Measuring the Power of Learning.™

Research Report
ETS RR–18-15

SARM: A Computer Program for Estimating Speed-Accuracy Response Models for Dichotomous Items

Peter W. van Rijn

Usama S. Ali

December 2018

Discover this journal online at
Wiley Online Library
wileyonlinelibrary.com

ETS Research Report Series

EIGNOR EXECUTIVE EDITOR

James Carlson
Principal Psychometrician

ASSOCIATE EDITORS

Beata Beigman Klebanov
Senior Research Scientist

Heather Buzick
Senior Research Scientist

Brent Bridgeman
Distinguished Presidential Appointee

Keelan Evanini
Research Director

Marna Golub-Smith
Principal Psychometrician

Shelby Haberman
Distinguished Research Scientist, Edusoft

Anastassia Loukina
Research Scientist

John Mazzeo
Distinguished Presidential Appointee

Donald Powers
Principal Research Scientist

Gautam Puhan
Principal Psychometrician

John Sabatini
Managing Principal Research Scientist

Elizabeth Stone
Research Scientist

Rebecca Zwick
Distinguished Presidential Appointee

PRODUCTION EDITORS

Kim Fryer
Manager, Editing Services

Ayleen Gontz
Senior Editor

Since its 1947 founding, ETS has conducted and disseminated scientific research to support its products and services, and to advance the measurement and education fields. In keeping with these goals, ETS is committed to making its research freely available to the professional community and to the general public. Published accounts of ETS research, including papers in the ETS Research Report series, undergo a formal peer-review process by ETS staff to ensure that they meet established scientific and professional standards. All such ETS-conducted peer reviews are in addition to any reviews that outside organizations may provide as part of their own publication processes. Peer review notwithstanding, the positions expressed in the ETS Research Report series and other published accounts of ETS research are those of the authors and not necessarily those of the Officers and Trustees of Educational Testing Service.

The Daniel Eignor Editorship is named in honor of Dr. Daniel R. Eignor, who from 2001 until 2011 served the Research and Development division as Editor for the ETS Research Report series. The Eignor Editorship has been created to recognize the pivotal leadership role that Dr. Eignor played in the research publication process at ETS.

ETS Research Report Series ISSN 2330-8516

RESEARCH REPORT

SARM: A Computer Program for Estimating Speed-Accuracy Response Models for Dichotomous Items

Peter W. van Rijn[1] & Usama S. Ali[2]

1 ETS Global, Amsterdam, The Netherlands
2 Educational Testing Service, Princeton, NJ

A computer program was developed to estimate speed-accuracy response models for dichotomous items. This report describes how the models are estimated and how to specify data and input files. An example using data from a listening section of an international language test is described to illustrate the modeling approach and features of the computer program.

Keywords Item response theory; response times; estimation software

doi:10.1002/ets2.12200

A computer program[1] was developed to estimate speed-accuracy response models for dichotomous items. This type of model, introduced by Maris and van der Maas (2012), is characterized by a scoring rule that makes use of both speed and accuracy of item responses and specific time limits for each item. The models essentially describe a smooth curve for the expected score as a function of a latent variable. This report describes how the models are estimated and how to specify data and input files. For more details about the model and estimation procedures, the reader is referred to van Rijn and Ali (2018). Comparisons with other models and applications of the model in the context of adaptive testing are provided by van Rijn and Ali (2017).

