip} = 0 \mid \eta_p, \boldsymbol{\phi}_i) = 1 - P_i(\eta_p) \tag{2.2}$$

The linear predictor z in Equation 2.1 is expressed in the slope-intercept form rather than the discrimination-difficulty form that one might be more familiar with. The two forms are related. The slope a_i is the discrimination parameter, which quantifies how sensitive an item is to slight differences in individuals' latent scores: the greater the sensitivity, the higher the discrimination value. In unidimensional IRT, the intercept parameter is the negative product of the slope and the difficulty parameters: $c_i = -a_i b_i$. The difficulty parameter b_i locates the point on the latent continuum at which the probability of responding 1 is .5, assuming no guessing.

Furthermore, let $\boldsymbol{y}_p$ denote Person p's response vector containing responses to I items. Under the assumption of local independence, the probability of observing $\boldsymbol{y}_p$

conditional on η_p and $\boldsymbol{\phi}$ is

$$P(\boldsymbol{y}_p \mid \eta_p, \boldsymbol{\phi}) = \prod_{i=i}^{I} P_i(\eta_p)^{y_{ip}} \cdot Q_i(\eta_p)^{1-y_{ip}} \tag{2.3}$$

The marginal probability of $\boldsymbol{y}_p$ is

$$P(\boldsymbol{y}_p \mid \boldsymbol{\phi}) = \int P(\boldsymbol{y}_p \mid \eta, \boldsymbol{\phi}) h(\eta) d\eta \tag{2.4}$$

where $h(\eta)$ is the standard normal density function for variable η.

Under the assumption of P independent individuals, the marginal log-likelihood for P response vectors collected in matrix $\boldsymbol{Y}$ is

$$l(\boldsymbol{\phi} \mid \boldsymbol{Y}) = \sum_{p=1}^{P} \log P(\boldsymbol{y}_p \mid \boldsymbol{\phi}) \tag{2.5}$$

2.2 Parameter Estimation with the Standard EM Algorithm

With the standard EM algorithm (Bock & Aitkin, 1981), the integration in Equation 2.4 is numerically approximated by a weighted sum over K quadrature nodes $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_K]'$. In its quadrature form, the marginal log-likelihood can be written as follows (Baker & Kim, 2004; Weissman, 2013):

$$\begin{aligned} l(\boldsymbol{\phi} \mid \boldsymbol{Y}) = & \sum_{p=1}^{P} \sum_{k=1}^{K} g(\eta_k \mid \boldsymbol{y}_p, \boldsymbol{\phi}) \left[\sum_{i=i}^{I} \log L(y_{ip} \mid \eta_k, \boldsymbol{\phi}_i) + \log h(\eta_k) \right] \\ & - \sum_{p=1}^{P} \sum_{k=1}^{K} g(\eta_k \mid \boldsymbol{y}_p, \boldsymbol{\phi}) \log g(\eta_k \mid \boldsymbol{y}_p, \boldsymbol{\phi}) \end{aligned} \tag{2.6}$$

where

$$g(\eta_k \mid \boldsymbol{y}_p, \boldsymbol{\phi}) = \frac{\prod_{i=i}^{I} L(y_{ip} \mid \eta_k, \boldsymbol{\phi}_i) h(\eta_k)}{\sum_{m=1}^{K} \prod_{i=i}^{I} L(y_{ip} \mid \eta_m, \boldsymbol{\phi}_i) h(\eta_m)} \tag{2.7}$$

is the posterior probability that the latent score of Person p is η_k.

By comparing Equation 2.6 to Equation 3.2[1] in Dempster et al. (1977), we observe that the two terms correspond to functions $Q(\cdot)$ and $H(\cdot)$. For Item i, the E-step then amounts to computing the Q function, that is, computing the expectation of the complete data log-likelihood with respect to the posterior distribution of $\boldsymbol{\phi}_i$ defined in Equation 2.7 and the current parameter estimates $\boldsymbol{\phi}_i^{(t)}$. The M-step amounts to finding $\boldsymbol{\phi}_i^{(t+1)}$ by maximizing the expectation computed in the E-step.

$$Q(\boldsymbol{\phi}_i \mid \boldsymbol{\phi}_i^{(t)}) = \sum_{k=1}^{K} \sum_{p=1}^{P} \Big\{ g(\eta_k \mid \boldsymbol{y}_p, \boldsymbol{\phi}^{(t)}) \left[y_{ip} \log P_i(\eta_k) + (1 - y_{ip}) \log Q_i(\eta_k) \right] \Big\} \tag{2.8}$$

Furthermore, the summation over p in Equation 2.8 can be condensed into one summation over the 1 responses and another over the 0 responses:

$$\tilde{r}_{1ik} = \sum_{p=1}^{P} g(\eta_k \mid \boldsymbol{y}_p, \boldsymbol{\phi}^{(t)}) y_{ip} \tag{2.9}$$

$$\tilde{r}_{0ik} = \sum_{p=1}^{P} g(\eta_k \mid \boldsymbol{y}_p, \boldsymbol{\phi}^{(t)}) (1 - y_{ip}) \tag{2.10}$$

where $\tilde{r}_{1ik}$ and $\tilde{r}_{0ik}$ denote the expected counts of 1 and 0 responses at quadrature node η_k for Item i.

In the standard EM algorithm, the E-step at Iteration $(t+1)$ uses Equations 2.9 and 2.10 to compute $\tilde{\boldsymbol{r}}_{1ik}$ and $\tilde{\boldsymbol{r}}_{0ik}$. The M-step then maximizes Equation 2.8 that simplifies to

$$Q(\boldsymbol{\phi}_i \mid \boldsymbol{\phi}_i^{(t)}) = \sum_{k=1}^{K} \Big\{ \tilde{r}_{1ik} \log P_i(\eta_k) + \tilde{r}_{0ik} \log Q_i(\eta_k) \Big\} \tag{2.11}$$

In the GEM algorithm (Dempster et al., 1977), a modified version of the M-step finds parameters $\boldsymbol{\phi}^*$ such that $Q(\boldsymbol{\phi}^* \mid \boldsymbol{\phi}^{(t)}) \geq Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(t)})$, instead of maximizing Equation 2.8. In other words, GEM proposes a more relaxed M-step that only needs to improve

[1] $L(\boldsymbol{\phi}') = Q(\boldsymbol{\phi}' | \boldsymbol{\phi}) - H(\boldsymbol{\phi}' | \boldsymbol{\phi})$

the log-likelihood.

2.3 Features That Motivate the Proposed Modifications

2.3.1 Assumption of Conditional Independence

The observed responses are conditionally independent given the latent variable. Unless otherwise specified by design, responses from different individuals are also assumed to be independent. These two types of independence allow the complete-data log-likelihood to be expressed as a sum over independent person-level (or item-level) contributions. This structure motivates the use of partial E-steps, where expectations are computed over randomly sampled data subsets (without replacement). This approach reduces the per-iteration computational cost without requiring significant changes to the underlying estimation logic.

2.3.2 Shape of the Log-Likelihood Surface Near the MLE

The log-likelihood surface near the MLE tends to be smooth and locally quadratic. When using Newton-style updates with a small fixed step size, noise introduced by partial E-steps causes the parameter iterates to fluctuate around the MLE within a bounded neighborhood. This characteristic behavior, visualized as a point cloud or noise ball, has been documented in the stochastic optimization literature (Gower et al., 2020). Averaging (Polyak & Juditsky, 1992) is a common heuristic to reduce this variability and improve estimate precision. A key consideration is whether the iterates have exited the initial phase and entered the noise ball region, where averaging becomes effective. The next chapter describes how to adaptively detect this transition and presents components of the proposed algorithm.

CHAPTER 3

The Modified EM Algorithm

This chapter introduces modifications to the standard EM algorithm for IRT estimation. In the standard formulation, maximizing Equation 2.11 involves maximizing some form of expectation over a data-defined distribution. Many properties of the log-likelihood function, such as the gradients, are also sums. When applied to large data sets, these operations become computationally expensive, as the algorithm must iterate through all observations before performing a single parameter update.

A more efficient approach is to sample smaller subsets of the data and perform the same operations on each subset, yielding noisier yet "good enough" estimates. This motivates a partial E-step strategy, where the E-step is performed on data subsets to compute subset-specific expected counts. Moreover, the M-step can be relaxed as in GEM, where parameters are updated incrementally using partial information—such as gradients or Hessians based on subset-specific expectations—rather than solving for an exact M-step solution. These two modifications—(a) performing partial E-steps and (b) updating parameters incrementally in the M-step—form the basis of the proposed algorithm.

3.1 Overview

The proposed algorithm consists of two stages to obtain the point estimates. In Stage 1, it uses the standard EM algorithm to move parameter estimates toward the neighborhood of the MLE. Once a pre-specified stability criterion is met, it transitions to Stage 2 to operate on data subsets to accelerate convergence. Also during Stage 2, it incrementally

constructs a cross-product approximation of the information matrix, which stabilizes upon convergence and is then used to compute standard errors.

3.1.1 Pattern-Based Representation of Item Response Data

The item response matrix for P individuals can be compactly represented by R unique response patterns and their associated frequencies. Let $\boldsymbol{u}_r = [u_{1r}, u_{2r}, \ldots, u_{Ir}]'$ denote the r-th pattern and f_r denote the frequency of $\boldsymbol{u}_r$. Previously, we have also defined $P_i(\eta_k)$ to be the probability of a correct response to item i given node η_k, and $h(\eta_k)$ be the prior standard normal density at node η_k.

3.2 Parameter Estimation: A Two-Stage Approach

3.2.1 Stage 1: Moving Iterates Toward the MLE

Recall that the standard EM algorithm is efficient in early iterations, making substantial improvements to parameter estimates. However, as the iterates approach the MLE, the algorithm continues to process the full dataset in each iteration while yielding diminishing improvements, making later updates computationally more expensive.

The proposed algorithm addresses this inefficiency by exploiting the early performance of standard EM and then switching to a more computationally efficient approach. To determine when to transition from full-data standard EM (Stage 1) to a subset-based modified EM (Stage 2), it uses a log-likelihood-based switching rule similar to Tian et al. (2013, p. 423). This rule detects when the iterates have "gotten over the hump"—that is, when they have moved past the initial large updates and roughly entered the neighborhood of the MLE. This ensures that major updates driven by full-data EM have taken place before transition, and allows Stage 2 to refine parameter estimates with lower computational and time costs. Pseudocode 1 provides an overview of Stage 1's implementation.

