A Two-Level Adaptive Test Battery

Wim J. van der Linden

Qi Diao

Introduction

Most educational and psychological tests are organized as a coherent battery of subtests. Score profiles on these batteries are much more informative than single scores on one overall test.

There also are valuable psychometric benefits related to the use of test batteries:

- Much better fit of a unidimensional IRT model to each of the subtests than for one overall test;
- As the subtests typically correlate strongly, it is possible to exchange collateral information between subtests.

Introduction

For a fixed timeslot, the design of a test battery must solve the dilemma between the *richness* and *accuracy* of its score profiles: the larger the number of subtests, the less accurate their scores.

Making the subtests adaptive relaxes the dilemma considerably. Following a well-known rule of thumb, we can reduce the testing time by 50% or double the number of subtests without sacrificing any accuracy.

No wonder the early pioneers of adaptive testing immediately explored these new opportunities (Brown & Weiss, 1977; Gialucca & Weiss, 1979)!

Introduction

However, it is possible to go one step further and introduce another level of adaptation: rather than administering the battery in an (arbitrarily) fixed order, we could also select each next subtest adaptively.

As the subtests generally correlate, valuable gains in accuracy and/or reduction of test length are possible.

Introduction

The extra step thus leads to two levels of adaptation:
- within-subtest adaptation (item selection)
- between-subtest adaptation (subpool selection)

The specific application we have in mind is a large battery for cognitive diagnosis with subtests of, say, 5-7 items each.

Introduction

In the remainder of this presentation, we (1) introduce the two-level model necessary to run the battery, (2) discuss the rules for adaptive item and subtest selection, and (3) finish with a presentation of empirical results.

Two-Level Model

First level:

$$\Pr\{U_i = 1\} = c_i + (1 - c_i)[1 + \exp(-a_i(\theta_i - b_i))]^{-1}$$

Second level:

$$f(\theta_1, \ldots, \theta_H) = MVN(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$$

Two-Level Model

The choice of the 3PL model is for presentation purposes only. Any other model with a single ability parameter can be used similarly.

For identification purposes, we restrict covariance matrix $\boldsymbol{\Sigma}_\theta$ to a matrix with $\mu_\theta = 0$ and $\sigma_\theta = 1$ for all ability parameters; that is, use their correlation matrix.

If necessary, the ability parameters can be transformed back and forth to improve the fit of the multivariate normal.

Item and Subpool Selection Rules

Both levels of adaptation are implemented using the shadow-test approach. The approach allows us

1. to impose whatever necessary content and statistical constraints on each of the subtests;
2. manage all other possible adaptive processes simultaneously; and, importantly,
3. avoid the error of choosing items or subpools that appear to be suboptimal at a later stage.

Item and Subpool Selection Rules

We also use a sequential Bayesian approach to adaptive testing (van der Linden & Ren, JEBS, 2020, 58-85):

1. Rather than point estimates, small samples from the last posterior distributions of the ability parameters, item parameters, and correlations are saved in system;
2. Update of the ability parameters using a Gibbs sampler, resampling all item parameters and correlations but with a Metropolis-Hastings (MH) step for the ability parameters;
3. Maximum information (MI) criterion averaged across all posterior samples to fully allow for parameter uncertainty.

Selection of First Subpool

1. Assemble the first shadow test from each of the subpools, using a model with all desired constraints and the MI criterion averaged across all posterior samples as objective function
2. Select the subpool with the best value for the objective function for its shadow test and begin the subtest with its best item.
3. Continue the subtest updating the posterior distribution of the ability parameter and re-assembling the shadow test after each next item.

Selection of Next Subpools

1. Assemble the first shadow test from each of the remaining subpools, but now averaging the MI criterion over the *predictive posterior distribution* of their ability parameters given the final update of the ability parameter for the previous subpools.
2. Choose the subpool with the best shadow test and begin its subtest.

The *second-level model* is used at Step 1. Specifically, we use its conditional distribution of the ability parameter for each candidate subpool given all parameters for the earlier pools.