Model

We first introduce some notation. Test takers are indicated by $i = 1, 2, \ldots, n$, with $n > 1$ being the total number of test takers, and items are indicated by $j = 1, 2, \ldots, m$, with $m > 1$ being the total number of items. The following general scoring rule is used to obtain score variable S_{ij} for test taker i on item j (Maris & van der Maas, 2012, Equation 55):

$$S_{ij} = S\left(X_{ij}, T_{ij}\right) = \left[w_{j0}\left(1 - X_{ij}\right) + w_{j1}X_{ij}\right]\left(d_j - T_{ij}\right), \tag{1}$$

where w_{j0} is a known negative weight (e.g., -1) for the incorrect response and w_{j1} is a known positive weight (e.g., 1) for the correct response; X_{ij} is the variable for dichotomously scored item response of person i to item j, with 0 for an incorrect response and 1 for a correct response; d_j is the time limit for item j; and T_{ij} is the response time variable. It is assumed that the response time variable is nonnegative, continuous, and not larger than the time limit. Ideally, test takers should be informed about the scoring rule (i.e., the weights and the time limits should be known to test takers). Figure 1 shows two examples of a scoring rule in which the score is graphed as a function of response time for both incorrect and correct responses. The left panel shows the so-called signed residual time scoring rule, with $w_0 = -1$, $w_1 = 1$, and $d = 30$. The right panel shows an asymmetrical example with $w_0 = -1/4$, $w_1 = 1$, and $d = 30$. This is akin to formula scoring. For both examples, it holds that the score fades to 0 as the response time reaches the item time limit.

For test taker i and item j, we can use the scoring rule in the following way to obtain the density of the score S_{ij}:

$$f\left(s_{ij}|\theta_i\right) = \frac{\exp\left(s_{ij}\eta_{ij}\right)}{C\left(\eta_{ij}\right)}, \tag{2}$$

Corresponding author: P. W. van Rijn, E-mail: pvanrijn@ets.org

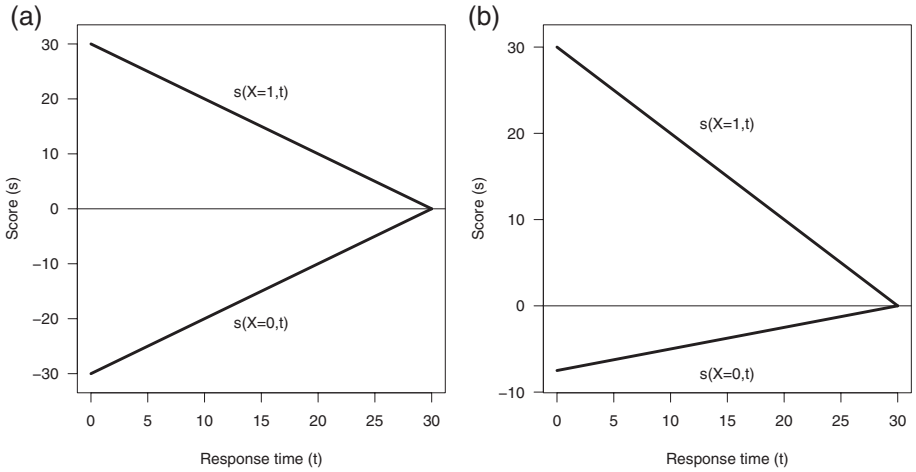


Figure 1 Examples of scoring rules: (left) $w_0 = -1$, $w_1 = 1$, $d = 30$; (right) $w_0 = -1/4$, $w_1 = 1$, $d = 30$.

where $\eta_{ij} = \alpha_j\theta_i + \beta_j$, θ_i is a person parameter, α_j is an item slope parameter, β_j is an item intercept parameter, and $C_j(\eta_{ij})$ is a normalizing factor. This model is referred to as the two-parameter speed-accuracy response model (2P-SARM). A one-parameter version of the model (1P-SARM) is obtained by dropping the item index of the slope parameter α_j. The normalizing factor is

$$\begin{aligned} C\left(\eta_{ij}\right) &= \sum_{k=0}^{1} \int_0^{d_j} \exp\left[w_{jk}\left(d_j - t\right)\eta_{ij}\right] dt \\ &= \frac{\exp\left(w_{j0} d_j \eta_{ij}\right) - 1}{w_{j0}\eta_{ij}} + \frac{\exp\left(w_{j1} d_j \eta_{ij}\right) - 1}{w_{j1}\eta_{ij}}. \end{aligned} \tag{3}$$