Pseudocode 1 Overview of Stage 1

$\boldsymbol{U}$: response pattern matrix based on the full data
$\boldsymbol{f}$: vector of pattern frequencies
$\boldsymbol{\eta}$: vector of quadrature nodes (default: 121 evenly spaced points in $[-6, 6]$)
`delta_thr`: threshold for Stage 2 transition (default: 0.99)
`LL`: observed (marginal) log-likelihood
`tol`: convergence threshold for parameter changes (default: 10^{-6})
`sample_size`: number of individuals in full dataset
Initialize $\boldsymbol{a}^{(0)} \leftarrow \mathbf{1}_I$, $\boldsymbol{c}^{(0)} \leftarrow \mathbf{0}_I$, $\texttt{LL}^{(j-1)} \leftarrow$ `None`
for each Iteration j in Stage 1 **do**
    $(\boldsymbol{r}_0^{(j)}, \boldsymbol{r}_1^{(j)}, \texttt{LL}^{(j)}) \leftarrow$ RUN_ESTEP$(\boldsymbol{a}^{(j-1)}, \boldsymbol{c}^{(j-1)}, \boldsymbol{U}, \boldsymbol{f}, \boldsymbol{\eta})$
    $(\boldsymbol{a}^{(j)}, \boldsymbol{c}^{(j)}) \leftarrow$ RUN_MSTEP_FULL$(\boldsymbol{a}^{(j-1)}, \boldsymbol{c}^{(j-1)}, \boldsymbol{r}_0^{(j)}, \boldsymbol{r}_1^{(j)}, \boldsymbol{\eta})$
    $\delta^{(j)} \leftarrow$ GET_EXP_DIFF$(\texttt{LL}^{(j)}, \texttt{LL}^{(j-1)}, \texttt{sample_size})$ ▷ See Equation 3.1
    `converged` $\leftarrow \max |\boldsymbol{a}^{(j)} - \boldsymbol{a}^{(j-1)}| \leq$ `tol` **and** $\max |\boldsymbol{c}^{(j)} - \boldsymbol{c}^{(j-1)}| \leq$ `tol`
    if `converged` **then**
        Exit Stage 1
    if running Stage 2 **and** $\delta^{(j)} \geq$ `delta_thr` **then**
        Exit Stage 1 (and enter Stage 2)

3.2.1.1 Adaptive Transition to Stage 2

The switching rule defines $\delta^{(j)}$ as an exponentiated difference between successive average log-likelihood values

$$\delta^{(j)} = \exp\left[-\frac{\texttt{LL}^{(j)} - \texttt{LL}^{(j-1)}}{N}\right] \tag{3.1}$$

where N is the sample size, and `LL` denotes the observed log-likelihood obtained from the E-step at Iteration j of Stage 1. When $\delta^{(j)}$ exceeds a predefined threshold (e.g., 0.99), the algorithm transitions to Stage 2. In the implementation, $\delta^{(j)}$ is set to 0 at the first iteration where $\texttt{LL}^{(j-1)}$ is not yet defined.

3.2.1.2 Early Stopping Condition

When only Stage 1 is run, convergence is determined based on the maximum absolute change in parameter estimates between iterations. Specifically, the algorithm checks

whether the largest absolute changes in A estimates and in C estimates both fall below a predefined threshold (e.g., 10^{-6}). Once this condition is satisfied, the algorithm exits Stage 1.

3.2.2 Stage 2: Accelerating Updates with Data Subsets

In Stage 2, the algorithm partitions the data into subsets and iterates through them to perform a partial E-step and a modified M-step. Pseudocode 2 provides an overview of Stage 2's implementation.

Pseudocode 2 Overview of Stage 2

`rnd_seed`: random seed for reproducible subset sampling
`subset_size`: number of individuals per subset
$\boldsymbol{Y}$: full item response data matrix
$\boldsymbol{\eta}$: vector of quadrature nodes (default: 121 evenly spaced points in $[-6, 6]$)
γ: fixed step size for M-step update (default: 0.5)
`tol`: convergence threshold for parameter changes (default: 10^{-6})
Given $\boldsymbol{a}^{(t-1)}$ and $\boldsymbol{c}^{(t-1)}$ from Stage 1
for each Iteration t in Stage 2 **do**
    `subsets` $\leftarrow$ CREATE_SUBSETS($\boldsymbol{Y}$, `rnd_seed`, `subset_size`)
    for each $(\boldsymbol{U}^{(s)}, \boldsymbol{f}^{(s)})$ in `subsets` **do**
        $(\boldsymbol{r}_0^{(t,s)}, \boldsymbol{r}_1^{(t,s)}, \mathrm{LL}^{(t,s)}) \leftarrow$ RUN_ESTEP($\boldsymbol{a}^{(t-1)}, \boldsymbol{c}^{(t-1)}, \boldsymbol{U}^{(s)}, \boldsymbol{f}^{(s)}, \boldsymbol{\eta}$)
        $(\boldsymbol{a}^{(t)}, \boldsymbol{c}^{(t)}) \leftarrow$ RUN_MSTEP_SUBSET($\boldsymbol{a}^{(t-1)}, \boldsymbol{c}^{(t-1)}, \boldsymbol{r}_0^{(t,s)}, \boldsymbol{r}_1^{(t,s)}, \boldsymbol{\eta}, \gamma$)
        $(\tilde{\boldsymbol{a}}^{(t)}, \tilde{\boldsymbol{c}}^{(t)}) \leftarrow$ UPDATE_SMA($\boldsymbol{a}^{(t)}, \boldsymbol{c}^{(t)}$) ▷ See Pseudocode 3
        `converged` $\leftarrow \max|\tilde{\boldsymbol{a}}^{(t)} - \tilde{\boldsymbol{a}}^{(t-1)}| \leq$ `tol` **and** $\max|\tilde{\boldsymbol{c}}^{(t)} - \tilde{\boldsymbol{c}}^{(t-1)}| \leq$ `tol`
        if `converged` **then**
            Break both inner and outer loops and exit Stage 2

3.2.2.1 Creating Data Subsets

Shuffling. The row indices of the full item response data data are shuffled to mimic random selection and minimize data-ordering effects (Goodfellow et al., 2016). Faster convergence has also been observed if the data subsets were randomly ordered for each iteration (Bengio, 2012).

Partitioning. The vector of shuffled indices is partitioned into M subsets, where each subset indexes the corresponding rows in the full data. For example, 5% of 10,000 translates to a subset size of 500 and a M of 20. In the case where the sample size is not divisible by the subset size, the remaining data will be its own subset. Each subset is then summarized into response pattern matrix $\boldsymbol{U}$ and frequency vector $\boldsymbol{f}$.

3.2.2.2 Partial E-Steps

The computation of the E-step follows the same procedure as in the standard EM algorithm. The difference is that each E-step is performed on a subset s, consisting of response pattern data $\boldsymbol{U}^{(s)}$ from $P^{(s)}$ individuals randomly drawn from the full set of P individuals. Each partial E-step yields subset-specific expected counts of 1 and 0 responses for I items across K quadrature nodes:

$$\tilde{r}_{1ik}^{(t,s)} = \sum_r f_r^{(s)} \cdot g\left(\eta_k \mid \boldsymbol{u}_r^{(s)}, \boldsymbol{\phi}^{(t)}\right) \cdot u_{ir}^{(s)} \tag{3.2}$$

$$\tilde{r}_{0ik}^{(t,s)} = \sum_r f_r^{(s)} \cdot g\left(\eta_k \mid \boldsymbol{u}_r^{(s)}, \boldsymbol{\phi}^{(t)}\right) \cdot \left(1 - u_{ir}^{(s)}\right) \tag{3.3}$$

where $f_r^{(s)}$ denotes the frequency of the r-th pattern in Subset s.

3.2.2.3 Modified M-Steps

Each partial E-step is followed by a modified M-step that performs a Newton-style update for Item i's parameters:

$$\boldsymbol{\phi}_i^{(t+1)} = \boldsymbol{\phi}_i^{(t)} + \gamma \left(\boldsymbol{H}_i^{(t)}\right)^{-1} \boldsymbol{g}_i^{(t)} \tag{3.4}$$

In Equation 3.4, $\boldsymbol{g}^{(t)}$ and $\boldsymbol{H}^{(t)}$ denote the gradient and Hessian of the expected complete data log-likelihood at Iteration t. The fixed step size γ controls the update magnitude.[1]

[1] An alternative expression of Equation 3.4 is minimizing the negative log-likelihood, which is the default behavior for optimizers such as `nlm()` or `optim()` in R and `scipy.optimize.minimize()` in Python.

Gradients of Expected Complete-Data Log-Likelihood. For Item i, the gradients of Equation 2.11 with respect to the item parameters are

$$\begin{aligned}\frac{\partial Q}{\partial a_i} &= \sum_k \left[\tilde{r}_{1ik} - (\tilde{r}_{1ik} + \tilde{r}_{0ik}) P_i(\eta_k)\right] \eta_k \\ &= \sum_k \left[\tilde{r}_{1ik} - \tilde{n}_{ik} P_i(\eta_k)\right] \eta_k \qquad (3.5) \\ \frac{\partial Q}{\partial c_i} &= \sum_k \left[\tilde{r}_{1ik} - (\tilde{r}_{1ik} + \tilde{r}_{0ik}) P_i(\eta_k)\right] \\ &= \sum_k \left[\tilde{r}_{1ik} - \tilde{n}_{ik} P_i(\eta_k)\right] \qquad (3.6)\end{aligned}$$

where $P_i(\eta_k)$ denotes the response function (Equation 2.1) evaluated at η_k, and $\tilde{n}_{ik} = \tilde{r}_{1ik} + \tilde{r}_{0ik}$ denotes the expected number of individuals at η_k.

Hessian of Expected Complete-Data Log-Likelihood. Elements of the Hessian matrix are given by

$$\frac{\partial^2 Q}{\partial a_i^2} = -\sum_k \tilde{n}_{ik} P_i(\eta_k) Q_i(\eta_k) \eta_k^2 \qquad (3.7)$$

$$\frac{\partial^2 Q}{\partial c_i^2} = -\sum_k \tilde{n}_{ik} P_i(\eta_k) Q_i(\eta_k) \qquad (3.8)$$

$$\frac{\partial^2 Q}{\partial a_i \partial c_i} = -\sum_k \tilde{n}_{ik} P_i(\eta_k) Q_i(\eta_k) \eta_k \qquad (3.9)$$

where $Q_i(\eta_k) = 1 - P_i(\eta_k)$, as defined in Equation 2.2.