Calculation of Final Score Profiles

So far, the procedure has been asymmetric; later subtests profit more from their predecessors than earlier subtests.

After the last subtest, we therefore score each earlier subtest recalculating the posterior distribution of its ability parameter given the responses on *all* other subtests.

A simple way of doing so is to rerun the program with each of the subtests in the last position, feeding the test taker's earlier responses into the system again.

“One Long Adaptive Test”

A simple way of interpreting the entire procedure is as one long continuous adaptive test across multiple unidimensional content domains.

The only difference with a single-level adaptive test is the replacement of the within-test *MI criterion* with its *posterior prediction* when moving between subtests.

Posterior prediction does not require unidimensionality. We can predict any ability parameter from any other, provided we know their joint distribution.

Empirical Example

The procedure was applied to a real-world battery with four subtests.

Each of the subpools had 150 items randomly sampled from an inventory of retired operational items.

For a typical population of test takers, the means and correlations of the ability parameters were

$$\boldsymbol{\mu}_\theta = (-0.92, -1.00, -0.62, -0.96) \qquad \boldsymbol{\Sigma}_\theta = \begin{bmatrix} 1.02 & & & \\ 0.63 & 0.80 & & \\ 0.85 & 0.64 & 1.21 & \\ 0.78 & 0.71 & 0.89 & 1.18 \end{bmatrix}$$

Empirical Example

The second-level model with these empirical values was used to generate the response data for recalibration of the item parameters and correlations (1,000 test takers per item).

The recalibration was executed using the MH within Gibbs sampler in *JAGS*.

We then checked the autocorrelation in the Markov chains to determine the size of the vectors with posterior draws used by the system during adaptive testing (500 independent draws for each parameter and correlation).

Empirical Example

Finally, the adaptive battery was simulated in *R* using the MIP solver in the *lpSolveAPI* package.

The simulated conditions were:

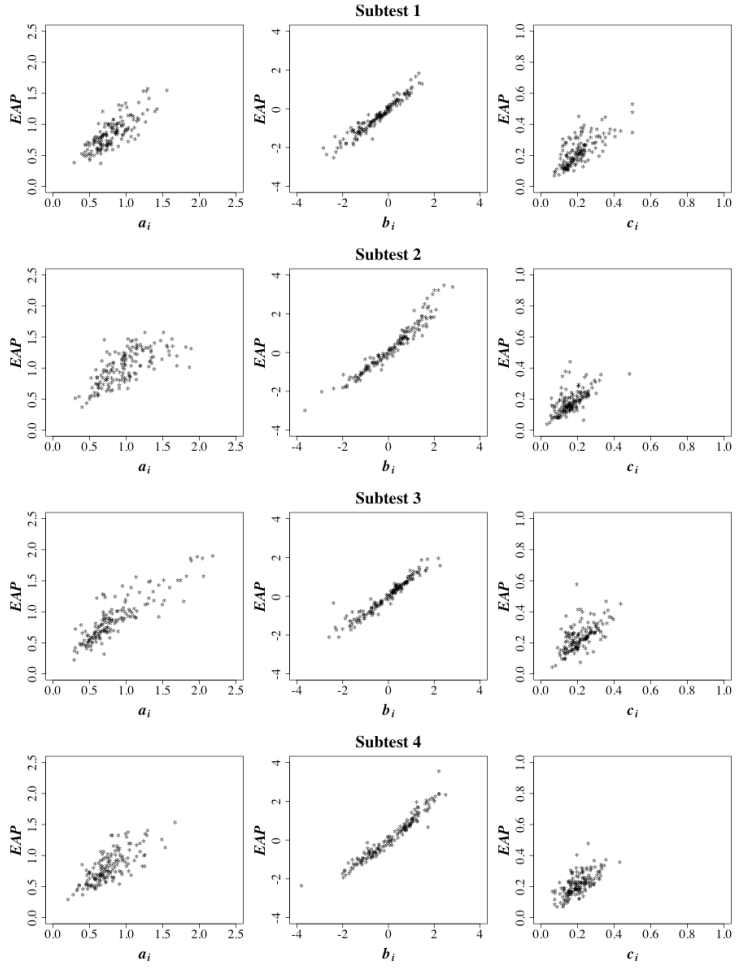
- two-level v. one-level adaptive testing;
- subtest length of 5 and 10 items.