We refer to the expectation of the score as the item score function (ISF), which is given by

$$\begin{aligned} E\left(S_{ij}|\theta_i\right) &= \sum_{k=0}^{1} \frac{1}{C\left(\eta_{ij}\right)} \int_0^{d_j} w_{jk}\left(d_j - t\right) \exp\left[w_{jk}\left(d_j - t\right)\eta_{ij}\right] dt \\ &= \frac{1}{C\left(\eta_{ij}\right)} \sum_{k=0}^{1} \frac{\exp\left(w_{jk} d_j \eta_{ij}\right)\left(w_{jk} d_j \eta_{ij} - 1\right) + 1}{w_{jk}\eta_{ij}^2}. \end{aligned} \tag{4}$$

The variance of the score, $\text{Var}(S_{ij}|\theta_i)$, can be derived in a similar fashion. More details of the model can be found in van Rijn and Ali (2017), but it is important to note here that the marginal model for accuracy of the 2P-SARM equals the two-parameter logistic model (2PLM). That is, the 2P-SARM can be seen as an extension of the 2PLM, in which response time is entered by a particular scoring rule. Maris and van der Maas (2012; Figure 6) have shown that this leads to additional information for the estimation of θ throughout the ability range (if the model holds).

Estimation

The program performs maximum marginal likelihood estimation of the item parameters for the one- and two-parameter versions of the model. The item parameter vector is defined by $\boldsymbol{\gamma} = (\boldsymbol{\alpha}', \boldsymbol{\beta}')'$. The likelihood is then given by

$$L(\boldsymbol{\gamma}; \mathbf{S}, \boldsymbol{\theta}) = \prod_{i=1}^{n} f\left(\mathbf{s}_i|\theta_i\right) f\left(\theta_i\right), \tag{5}$$

where $\mathbf{s}_i = \{s_{ij}, 1 \le j \le m\}$, $\mathbf{S} = \{\mathbf{s}_i, 1 \le i \le n\}$, $\theta = \{\theta_i, 1 \le i \le n\}$, and $f(\theta_i)$ is a density. A standard normal density is assumed for all test takers, which identifies the model. The index i can then be dropped from θ in finding the marginal likelihood:

$$L(\boldsymbol{\gamma}; \mathbf{S}) = \prod_{i=1}^{n} \int_{-\infty}^{\infty} f\left(\mathbf{s}_i|\theta\right) f(\theta)\, d\theta. \tag{6}$$

Furthermore, we write the log-likelihood as

$$\ell(\boldsymbol{\gamma}) = \ln L(\boldsymbol{\gamma}; \mathbf{S}), \tag{7}$$

with the contribution of test taker i written as $\ell_i(\gamma)$. Under local independence, we can write

$$f\left(\mathbf{s}_i|\theta\right) = \prod_{j=1}^{m} f\left(s_{ij}|\theta\right). \tag{8}$$

If we let $f(\mathbf{s}_i) = \int f(\mathbf{s}_i|\,\theta) f(\theta) d\theta$, then the posterior of θ is

$$f\left(\theta|\mathbf{s}_i\right) = \frac{f\left(\mathbf{s}_i|\theta\right) f(\theta)}{f\left(\mathbf{s}_i\right)}. \tag{9}$$

The program can employ both the expectation–maximization (EM) and Newton–Raphson (NR) algorithms for estimating the item parameters (see the estimspec section for further details). Let $\hat{\boldsymbol{\gamma}}$ be the current parameter estimate (or starting value). To set up the EM algorithm, the Q-function is (Dempster, Laird, & Rubin, 1977)

$$\begin{aligned} Q\left(\boldsymbol{\gamma}|\hat{\boldsymbol{\gamma}}\right) &= \mathrm{E}_{\theta|\mathbf{S},\hat{\boldsymbol{\gamma}}}\left[\ell(\boldsymbol{\gamma})\right] \\ &= \sum_{i=1}^{n} \int \sum_{j=1}^{m} \left[s_{ij}\left(\alpha_j\theta + \beta_j\right) - \log C\left(\eta_{ij}\right) - \log\left(\sqrt{2\pi}\right) - \frac{1}{2}\theta^2\right] f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right) d\theta, \end{aligned} \tag{10}$$