Iterates Averaging. With unidimensional 2PL IRT models, the log-likelihood surface near the MLE is approximately quadratic. Within this region, M-step updates with a fixed step size tend to oscillate around the MLE. Instead of tuning the step size, the algorithm approximates the MLE by taking simple moving averages of recent parameter values to track the center of the oscillation. Averaging starts after a fixed warm-up period. Once the average stabilizes, it is no longer updated, and the most recent value

is retained. Stability is determined by whether the variance of recent averaged values, computed over a moving window, falls below a predefined threshold. Pseudocode 3 outlines two core functions used for averaging and checking stability.

Pseudocode 3 Simple Moving Average: `update_sma()` and `check_stable()`

Internal variables:
- `est_vals`: list of input estimates
- `center_vals`: list of moving averages
- `wait`: number of values before averaging starts (default: 30)
- `window_check`: number of recent averages used to check stability (default: 5)
- `tol`: threshold for stability check (default: 10^{-6})
- `stabilized`: boolean flag indicating stability

function CHECK_STABLE
    if length of `center_vals` $<$ `window_check` **then**
        return `False`
    $\delta \leftarrow$ variance of last `window_check` values in `center_vals`
    return $\delta \leq$ `tol`

function UPDATE_SMA(`est_new`)
    Append `est_new` to `est_vals`
    if `stabilized` **then**
        return last value in `center_vals`
    `center` $\leftarrow$ mean of `est_vals`
    Append `center` to `center_vals`
    if length of `est_vals` $<$ `wait` **then**
        return `est_new`
    `stabilized` $\leftarrow$ CHECK_STABLE
    return `center`

3.2.2.4 Early Stopping Condition

Stage 2 includes a convergence check similar to that used in Stage 1. Specifically, if the largest absolute changes in the smoothed A and C parameters fall below a predefined tolerance (e.g., 10^{-6}), the algorithm exits Stage 2.

3.3 Standard Error Estimation

The EM framework does not produce the parameter information matrix or error covariance matrix, and hence provides no standard errors (SEs) upon convergence.

3.3.1 Computing SEs Based on the Empirical Cross-Product Information Matrix

One solution is the empirical cross-product (XPD) approximation, which computes the information matrix as the outer product of the score function of the marginal log-likelihood. In the proposed algorithm, the XPD matrix is computed incrementally during Stage 2 by accumulating score outer-products over disjoint data subsets. Upon convergence, the inverse of the final XPD matrix yields the parameter error covariance matrix. SEs are then obtained as the square roots of the diagonal elements of the covariance matrix. The subsections below present key numeric results. Pseudocode 4 outlines the implementation for accumulating the XPD information matrix and computing standard errors.

3.3.1.1 Gradients of Marginal Log-Likelihood

The gradients with respect to Item i's parameters are:

$$\frac{\partial l_r}{\partial a_i} = \sum_{k=1}^{K} g_{kr} \left[u_{ir} - P_i(\eta_k) \right] \eta_k \tag{3.10}$$

$$\frac{\partial l_r}{\partial c_i} = \sum_{k=1}^{K} g_{kr} \left[u_{ir} - P_i(\eta_k) \right] \tag{3.11}$$

where g_{kr} denotes the posterior probability $g(\eta_k \mid \boldsymbol{u}_r, \boldsymbol{\phi})$, defined analogously to Equation 2.7.

3.3.1.2 XPD Information Matrix

Let the sample gradient vector for Pattern r be

$$v_r = \left[\frac{\partial l_r}{\partial a_1}, \ \frac{\partial l_r}{\partial c_1}, \ \ldots, \ \frac{\partial l_r}{\partial a_I}, \ \frac{\partial l_r}{\partial c_I} \right]' \tag{3.12}$$

The XPD information matrix is then approximated by

$$\mathrm{I}_{\mathrm{xpd}}(\boldsymbol{\phi}) = \sum_{r=1}^{R} f_r \left[v_r \left(v_r \right)' \right] \tag{3.13}$$

Pseudocode 4 Accumulating the XPD Information Matrix and Computing SEs

```
get_se: boolean flag indicating whether SEs should be computed
Initialize XPD ← 0_{2I×2I}
for each iteration in Stage 2 do
    Create data subsets
    for each subset do
        Run one partial E-step
        if get_se then
            XPD ← XPD + COMPUTE_XPD(U^(s), f^(s), η)
        Run one modified M-step
        if converged then
            Break both inner and outer loops and exit Stage 2
if get_se then
    COV ← inverse(XPD)
    SEs ← sqrt(diag(COV))
```

3.3.2 Computing SEs Based on the Fisher Expected Information Matrix

To evaluate the bias and RMSE of XPD-based SE estimates in later simulation studies, an additional set of SEs is computed from the Fisher expected information (FIS) matrix evaluated at the true parameter values. These serve as a theoretical benchmark or gold standard for comparison.

3.3.2.1 FIS Matrix

The FIS is defined as

$$\mathcal{F}(\boldsymbol{\phi}) = \mathcal{J}(\boldsymbol{\phi})' \left\{ \text{diag}[\boldsymbol{\pi}(\boldsymbol{\phi})] \right\}^{-1} \mathcal{J}(\boldsymbol{\phi}) \tag{3.14}$$

In Equation 3.14, $\boldsymbol{\phi}$ is a vector collecting all item parameters. The matrix $\mathcal{J}(\boldsymbol{\phi})$ contains the gradients of the marginal log-likelihood, where each row corresponds to a response pattern and each column to a model parameter. Each non-zero entry in the diagonal matrix $\text{diag}[\boldsymbol{\pi}(\boldsymbol{\phi})]$ shows the probability of observing a given pattern under the 2PL IRT model. The number of unique response patterns increases exponentially with the number of items, following 2^I where I is the number of items. For example, 12 binary items yield $2^{12} = 4096$ possible patterns.

In simulation studies, FIS can be evaluated at the true parameter values, denoted by $\mathcal{F}(\boldsymbol{\phi}_0)$. Given a sample size of N, the matrix $[N\mathcal{F}(\boldsymbol{\phi}_0)]^{-1}$ is the parameter error covariance matrix. The square roots of its diagonal elements provide the gold-standard SEs (Cai, 2008; Tian et al., 2013). Appendix A provides a R function for computing FIS-based SEs using output from flexMIRT 3.72 (Cai, 2024).

CHAPTER 4

Simulation Studies

4.1 Simulation Study 1

The first simulation study evaluates the effectiveness of the proposed algorithm in reducing time to convergence and recovering item parameters under different subset sizes and item set lengths.

4.1.1 Simulation Conditions

Study 1 varied two factors: (a) percentage subset size (10%, 20%, 30%, 50%, or 100% of the total sample size) and (b) number of items (18 or 36). When the percentage was 100%, item response data from all individuals were used, and the standard EM algorithm was applied by running only Stage 1, without the adaptive transition described in Chapter 3. Each condition was replicated $R = 100$ times; all replications were run until convergence.

For each replication, binary responses were simulated for $N = 60,000$ individuals under a unidimensional 2PL IRT model, given true item parameters and a standard normal distribution for the latent variable. Table 4.1 summarizes the simulation conditions. The next section describes how the true item parameters were generated.

4.1.2 Grid Sampling of True Item Parameters

The true A (slope) and B (difficulty) parameter values were randomly sampled at three levels: low, medium, high. The levels of A were selected based on literature suggesting that reasonably "good" values tend to fall between 0.8 and 2.5 (Baker & Kim, 2004; de

Table 4.1. Study 1: Simulation Conditions ($N = 60,000$)

Condition	Subset %	Subset size	No. of items
1	10	6,000	18
2	10	6,000	36
3	20	12,000	18
4	20	12,000	36
5	30	18,000	18
6	30	18,000	36
7	50	30,000	18
8	50	30,000	36
9	100	60,000	18
10	100	60,000	36

Ayala, 2009). The levels of B were selected considering that approximately 99.73% of cases fall within ± 3 standard deviations of a standard normal distribution. Table 4.2 shows the sampling ranges for these levels.

These levels formed a 3-by-3 grid sampling space. Within each cell of the grid, an equal number of item parameters was sampled from an uniform distribution based on the specified ranges. For example, with 18 items, two items from the Low A and Low B cell were sampled, and this process was repeated for the remaining eight cells. Similarly, for the 36-item case, six items were sampled per cell.

Table 4.2. Study 1: Level Ranges for Item Parameters

Item parameter	Low	Medium	High
A (discrimination/slope)	[0.50, 1.50)	[1.50, 2.50)	[2.50, 3.50]
B (difficulty)	[-3.00, -1.50)	[-1.50, 1.50)	[1.50, 3.00]
C (intercept)	[-10.50, -3.75)	[-3.75, 3.75)	[3.75, 10.50]

The true A and B values were used to compute the true C (intercept) values using the formula $C = -A \times B$. The full range of possible C values, $[-10.50, 10.50]$, was derived from the outer bounds of $A \in [0.50, 3.50]$ and $B \in [-3.00, 3.00]$. The cutoffs at ± 3.75 divide this range into three approximately equal-width intervals for use in descriptive summaries and subgroup analyses. As a reminder, the A and C parameters are those being estimated. Table 4.3 and 4.4 present the true A, B, and C parameters used in the 18-item and 36-item conditions.

Table 4.3. Study 1: True Item Parameters for 18-Item Conditions

	True value			Level		
Item	A	B	C	A	B	C
1	0.64	−1.67	1.06	low	low	med
2	1.43	−2.33	3.34	low	low	med
3	0.89	−0.73	0.65	low	med	med
4	1.16	−0.02	0.03	low	med	med
5	1.46	2.70	−3.96	low	high	low
6	0.96	2.70	−2.58	low	high	med
7	1.54	−1.85	2.85	med	low	med
8	1.50	−2.56	3.85	med	low	high
9	2.11	1.24	−2.62	med	med	med
10	1.80	−0.75	1.36	med	med	med
11	2.17	2.98	−6.46	med	high	low
12	1.97	1.68	−3.32	med	high	med
13	3.42	−1.58	5.40	high	low	high
14	2.78	−2.22	6.17	high	low	high
15	2.65	−1.46	3.87	high	med	high
16	2.82	1.47	−4.16	high	med	low
17	3.01	2.81	−8.48	high	high	low
18	2.57	1.93	−4.95	high	high	low

Note. $C = -A \times B$. Parameter levels are based on cutoffs in Table 4.2.