Empirical Example

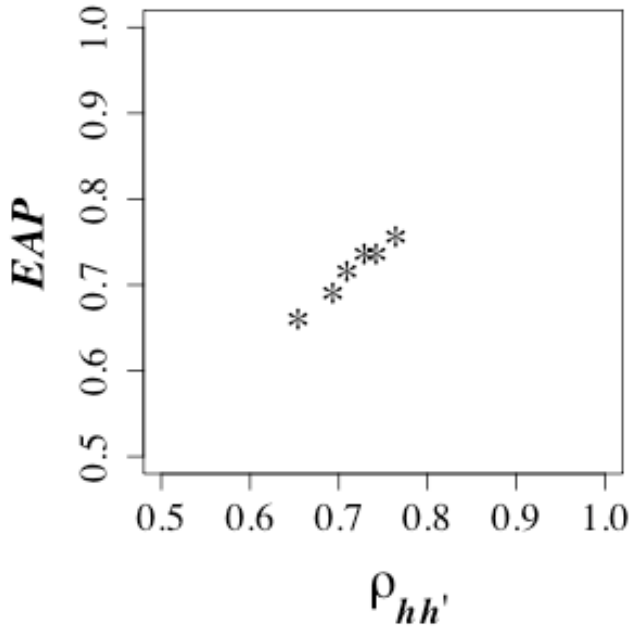
Disclaimers:

- For security reasons, we had no access to the content constraints in use for the battery and used the shadow tests only to set the length of the subtests.
- We plan to rerun the simulation to implement Bayesian model identification (through prior distributions) rather than the choice of means and variances equal to zero and one for the second-level model.

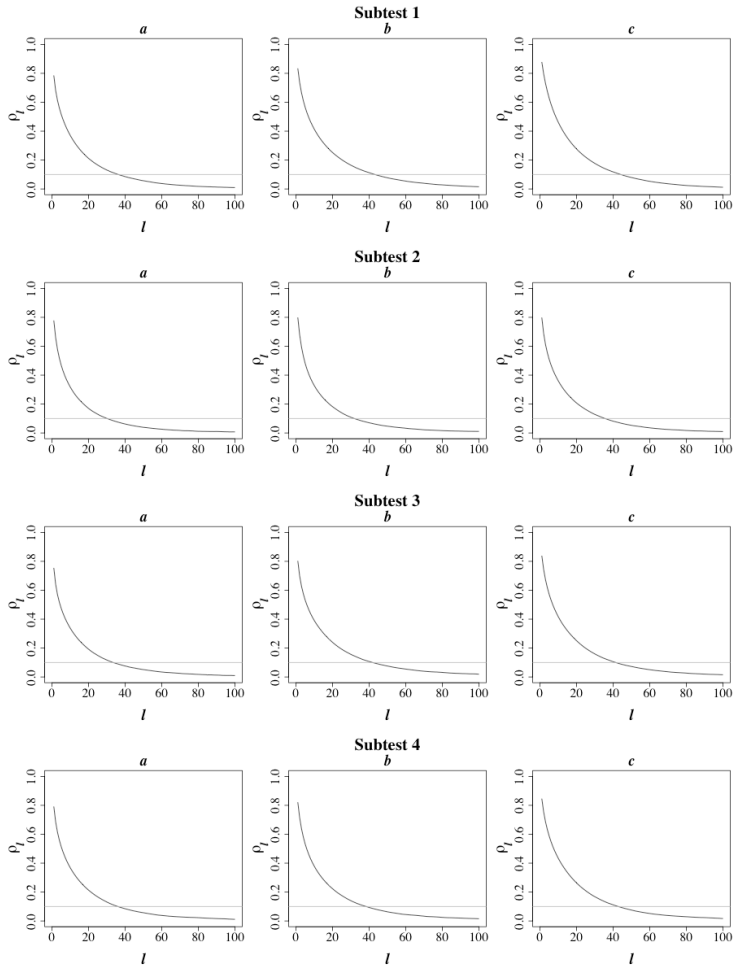
Posterior means of item parameters against true values