4.1.3 Evaluation Criteria

The proposed algorithm was evaluated using two sets of criteria: reduction in time to convergence and parameter recovery performance. The latter includes bias and root mean square error (RMSE).

Time to Convergence

$$\bar{\mathrm{T}}_m = \frac{\sum_{r=1}^{R} \mathrm{Time}_r}{R} \tag{4.1}$$

$$\%\ \text{Time Change} = \frac{\bar{\mathrm{T}}_m - \bar{\mathrm{T}}_f}{\bar{\mathrm{T}}_f} \times 100 \tag{4.2}$$

where $\bar{\mathrm{T}}_f$ is the mean time based on the full-data standard EM, and $\bar{\mathrm{T}}_m$ is the mean time based on the proposed algorithm. All times are in seconds. r denotes the rth replication,

Table 4.4. Study 1: True Item Parameters for 36-Item Conditions

	True value			Level		
Item	A	B	C	A	B	C
1	0.64	−1.67	1.06	low	low	med
2	1.43	−2.33	3.34	low	low	med
3	0.89	−2.61	2.32	low	low	med
4	1.16	−2.26	2.62	low	low	med
5	1.46	0.90	−1.32	low	med	med
6	0.96	0.90	−0.86	low	med	med
7	0.54	0.81	−0.44	low	med	med
8	0.50	−0.62	0.31	low	med	med
9	1.11	2.87	−3.19	low	high	med
10	0.80	1.87	−1.50	low	high	med
11	1.17	2.98	−3.48	low	high	med
12	0.97	1.68	−1.63	low	high	med
13	2.42	−1.58	3.82	med	low	high
14	1.78	−2.22	3.95	med	low	high
15	1.65	−2.98	4.93	med	low	high
16	1.82	−1.51	2.76	med	low	med
17	2.01	1.13	−2.27	med	med	med
18	1.57	−0.65	1.01	med	med	med
19	1.97	0.79	−1.55	med	med	med
20	2.42	−0.32	0.78	med	med	med
21	2.43	2.25	−5.46	med	high	low
22	2.30	2.84	−6.55	med	high	low
23	1.98	2.77	−5.49	med	high	low
24	1.78	2.01	−3.57	med	high	med
25	2.61	−2.22	5.77	high	low	high
26	3.15	−1.65	5.20	high	low	high
27	2.60	−2.53	6.59	high	low	high
28	2.58	−1.59	4.11	high	low	high
29	3.46	−0.93	3.20	high	med	med
30	3.45	−0.17	0.58	high	med	med
31	2.76	0.51	−1.41	high	med	med
32	2.88	1.46	−4.19	high	med	low
33	3.17	2.99	−9.46	high	high	low
34	3.47	1.77	−6.16	high	high	low
35	2.94	2.51	−7.38	high	high	low
36	3.15	2.07	−6.54	high	high	low

Note. $C = -A \times B$. Parameter levels are based on cutoffs in Table 4.2.

and R is the total number of replications.

Parameter Recovery

$$\text{Bias}\left(\hat{\phi}\right) = \frac{\sum_{r=1}^{R}\left(\hat{\phi}_r - \phi\right)}{R} \tag{4.3}$$

$$\text{RMSE}\left(\hat{\phi}\right) = \sqrt{\frac{\sum_{r=1}^{R}\left(\hat{\phi}_r - \phi\right)^2}{R}} \tag{4.4}$$

where ϕ is the true parameter generated following Section 4.1.2, and $\hat{\phi}_r$ is the parameter estimate from the rth replication.

4.2 Simulation Study 2

The second study evaluates the applicability and performance of SE estimation under the proposed algorithm. The methods for computing SEs—based on the Fisher expected information matrix (FIS) and the empirical cross-product approximation (XPD)—are described in Section 3.3.

4.2.1 Simulation Conditions

Study 2 varied the percentage subset size (20%, 30%, 50%, or 100%).[1] Each condition had data from 60,000 individuals ($N = 60,000$) and 1,000 replications ($R = 1,000$). All replications were run until convergence. Table 4.5 summarizes the four simulation conditions.

Table 4.6 presents the true parameter values and their FIS-based SEs. Given the true parameters, dichotomously scored item responses to 12 items were simulated for 60,000 individuals in each replication, assuming a unidimensional 2PL IRT model and a

[1]The 10% condition was omitted due to the high variability in parameter estimates observed in Simulation 1.

standard normal distribution for the latent variable.

Table 4.5. Study 2: Simulation Conditions ($N = 60,000$)

Condition	Subset %	Subset size
1	20	12,000
2	30	18,000
3	50	30,000
4	100	60,000

Table 4.6. Study 2: True Item Parameters and Standard Errors

Item	A	SE(a)	C	SE(c)
1	0.68	0.015	1.60	0.014
2	1.27	0.022	2.24	0.021
3	1.56	0.026	2.39	0.024
4	1.22	0.019	1.57	0.016
5	1.38	0.020	1.14	0.015
6	1.80	0.025	0.21	0.014
7	1.33	0.018	−0.16	0.012
8	2.02	0.029	−1.19	0.018
9	1.16	0.018	−1.23	0.014
10	1.11	0.019	−1.70	0.016
11	0.94	0.017	−1.66	0.015
12	0.80	0.017	−1.83	0.015

Note. Given $B = -C/A$, the range of B values (item difficulties) is from -2.35 to 2.29.

4.2.2 Evaluation Criteria

For each true parameter, the FIS-based standard error $\mathrm{SE}_{\mathrm{fis}}(\phi)$ was computed. For each condition, the mean of XPD-based estimated SEs across replications was computed. The following metrics quantify the difference between $\mathrm{SE}_{\mathrm{fis}}(\phi)$ and the estimated SE:

$$\mathrm{Bias}\left[\mathrm{SE}(\phi)\right] = \frac{\sum_{r=1}^{R} \mathrm{SE}(\hat{\phi}_r)}{R} - \mathrm{SE}_{\mathrm{fis}}(\phi) \tag{4.5}$$

$$\mathrm{RMSE}\left[\mathrm{SE}(\phi)\right] = \sqrt{\frac{\sum_{r=1}^{R} \left[\mathrm{SE}(\hat{\phi}_r) - \mathrm{SE}_{\mathrm{fis}}(\phi)\right]^2}{R}} \tag{4.6}$$

CHAPTER 5

Results and Discussion

5.1 Study 1 Results

5.1.1 Reduction in Time to Convergence

The results presented in Table 5.1 compare the time to convergence under the proposed algorithm against the standard EM algorithm across two item conditions: 18 items and 36 items.[1] In the 18-item condition, the proposed algorithm achieved a lower time to convergence than standard EM only when using the 10% subset, with a 10.39% reduction in time. As the subset size increased, the time to convergence also rose. At 20%, the time was slightly higher, and this trend continued at 30% and was most pronounced at 50% where the time to convergence was nearly double that of standard EM.

In contrast, the proposed algorithm consistently had lower convergence times than standard EM in the 36-item condition. With the 10% subset size, the time was reduced by 78.68%, and with the 50% subset size, it remained 24.14% lower than that of the standard EM. These findings suggest that the proposed algorithm is more efficient when applied to larger item sets. This may be because, as the number of items increases, the cost of summing over response patterns in each EM iteration becomes more pronounced. In such cases, a subset-based approach reduces the per-iteration computational cost. This is particularly advantageous when the response pattern matrix begins to resemble the full item response matrix. With larger item sets, individuals are much more likely to produce unique or near-unique response patterns, leading to a substantial increase in

[1] Values in Table 5.1 and later tables are shown with rounding; minor differences may occur.

the size of the response pattern matrix.

Table 5.1. Study 1: Mean and SD of Time to Convergence in Seconds

	18 Items			36 Items		
Subset %	*M*	*SD*	Change %	*M*	*SD*	Change %
10	2.67	0.48	−10.39	26.02	3.17	−78.68
20	3.10	0.40	4.07	40.02	1.43	−67.21
30	3.79	0.58	27.16	51.16	4.79	−58.07
50	5.78	0.20	93.92	92.58	10.83	−24.14
Standard EM	2.98	0.07	0.00	122.03	12.62	0.00

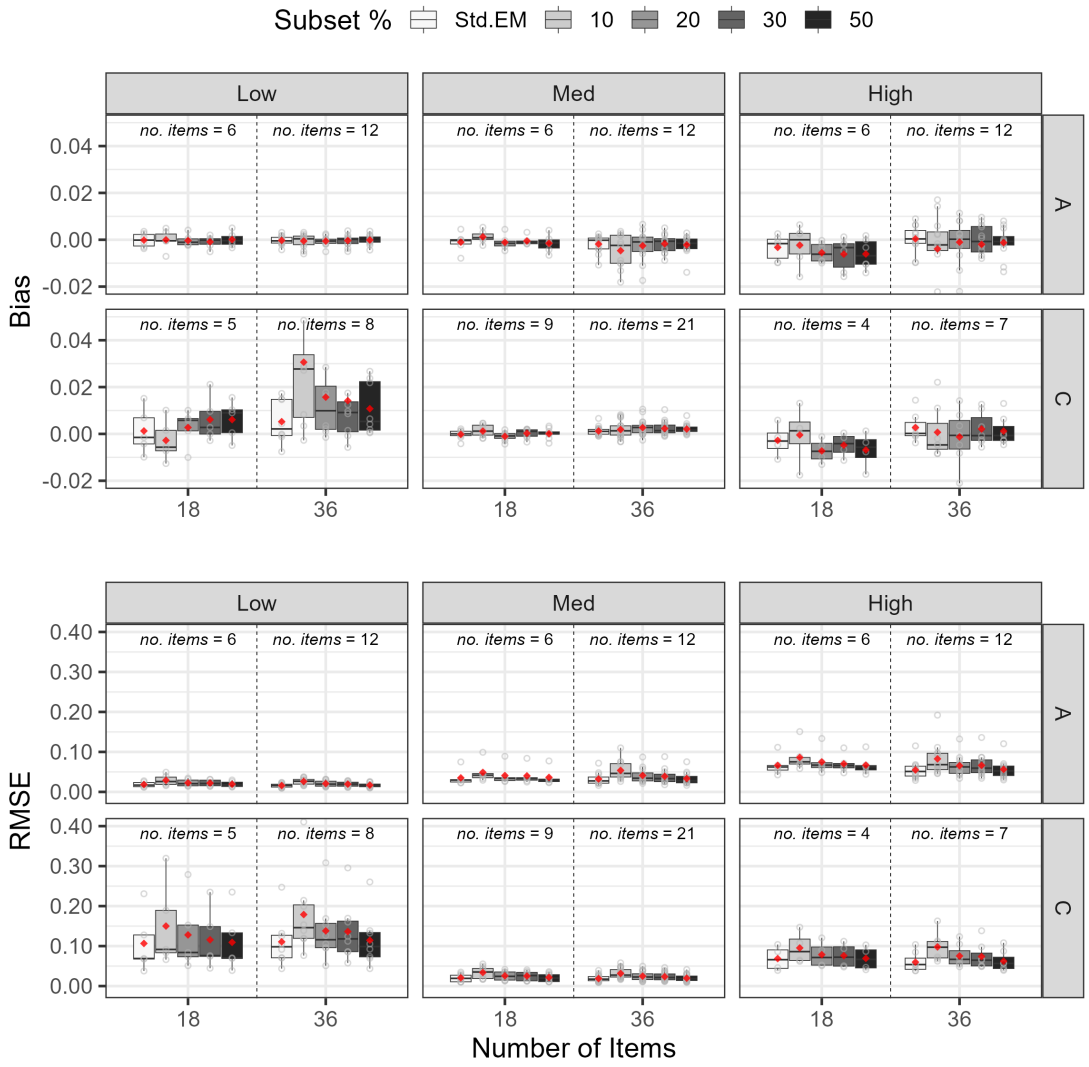
5.1.2 Bias and RMSE of Estimated Item Parameters

Figure 5.1 presents bias and RMSE values, broken down by subset size, parameter type, and parameter magnitude levels defined in Section 4.1.2. In the figure, gray hollow circles show individual values, red diamonds indicate the mean for each condition, and the number of items whose A or C parameters were categorized as low, medium (med), or high is noted in each panel.

Several observations can be drawn from this figure. First, subset sizes of 20% and above had bias and RMSE values much closer to those of standard EM. Second, regardless of the number of items, the true slope parameters (A) were easier to recover than the intercept parameters (C), as indicated by the lower and less dispersed bias and RMSE. Third, parameters with extreme values were more challenging to estimate compared to parameters with moderate values. This was particularly evident when both the slope (A) and difficulty (B) parameters were high, resulting in extreme and hard-to-estimate intercept parameters (C). Estimation in these cases was challenging because the items were highly sensitive to differences in the latent score, but only for individuals at the extremes of the distribution, where few are located under a standard normal distribution.

To complement results shown in Figure 5.1, Tables 5.2 to 5.6 report mean estimate, standard deviation, bias, and RMSE by individual parameter under the 36-item condition and five different subset sizes. All values were computed using unrounded data but are

Figure 5.1. Study 1: Distribution of Bias and RMSE of Parameter Estimates



reported to two decimal places for readability. The patterns described for Figure 5.1 and observed under the 18-item condition are similarly reflected here. Larger bias and RMSE values tended to occur for parameters with extreme true values, particularly among the C parameters. For example, Item 33's parameters ($A_{true} = 3.17, C_{true} = -9.46, B_{true} = 2.98$) were the hardest to estimate accurately and precisely.

Table 5.2. Study 1 (10% Subset Size + 36 Items): Bias and RMSE by Parameter

	Parameter A					Parameter C				
Item	True value	M_{est}	SD_{est}	Bias	RMSE	True value	M_{est}	SD_{est}	Bias	RMSE
1	0.64	0.64	0.02	0.00	0.02	1.06	1.06	0.02	0.00	0.02
2	1.43	1.44	0.04	0.00	0.04	3.34	3.34	0.05	0.00	0.05
3	0.89	0.89	0.03	0.00	0.03	2.32	2.32	0.03	0.00	0.03
4	1.16	1.15	0.03	−0.01	0.03	2.62	2.62	0.03	0.00	0.03
5	1.46	1.46	0.03	0.00	0.03	−1.32	−1.32	0.02	0.00	0.02
6	0.96	0.96	0.02	0.00	0.02	−0.86	−0.86	0.02	0.00	0.02
7	0.54	0.54	0.02	0.00	0.02	−0.44	−0.44	0.02	0.00	0.02
8	0.50	0.50	0.02	0.00	0.02	0.31	0.31	0.02	0.00	0.02
9	1.11	1.10	0.04	−0.01	0.04	−3.19	−3.18	0.04	0.01	0.05
10	0.80	0.80	0.02	0.00	0.02	−1.50	−1.50	0.02	0.00	0.02
11	1.17	1.17	0.03	0.00	0.03	−3.48	−3.48	0.05	0.00	0.05
12	0.97	0.97	0.02	0.00	0.02	−1.63	−1.63	0.02	0.00	0.02
13	2.42	2.41	0.05	−0.01	0.05	3.82	3.81	0.06	0.00	0.06
14	1.78	1.78	0.05	0.00	0.05	3.95	3.96	0.07	0.01	0.07
15	1.65	1.65	0.07	−0.01	0.07	4.93	4.92	0.11	−0.01	0.11
16	1.82	1.83	0.04	0.00	0.04	2.76	2.76	0.04	0.00	0.04
17	2.01	2.02	0.04	0.00	0.04	−2.27	−2.27	0.03	0.00	0.03
18	1.57	1.57	0.03	0.00	0.03	1.01	1.01	0.02	0.00	0.02
19	1.97	1.97	0.03	0.00	0.03	−1.55	−1.54	0.03	0.00	0.03
20	2.42	2.42	0.04	0.00	0.04	0.78	0.78	0.03	0.00	0.03
21	2.43	2.42	0.07	−0.01	0.07	−5.46	−5.44	0.12	0.03	0.12
22	2.30	2.28	0.11	−0.02	0.11	−6.55	−6.52	0.20	0.03	0.20
23	1.98	1.97	0.08	−0.01	0.08	−5.49	−5.47	0.12	0.03	0.12
24	1.78	1.78	0.04	0.00	0.04	−3.57	−3.58	0.06	0.00	0.06
25	2.61	2.61	0.07	0.00	0.07	5.77	5.77	0.11	0.00	0.11
26	3.15	3.16	0.07	0.01	0.07	5.20	5.22	0.10	0.02	0.10
27	2.60	2.60	0.09	0.00	0.09	6.59	6.59	0.16	−0.01	0.16
28	2.58	2.57	0.05	0.00	0.05	4.11	4.10	0.07	0.00	0.07
29	3.46	3.46	0.06	0.01	0.06	3.20	3.21	0.05	0.01	0.05
30	3.45	3.47	0.04	0.02	0.05	0.58	0.59	0.03	0.01	0.03
31	2.76	2.76	0.05	−0.01	0.05	−1.41	−1.40	0.03	0.01	0.03
32	2.88	2.87	0.07	0.00	0.07	−4.19	−4.18	0.08	0.01	0.08
33	3.17	3.12	0.19	−0.05	0.19	−9.46	−9.36	0.40	0.10	0.41
34	3.47	3.48	0.10	0.00	0.10	−6.16	−6.15	0.14	0.00	0.14
35	2.94	2.92	0.11	−0.02	0.11	−7.38	−7.33	0.21	0.05	0.21
36	3.15	3.15	0.10	0.00	0.10	−6.54	−6.54	0.15	0.00	0.15

Table 5.3. Study 1 (20% Subset Size + 36 Items): Bias and RMSE by Parameter

	Parameter A					Parameter C				
Item	True value	M_{est}	SD_{est}	Bias	RMSE	True value	M_{est}	SD_{est}	Bias	RMSE
1	0.64	0.64	0.01	0.00	0.01	1.06	1.06	0.02	0.00	0.02
2	1.43	1.43	0.03	0.00	0.03	3.34	3.34	0.04	0.00	0.04
3	0.89	0.89	0.02	0.00	0.02	2.32	2.32	0.02	0.00	0.02
4	1.16	1.16	0.03	0.00	0.03	2.62	2.62	0.03	0.01	0.03
5	1.46	1.46	0.02	0.00	0.02	−1.32	−1.32	0.02	0.00	0.02
6	0.96	0.95	0.01	0.00	0.01	−0.86	−0.86	0.01	0.00	0.01
7	0.54	0.54	0.01	0.00	0.01	−0.44	−0.44	0.01	0.00	0.01
8	0.50	0.50	0.01	0.00	0.01	0.31	0.31	0.01	0.00	0.01
9	1.11	1.11	0.03	−0.01	0.03	−3.19	−3.18	0.04	0.01	0.04
10	0.80	0.80	0.02	0.00	0.02	−1.50	−1.50	0.02	0.00	0.02
11	1.17	1.17	0.03	0.00	0.03	−3.48	−3.47	0.04	0.00	0.04
12	0.97	0.97	0.02	0.00	0.02	−1.63	−1.63	0.02	0.00	0.02
13	2.42	2.41	0.04	−0.01	0.04	3.82	3.81	0.05	0.00	0.05
14	1.78	1.78	0.04	0.00	0.04	3.95	3.96	0.05	0.01	0.05
15	1.65	1.65	0.04	0.00	0.04	4.93	4.92	0.07	−0.01	0.07
16	1.82	1.83	0.03	0.01	0.03	2.76	2.77	0.03	0.00	0.03
17	2.01	2.02	0.03	0.00	0.03	−2.27	−2.28	0.03	0.00	0.03
18	1.57	1.57	0.02	0.00	0.02	1.01	1.02	0.02	0.00	0.02
19	1.97	1.97	0.03	0.00	0.03	−1.55	−1.54	0.02	0.01	0.02
20	2.42	2.42	0.03	0.00	0.03	0.78	0.78	0.02	0.00	0.02
21	2.43	2.41	0.06	−0.02	0.06	−5.46	−5.44	0.09	0.03	0.10
22	2.30	2.29	0.09	−0.01	0.09	−6.55	−6.53	0.15	0.02	0.15
23	1.98	1.98	0.06	0.00	0.06	−5.49	−5.49	0.10	0.00	0.10
24	1.78	1.78	0.03	0.00	0.03	−3.57	−3.57	0.04	0.00	0.04
25	2.61	2.61	0.07	0.00	0.07	5.77	5.77	0.11	0.00	0.10
26	3.15	3.16	0.06	0.01	0.06	5.20	5.21	0.07	0.01	0.07
27	2.60	2.59	0.07	−0.01	0.07	6.59	6.57	0.12	−0.02	0.12
28	2.58	2.58	0.04	0.00	0.04	4.11	4.11	0.06	0.00	0.06
29	3.46	3.47	0.06	0.01	0.06	3.20	3.21	0.05	0.01	0.05
30	3.45	3.47	0.04	0.01	0.04	0.58	0.58	0.02	0.00	0.02
31	2.76	2.76	0.03	0.00	0.03	−1.41	−1.41	0.02	0.00	0.02
32	2.88	2.88	0.05	0.00	0.05	−4.19	−4.19	0.05	0.00	0.05
33	3.17	3.15	0.13	−0.02	0.13	−9.46	−9.40	0.30	0.06	0.31
34	3.47	3.47	0.07	0.00	0.07	−6.16	−6.15	0.11	0.00	0.11
35	2.94	2.93	0.09	−0.01	0.09	−7.38	−7.36	0.17	0.02	0.17
36	3.15	3.15	0.08	0.00	0.08	−6.54	−6.54	0.13	0.00	0.13