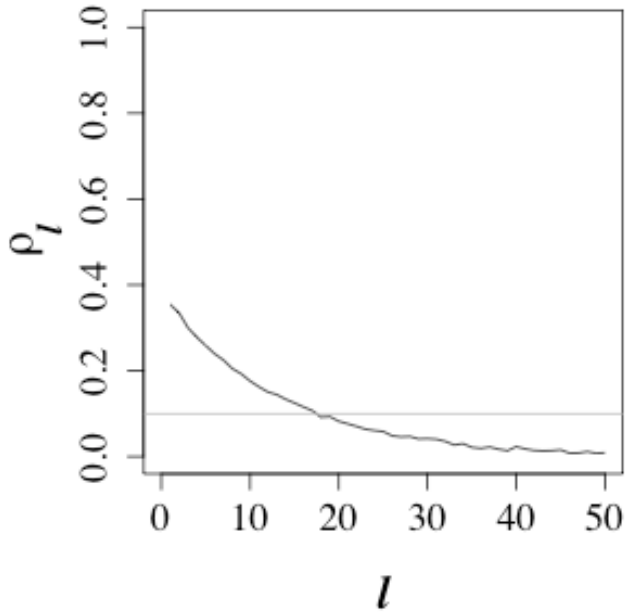
Posterior means of correlations against true values



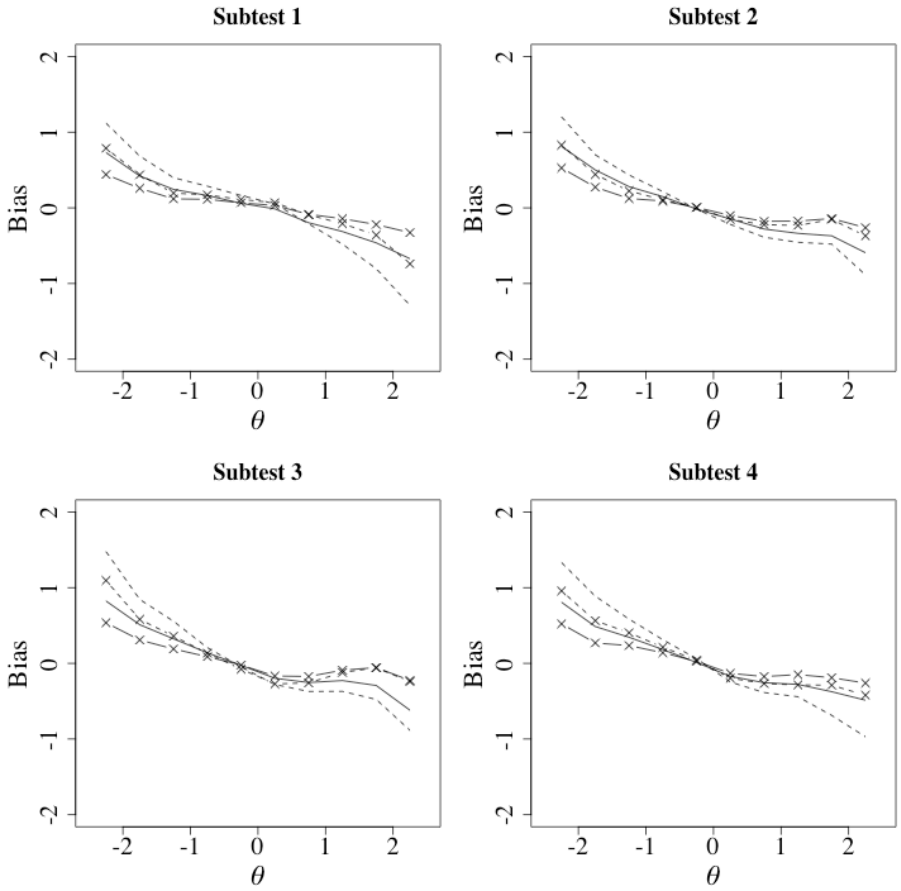
Average autocorrelation as function of lag size (item parameters)