Table 5.4. Study 1 (30% Subset Size + 36 Items): Bias and RMSE by Parameter

	Parameter A					Parameter C				
Item	True value	M_{est}	SD_{est}	Bias	RMSE	True value	M_{est}	SD_{est}	Bias	RMSE
1	0.64	0.64	0.01	0.00	0.01	1.06	1.06	0.01	0.00	0.01
2	1.43	1.43	0.03	0.00	0.03	3.34	3.34	0.03	0.00	0.03
3	0.89	0.89	0.02	0.00	0.02	2.32	2.32	0.02	0.00	0.02
4	1.16	1.16	0.02	0.00	0.02	2.62	2.62	0.03	0.00	0.03
5	1.46	1.47	0.02	0.00	0.02	−1.32	−1.32	0.02	0.00	0.02
6	0.96	0.96	0.02	0.00	0.02	−0.86	−0.86	0.01	0.00	0.01
7	0.54	0.54	0.01	0.00	0.01	−0.44	−0.44	0.01	0.00	0.01
8	0.50	0.50	0.01	0.00	0.01	0.31	0.31	0.01	0.00	0.01
9	1.11	1.11	0.03	−0.01	0.03	−3.19	−3.18	0.03	0.01	0.03
10	0.80	0.80	0.01	0.00	0.01	−1.50	−1.50	0.02	0.00	0.02
11	1.17	1.16	0.03	0.00	0.03	−3.48	−3.48	0.04	0.00	0.04
12	0.97	0.96	0.02	0.00	0.02	−1.63	−1.63	0.02	0.00	0.02
13	2.42	2.41	0.04	0.00	0.04	3.82	3.82	0.05	0.00	0.05
14	1.78	1.78	0.03	0.01	0.03	3.95	3.96	0.05	0.01	0.05
15	1.65	1.65	0.04	0.00	0.04	4.93	4.93	0.07	0.00	0.06
16	1.82	1.83	0.03	0.00	0.03	2.76	2.76	0.03	0.00	0.03
17	2.01	2.01	0.03	0.00	0.03	−2.27	−2.27	0.02	0.00	0.02
18	1.57	1.57	0.02	0.00	0.02	1.01	1.02	0.02	0.00	0.02
19	1.97	1.97	0.03	0.00	0.03	−1.55	−1.54	0.02	0.00	0.02
20	2.42	2.42	0.03	−0.01	0.03	0.78	0.78	0.02	0.00	0.02
21	2.43	2.42	0.06	−0.01	0.06	−5.46	−5.45	0.09	0.01	0.09
22	2.30	2.30	0.09	−0.01	0.09	−6.55	−6.54	0.16	0.01	0.16
23	1.98	1.98	0.05	0.00	0.05	−5.49	−5.50	0.09	0.00	0.09
24	1.78	1.78	0.03	0.00	0.03	−3.57	−3.57	0.04	0.00	0.04
25	2.61	2.61	0.06	0.00	0.06	5.77	5.77	0.10	0.00	0.10
26	3.15	3.16	0.05	0.01	0.05	5.20	5.21	0.07	0.01	0.07
27	2.60	2.60	0.08	−0.01	0.08	6.59	6.59	0.14	−0.01	0.14
28	2.58	2.58	0.04	0.00	0.04	4.11	4.11	0.05	0.00	0.05
29	3.46	3.47	0.06	0.01	0.06	3.20	3.21	0.05	0.01	0.05
30	3.45	3.46	0.04	0.01	0.05	0.58	0.58	0.02	0.00	0.02
31	2.76	2.76	0.03	0.00	0.03	−1.41	−1.40	0.02	0.01	0.02
32	2.88	2.87	0.05	0.00	0.05	−4.19	−4.18	0.06	0.01	0.06
33	3.17	3.14	0.13	−0.03	0.14	−9.46	−9.39	0.29	0.07	0.30
34	3.47	3.48	0.09	0.01	0.09	−6.16	−6.16	0.12	−0.01	0.12
35	2.94	2.93	0.09	−0.01	0.09	−7.38	−7.36	0.17	0.02	0.17
36	3.15	3.15	0.07	0.00	0.07	−6.54	−6.54	0.11	0.00	0.11

Table 5.5. Study 1 (50% Subset Size + 36 Items): Bias and RMSE by Parameter

	Parameter A					Parameter C				
Item	True value	M_{est}	SD_{est}	Bias	RMSE	True value	M_{est}	SD_{est}	Bias	RMSE
1	0.64	0.64	0.01	0.00	0.01	1.06	1.06	0.01	0.00	0.01
2	1.43	1.44	0.02	0.00	0.02	3.34	3.34	0.03	0.00	0.03
3	0.89	0.89	0.02	0.00	0.02	2.32	2.32	0.02	0.00	0.02
4	1.16	1.16	0.02	0.00	0.02	2.62	2.62	0.02	0.00	0.02
5	1.46	1.47	0.02	0.00	0.02	−1.32	−1.32	0.02	0.00	0.02
6	0.96	0.96	0.01	0.00	0.01	−0.86	−0.86	0.01	0.00	0.01
7	0.54	0.54	0.01	0.00	0.01	−0.44	−0.44	0.01	0.00	0.01
8	0.50	0.50	0.01	0.00	0.01	0.31	0.31	0.01	0.00	0.01
9	1.11	1.11	0.03	0.00	0.03	−3.19	−3.18	0.03	0.00	0.03
10	0.80	0.80	0.01	0.00	0.01	−1.50	−1.50	0.01	0.00	0.01
11	1.17	1.16	0.02	0.00	0.02	−3.48	−3.47	0.03	0.00	0.03
12	0.97	0.97	0.02	0.00	0.02	−1.63	−1.63	0.01	0.00	0.01
13	2.42	2.41	0.03	0.00	0.03	3.82	3.82	0.04	0.00	0.04
14	1.78	1.78	0.03	0.00	0.03	3.95	3.95	0.04	0.01	0.04
15	1.65	1.65	0.04	0.00	0.04	4.93	4.92	0.06	0.00	0.06
16	1.82	1.83	0.02	0.00	0.02	2.76	2.76	0.02	0.00	0.03
17	2.01	2.01	0.02	0.00	0.02	−2.27	−2.27	0.02	0.00	0.02
18	1.57	1.57	0.02	0.00	0.02	1.01	1.02	0.01	0.00	0.01
19	1.97	1.97	0.02	0.00	0.02	−1.55	−1.54	0.02	0.00	0.02
20	2.42	2.42	0.02	0.00	0.02	0.78	0.78	0.02	0.00	0.02
21	2.43	2.42	0.05	−0.01	0.05	−5.46	−5.44	0.07	0.02	0.07
22	2.30	2.30	0.08	0.00	0.08	−6.55	−6.54	0.13	0.00	0.13
23	1.98	1.98	0.05	0.00	0.05	−5.49	−5.49	0.07	0.00	0.07
24	1.78	1.77	0.03	0.00	0.03	−3.57	−3.57	0.03	0.00	0.03
25	2.61	2.60	0.06	0.00	0.06	5.77	5.77	0.09	0.00	0.09
26	3.15	3.16	0.04	0.01	0.05	5.20	5.21	0.06	0.01	0.06
27	2.60	2.60	0.06	0.00	0.06	6.59	6.59	0.11	0.00	0.11
28	2.58	2.58	0.03	0.00	0.03	4.11	4.11	0.05	0.00	0.05
29	3.46	3.46	0.05	0.00	0.05	3.20	3.21	0.04	0.01	0.04
30	3.45	3.46	0.04	0.01	0.04	0.58	0.58	0.02	0.00	0.02
31	2.76	2.76	0.03	0.00	0.03	−1.41	−1.41	0.02	0.00	0.02
32	2.88	2.88	0.04	0.00	0.04	−4.19	−4.19	0.04	0.00	0.04
33	3.17	3.16	0.12	−0.01	0.12	−9.46	−9.44	0.26	0.02	0.26
34	3.47	3.47	0.06	0.00	0.06	−6.16	−6.15	0.09	0.01	0.09
35	2.94	2.93	0.07	−0.01	0.07	−7.38	−7.35	0.14	0.03	0.14
36	3.15	3.15	0.07	0.00	0.07	−6.54	−6.54	0.11	0.00	0.11

Table 5.6. Study 1 (100% Subset Size + 36 Items): Bias and RMSE by Parameter

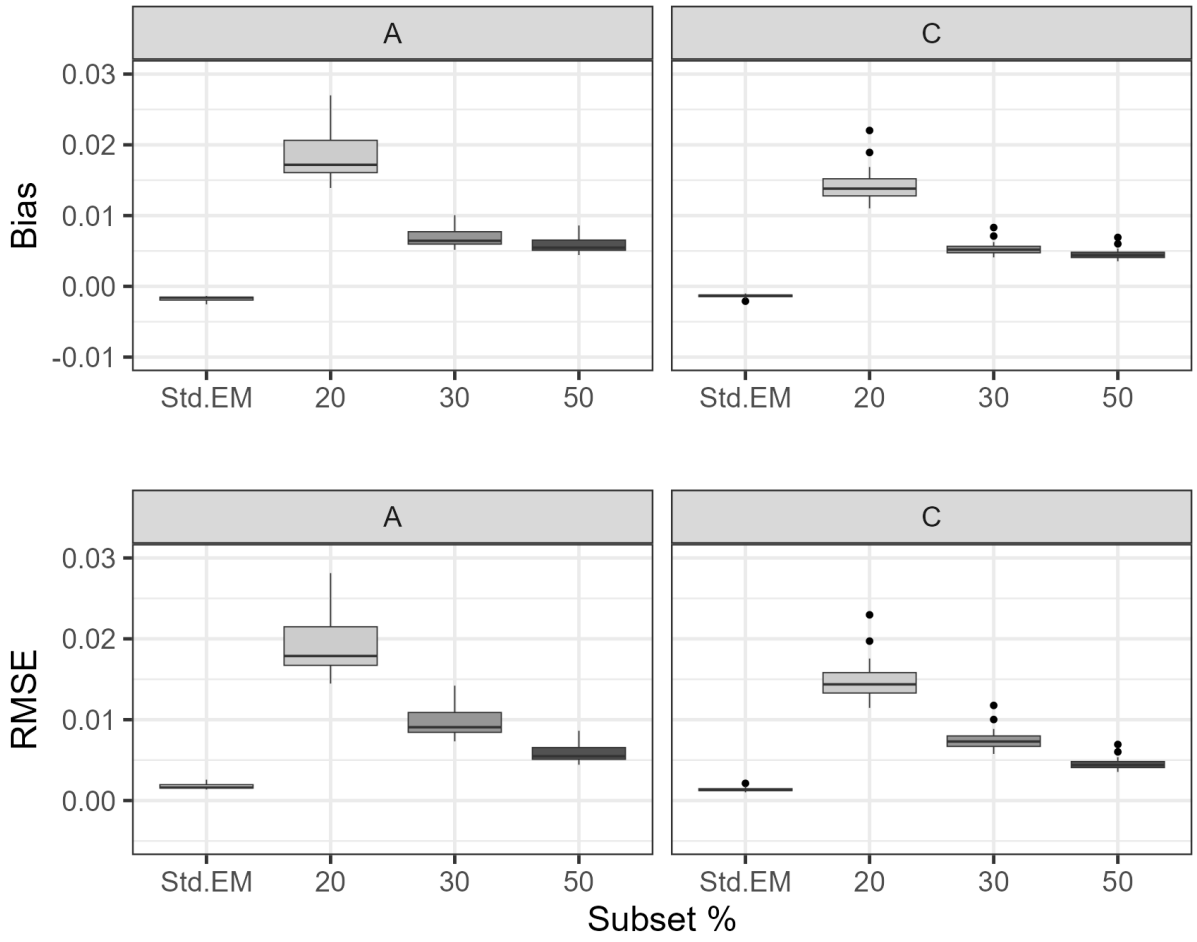
	Parameter A					Parameter C				
Item	True value	M_{est}	SD_{est}	Bias	RMSE	True value	M_{est}	SD_{est}	Bias	RMSE
1	0.64	0.64	0.01	0.00	0.01	1.06	1.06	0.01	0.00	0.01
2	1.43	1.44	0.02	0.00	0.02	3.34	3.34	0.03	0.00	0.03
3	0.89	0.89	0.02	0.00	0.02	2.32	2.32	0.02	0.00	0.02
4	1.16	1.16	0.02	0.00	0.02	2.62	2.62	0.02	0.00	0.02
5	1.46	1.47	0.02	0.00	0.02	−1.32	−1.32	0.01	0.00	0.01
6	0.96	0.96	0.01	0.00	0.01	−0.86	−0.86	0.01	0.00	0.01
7	0.54	0.54	0.01	0.00	0.01	−0.44	−0.44	0.01	0.00	0.01
8	0.50	0.50	0.01	0.00	0.01	0.31	0.31	0.01	0.00	0.01
9	1.11	1.11	0.02	0.00	0.02	−3.19	−3.18	0.03	0.00	0.03
10	0.80	0.80	0.01	0.00	0.01	−1.50	−1.50	0.01	0.00	0.01
11	1.17	1.16	0.02	0.00	0.02	−3.48	−3.48	0.03	0.00	0.03
12	0.97	0.97	0.02	0.00	0.02	−1.63	−1.63	0.01	0.00	0.01
13	2.42	2.41	0.03	0.00	0.03	3.82	3.82	0.04	0.00	0.04
14	1.78	1.78	0.03	0.00	0.03	3.95	3.95	0.04	0.01	0.04
15	1.65	1.66	0.04	0.00	0.04	4.93	4.93	0.05	0.00	0.05
16	1.82	1.83	0.02	0.00	0.02	2.76	2.76	0.02	0.00	0.02
17	2.01	2.01	0.02	0.00	0.02	−2.27	−2.27	0.02	0.00	0.02
18	1.57	1.57	0.02	0.00	0.02	1.01	1.02	0.01	0.00	0.01
19	1.97	1.97	0.02	0.00	0.02	−1.55	−1.54	0.02	0.00	0.02
20	2.42	2.42	0.02	0.00	0.02	0.78	0.78	0.02	0.00	0.02
21	2.43	2.42	0.05	−0.01	0.05	−5.46	−5.45	0.07	0.01	0.07
22	2.30	2.30	0.07	−0.01	0.07	−6.55	−6.54	0.13	0.00	0.13
23	1.98	1.98	0.04	0.00	0.04	−5.49	−5.49	0.07	0.00	0.07
24	1.78	1.78	0.03	0.00	0.03	−3.57	−3.57	0.03	0.00	0.03
25	2.61	2.61	0.05	0.00	0.05	5.77	5.77	0.09	0.00	0.08
26	3.15	3.16	0.04	0.01	0.04	5.20	5.21	0.05	0.01	0.05
27	2.60	2.60	0.06	0.00	0.06	6.59	6.59	0.10	0.00	0.10
28	2.58	2.58	0.03	0.00	0.03	4.11	4.11	0.05	0.00	0.04
29	3.46	3.47	0.05	0.01	0.05	3.20	3.21	0.04	0.01	0.04
30	3.45	3.46	0.04	0.01	0.04	0.58	0.58	0.02	0.00	0.02
31	2.76	2.76	0.03	0.00	0.03	−1.41	−1.41	0.02	0.00	0.02
32	2.88	2.88	0.04	0.00	0.04	−4.19	−4.19	0.04	0.00	0.04
33	3.17	3.16	0.12	−0.01	0.12	−9.46	−9.45	0.25	0.01	0.25
34	3.47	3.47	0.06	0.00	0.06	−6.16	−6.16	0.09	0.00	0.09
35	2.94	2.93	0.07	−0.01	0.07	−7.38	−7.36	0.13	0.02	0.13
36	3.15	3.16	0.07	0.00	0.07	−6.54	−6.55	0.11	−0.01	0.10

5.2 Study 2 Results

5.2.1 Bias and RMSE of Estimated SEs

SEs were estimated using XPD and compared to gold-standard SEs derived from FIS. Figure 5.2 shows the distribution of bias and RMSE values by condition. Under standard EM, overall bias and RMSE were less than 0.01. Under the proposed algorithm (i.e., the 20% to 50% conditions), upward bias and increased RMSE were observed—most noticeably for the 20% subset size and the C parameter—but the absolute magnitudes of both metrics remained below 0.03 in all conditions.

Figure 5.2. Study 2: Distribution of Bias and RMSE of XPD-Based SE Estimates



Tables 5.7 to 5.10 report the gold-standard SE, mean estimated SE, bias, and RMSE by

parameter under subset sizes of 20%, 30%, 50%, and 100%. All values were computed using unrounded data but are reported to three decimal places for readability.

Table 5.7. Study 2 (20% Subset Size): Bias and RMSE by Parameter

	Parameter A				Parameter C			
Item	SE_{fis}	Mean SE_{xpd}	Bias	RMSE	SE_{fis}	Mean SE_{xpd}	Bias	RMSE
1	0.015	0.029	0.014	0.015	0.014	0.026	0.012	0.013
2	0.022	0.042	0.020	0.021	0.021	0.040	0.019	0.020
3	0.026	0.049	0.024	0.025	0.024	0.046	0.022	0.023
4	0.019	0.036	0.017	0.018	0.016	0.030	0.015	0.015
5	0.020	0.038	0.018	0.019	0.015	0.028	0.013	0.014
6	0.025	0.047	0.023	0.023	0.014	0.027	0.013	0.013
7	0.018	0.035	0.017	0.017	0.012	0.023	0.011	0.012
8	0.029	0.056	0.027	0.028	0.018	0.035	0.017	0.018
9	0.018	0.034	0.016	0.017	0.014	0.027	0.013	0.013
10	0.019	0.036	0.017	0.018	0.016	0.031	0.015	0.015
11	0.017	0.033	0.016	0.016	0.015	0.029	0.014	0.014
12	0.017	0.032	0.015	0.016	0.015	0.029	0.014	0.015

Table 5.8. Study 2 (30% Subset Size): Bias and RMSE by Parameter

	Parameter A				Parameter C			
Item	SE_{fis}	Mean SE_{xpd}	Bias	RMSE	SE_{fis}	Mean SE_{xpd}	Bias	RMSE
1	0.015	0.021	0.005	0.007	0.014	0.018	0.005	0.007
2	0.022	0.029	0.008	0.011	0.021	0.028	0.007	0.010
3	0.026	0.035	0.009	0.013	0.024	0.032	0.008	0.012
4	0.019	0.026	0.007	0.009	0.016	0.021	0.005	0.008
5	0.020	0.026	0.007	0.010	0.015	0.020	0.005	0.007
6	0.025	0.033	0.008	0.012	0.014	0.019	0.005	0.007
7	0.018	0.024	0.006	0.009	0.012	0.016	0.004	0.006
8	0.029	0.040	0.010	0.014	0.018	0.025	0.006	0.009
9	0.018	0.024	0.006	0.009	0.014	0.019	0.005	0.007
10	0.019	0.025	0.006	0.009	0.016	0.022	0.006	0.008
11	0.017	0.023	0.006	0.008	0.015	0.020	0.005	0.007
12	0.017	0.023	0.006	0.008	0.015	0.020	0.005	0.007

5.3 Discussion

5.3.1 Summary of Findings

Study 1 evaluated the modified EM algorithm against standard EM under 18- and 36-item conditions across five different subset sizes. The results pointed to three main

Table 5.9. Study 2 (50% Subset Size): Bias and RMSE by Parameter

	Parameter A				Parameter C			
Item	SE_{fis}	Mean SE_{xpd}	Bias	RMSE	SE_{fis}	Mean SE_{xpd}	Bias	RMSE
1	0.015	0.020	0.004	0.004	0.014	0.017	0.004	0.004
2	0.022	0.028	0.006	0.006	0.021	0.027	0.006	0.006
3	0.026	0.033	0.007	0.007	0.024	0.031	0.007	0.007
4	0.019	0.025	0.006	0.006	0.016	0.020	0.005	0.005
5	0.020	0.026	0.006	0.006	0.015	0.019	0.004	0.004
6	0.025	0.032	0.007	0.007	0.014	0.018	0.004	0.004
7	0.018	0.024	0.005	0.005	0.012	0.016	0.004	0.004
8	0.029	0.038	0.009	0.009	0.018	0.024	0.005	0.005
9	0.018	0.023	0.005	0.005	0.014	0.018	0.004	0.004
10	0.019	0.024	0.005	0.005	0.016	0.021	0.005	0.005
11	0.017	0.022	0.005	0.005	0.015	0.019	0.004	0.004
12	0.017	0.022	0.005	0.005	0.015	0.020	0.004	0.004