Average autocorrelation as function of lag size (correlations)



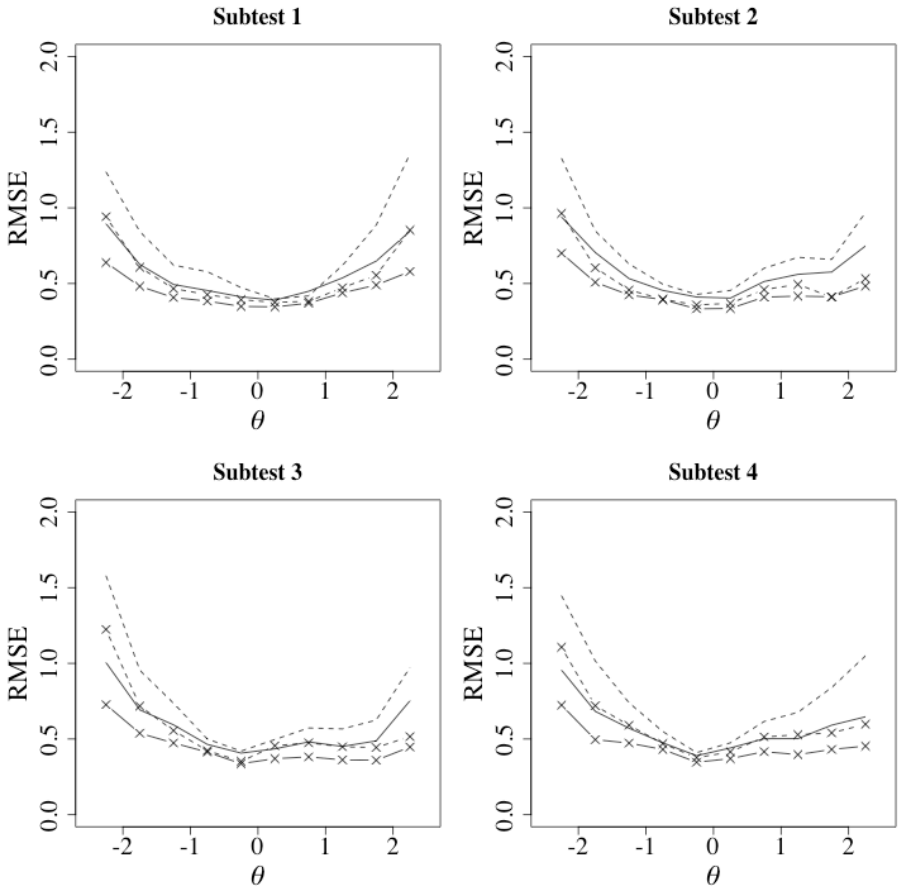
Bias functions of final subtest scores



dashed curve: one level
solid curve: two levels

no cross: five items
cross: ten items

RMSE functions of final subtest scores



dashed curve: one level
solid curve: two levels

no cross: five items
cross: ten items

Frequency of paths through battery followed by test takers

Possible Path	5-Item Subtest	10-Item Subtest
1234	461	50
1243	16	4
1324	318	24
2134	2474	2632
2143	164	304
2314	1028	1343
2341	501	612
3124	26	12
3214	9	13
3241	3	6

Running Times

Even though the battery was run in *R* on a standard PC, the average running times for a simulated test taker to finish the battery was 0.49 s (5-item subtests) and 0.66s (10-item subtests).

The time to calculate the final scores for the two subtest lengths was 5.33 s and 5.48 s.

Conclusion

The change from a traditional one-level to a two-level adaptive test battery does pay off.

Just by making the choice of each next subtest adaptive rather than arbitrarily fixing their order, the same item pool gives score profiles with much better RMSEs and less bias.

The example was only for a battery of four subtests. The score profiles will improve further with the addition of each extra subtest to the battery.