Table 5.10. Study 2 (100% Subset Size): Bias and RMSE by Parameter

	Parameter A				Parameter C			
Item	SE_{fis}	Mean SE_{xpd}	Bias	RMSE	SE_{fis}	Mean SE_{xpd}	Bias	RMSE
1	0.015	0.014	−0.001	0.001	0.014	0.012	−0.001	0.001
2	0.022	0.020	−0.002	0.002	0.021	0.019	−0.002	0.002
3	0.026	0.023	−0.002	0.002	0.024	0.022	−0.002	0.002
4	0.019	0.017	−0.002	0.002	0.016	0.014	−0.001	0.001
5	0.020	0.018	−0.002	0.002	0.015	0.013	−0.001	0.001
6	0.025	0.022	−0.002	0.002	0.014	0.013	−0.001	0.001
7	0.018	0.017	−0.002	0.002	0.012	0.011	−0.001	0.001
8	0.029	0.027	−0.003	0.003	0.018	0.017	−0.002	0.002
9	0.018	0.016	−0.002	0.002	0.014	0.013	−0.001	0.001
10	0.019	0.017	−0.002	0.002	0.016	0.015	−0.001	0.001
11	0.017	0.016	−0.002	0.002	0.015	0.014	−0.001	0.001
12	0.017	0.015	−0.001	0.001	0.015	0.014	−0.001	0.001

takeaways: (a) parameter recovery performance differed between slope and intercept parameters, as well as between moderate and extreme parameter values; (b) recovery improved with larger subset sizes; and (c) computational efficiency—measured by reduction in time to convergence—scaled with item set length. Overall, the proposed algorithm offered greater computational advantages and maintained comparable estimation performance in larger item sets.

Parameter recovery improved with subset sizes of 20% and above. True slope parameters (A) were generally easier to recover than intercept parameters (C), and extreme

parameter values led to increased bias and RMSE, particularly for C parameters.

In terms of computational efficiency, the proposed algorithm showed clear gains with 36 items but not with 18. In the 18-item condition, the algorithm reduced convergence time only at the smallest subset size (10%), while larger subsets led to slower performance. In contrast, the 36-item condition showed consistent time savings across all subset sizes. This difference may be explained primarily by the structure of the response pattern matrix. With more items, individuals are more likely to produce unique response patterns, increasing the size and complexity of the response pattern matrix. In such cases, enumerating all response patterns becomes computationally comparable to working with the full item response data, making a subset-based approach more efficient. By contrast, with fewer items (e.g., 18), the response pattern matrix remains compact, and the overhead of managing subsets can outweigh potential time savings.

Study 2 examined the bias and RMSE of SE estimates produced via XPD under the modified algorithm, using FIS-based SEs as the reference. On the one hand, the modified algorithm introduced small upward bias and RMSE compared to SEs obtained under standard EM (100% subset size). On the other hand, the magnitudes of these differences remained below 0.03 in all conditions. Notably, the 30% subset condition kept bias below 0.010 and RMSE below 0.015, and the 50% subset condition kept both below 0.010.

5.3.2 Implications for Future Work

This work highlights several directions for future research and application. Standard EM-based IRT estimation remains widely used in large-scale assessments and certification exams. One example is the SAT, which collects item-level response data from millions of examinees annually (College Board, 2025). In such contexts, the computational burden of full-data EM estimation for item calibration can be substantial. The proposed algorithm provides an alternative that reduces computation time without compromising estimation performance.

Another strength of this work is that the proposed modifications can be integrated

into existing IRT pipelines with minimal changes to core procedures. Future work could explore packaging them as modular add-ons to existing software to support adoption in operational contexts. Empirical validation using real-world data could also help identify practical limits and evaluate trade-offs between computational efficiency, accuracy, and precision.

Future research could examine whether the proposed algorithm and its variants extend to multidimensional IRT settings. This includes investigating whether selectively subsetting by item clusters or examinee groups yields similar or greater computational benefits, given that the number of quadrature nodes evaluated during the E-step increases exponentially with the number of latent variables. Another research direction is to explore parallel computing and ensemble-styled extensions, such as running multiple instances of the modified EM algorithm—or a combination of standard and modified EM—in parallel to mitigate the impact of idiosyncratic error from a single run, assess stability, and reduce overall runtime.

A R Function for Computing SEs Based on the Fisher Expected Information Matrix

```
#' Compute SEs based on the Fisher expected information matrix (FIS)
#'
#' Read a comma-delimited flexMIRT generated -cov.txt file based
#' on a unidimensional 2PL IRT model and compute SEs based on
#' the Fisher expected information matrix
#'
#' The -cov.txt file is saved with the following settings in flexMIRT 3.72
#'   1. N = 1
#'   2. MaxE = 0
#'   3. SaveCOV = Yes
#'   4. Use the Value operator to constrain parameters to their true values
#'
#' @param cov_path:
#'      Path to the -cov.txt file.
#' @param sample_size:
#'      The total sample size.
#' @param var_digits:
#'      No. sig. digits to round the variance values. Default is 2.
#' @param se_digits:
#'      No. sig. digits to round the SE values. Default is 1.
#'
#' @return A data frame with SEs computed based on FIS.
#'
#' The example below replicates FIS-based SEs in Table 3 of Cai (2008)
#' which used LSAT6 data: https://doi.org/10.1348/000711007X249603
#'
#' @examples
#' \dontrun{
#' result ← cov2Se("CAI2008-cov.txt", sample_size = 1000)
#' print(result)
#' }
```

```r

cov2Se <- function(cov_path, sample_size, var_digits = 2, se_digits = 1) {
  # Load the -cov.txt file saved from flexMIRT
  FINV <- as.matrix(read.csv(cov_path, header = FALSE))

  # Drop the last column if it's fully NA
  if (all(is.na(FINV[, ncol(FINV)]))) {
    FINV <- FINV[, -ncol(FINV)]
  }

  # Scale to get covariance matrix
  COV <- FINV / sample_size

  # Compute standard errors
  SEs <- sqrt(diag(COV))

  # Select indices: even positions for a, odd for c
  selected_indices <- c(seq(2, length(SEs), 2), seq(1, length(SEs), 2))

  # Construct parameter names
  num_items   <- length(SEs) / 2
  param_names <- sort(
      c(outer(c("a", "c"), sprintf("%02d", 1:num_items), paste0))
  )

  # Format variance without scientific notation
  var_formatted <- format(diag(COV)[selected_indices],
                          digits = var_digits,
                          nsmall = 2,
                          scientific = FALSE)

  # Format SE without scientific notation
  se_formatted  <- format(SEs[selected_indices],
                          digits = se_digits,
```

```
                              nsmall = 2,
                              scientific = FALSE)

  # Create results data frame
  results ← data.frame(
    Parameter        = param_names,
    Var_Scaled       = var_formatted,
    SE               = se_formatted,
    stringsAsFactors = FALSE
  )

  return(results)
}
```

BIBLIOGRAPHY

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

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (2nd ed., pp. 437–478). Springer.

Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika*, *46*(4), 443–459. https://doi.org/10.1007/BF02293801

Cai, L. (2024). *flexMIRT®: Flexible multilevel multidimensional item analysis and test scoring* (Version 3.72) [Computer software]. Vector Psychometric Group.

Cai, L. (2008). SEM of another flavour: Two new applications of the supplemented EM algorithm. *British Journal of Mathematical and Statistical Psychology*, *61*(2), 309–329. https://doi.org/10.1348/000711007X249603

Cappé, O., & Moulines, E. (2009). Online EM algorithm for latent data models. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, *71*(3), 593–613. https://doi.org/10.1111/j.1467-9868.2009.00698.x

Chen, J., Zhu, J., Teh, Y. W., & Zhang, T. (2018). Stochastic expectation maximization with variance reduction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 7967–7977, Vol. 31). Curran Associates.

College Board. (2025). *SAT participation continues to grow as the SAT suite successfully completes its transition to digital testing*. https://newsroom.collegeboard.org/sat-participation-continues-grow-sat-suite-successfully-completes-its-transition-digital-testing

de Ayala, R. J. (2009). *The theory and practice of item response theory*. Guilford Press.

Delyon, B., Lavielle, M., & Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, *27*(1), 94–128.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, *39*(1), 1–38.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gower, R. M., Schmidt, M., Bach, F., & Richtarik, P. (2020, October 2). *Variance-reduced methods for machine learning*. arXiv: 2010.00892 [cs]. https://doi.org/10.48550/arXiv.2010.00892

Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Weinberger (Eds.), *Advances in neural information processing systems* (pp. 315–323, Vol. 26). Curran Associates.

Liang, P., & Klein, D. (2009). Online EM for unsupervised models. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 611–619. https://doi.org/10.3115/1620754.1620843

Meng, X.-L., & Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, *80*(2), 267–278.

Neal, R. M., & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Springer. https://doi.org/10.1007/978-94-011-5014-9_12

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*(4), 838–855.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*(3), 400–407.

Thiesson, B., Meek, C., & Heckerman, D. (2001). Accelerating EM for large databases. *Machine Learning*, *45*(3), 279–299. https://doi.org/10.1023/A:1017986506241

Tian, W., Cai, L., Thissen, D., & Xin, T. (2013). Numerical differentiation methods for computing error covariance matrices in item response theory modeling: An eval-

uation and a new proposal. *Educational and Psychological Measurement*, *73*(3), 412–439. https://doi.org/10.1177/0013164412465875

von Davier, M. (2016). *High-performance psychometrics: The parallel-E parallel-M algorithm for generalized latent variable models* (ETS Research Report Series No. 2). Educational Testing Service. https://doi.org/https://doi.org/10.1002/ets2.12120

Weissman, A. (2013). Optimizing information using the EM algorithm in item response theory. *Annals of Operations Research*, *206*(1), 627–646. https://doi.org/10.1007/s10479-012-1204-4

Yin, J., Zhang, Y., & Gao, L. (2018). Accelerating distributed expectation–maximization algorithms with frequent updates. *Journal of Parallel and Distributed Computing*, *111*, 65–75.