where $f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right)$ is the posterior of θ evaluated at the current parameter estimate. Relevant derivatives for item parameters α_j and β_j for item j are

$$\frac{\partial Q\left(\boldsymbol{\gamma}|\hat{\boldsymbol{\gamma}}\right)}{\partial \alpha_j} = \sum_{i=1}^{n} \int \theta \left[s_{ij} - E\left(S_{ij}|\theta\right)\right] f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right) d\theta, \tag{11}$$

$$\frac{\partial Q\left(\boldsymbol{\gamma}|\hat{\boldsymbol{\gamma}}\right)}{\partial \beta_j} = \sum_{i=1}^{n} \int \left[s_{ij} - E\left(S_{ij}|\theta\right)\right] f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right) d\theta, \tag{12}$$

$$\frac{\partial Q\left(\boldsymbol{\gamma}|\hat{\boldsymbol{\gamma}}\right)}{\partial \alpha_j \partial \alpha_j} = -\sum_{i=1}^{n} \int \theta^2 \mathrm{Var}\left(S_{ij}|\theta\right) f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right) d\theta, \tag{13}$$

$$\frac{\partial Q\left(\boldsymbol{\gamma}|\hat{\boldsymbol{\gamma}}\right)}{\partial \alpha_j \partial \beta_j} = -\sum_{i=1}^{n} \int \theta \mathrm{Var}\left(S_{ij}|\theta\right) f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right) d\theta, \tag{14}$$

$$\frac{\partial Q\left(\boldsymbol{\gamma}|\hat{\boldsymbol{\gamma}}\right)}{\partial \beta_j \partial \beta_j} = -\sum_{i=1}^{n} \int \mathrm{Var}\left(S_{ij}|\theta\right) f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right) d\theta. \tag{15}$$

The expectation step consists of computing the individual posteriors $f\left(\theta|\mathbf{s}_i, \hat{\boldsymbol{\gamma}}\right)$, and the maximization step can then be performed with the NR algorithm using the preceding derivatives. In the program, the integrals are approximated by adaptive Gauss–Hermite (GH) quadrature (Fahrmeir & Tutz, 2001; Naylor & Smith, 1982; van Rijn & Ali, 2017).

The full NR algorithm can be employed as well by using the update at iteration $k \geq 0$:

$$\hat{\boldsymbol{\gamma}}_{k+1} = \hat{\boldsymbol{\gamma}}_k - \left[\nabla^2 \ell\left(\hat{\boldsymbol{\gamma}}_k\right)\right]^{-1} \nabla \ell\left(\hat{\boldsymbol{\gamma}}_k\right), \tag{16}$$

with $\nabla^2 \ell\left(\hat{\boldsymbol{\gamma}}_k\right)$ negative definite. To this end, we need the observed information matrix, which is (Yuan, Cheng, & Patton, 2014, Equation 11)

$$\mathbf{I}\left(\widehat{\boldsymbol{\gamma}}\right) = -\sum_{i=1}^{n}\int \frac{\ell_i(\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}\partial\boldsymbol{\gamma}'} f\left(\theta|\mathbf{s}_i,\widehat{\boldsymbol{\gamma}}\right)d\theta - \sum_{i=1}^{n}\int \frac{\ell_i(\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}\left[\frac{\ell_i(\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}\right]' f\left(\theta|\mathbf{s}_i,\widehat{\boldsymbol{\gamma}}\right)d\theta$$
$$+\sum_{i=1}^{n}\left\{\int \frac{\ell_i(\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}} f\left(\theta\mathbf{s}_i\widehat{\boldsymbol{\gamma}}\right)d\theta \int \left[\frac{\ell_i(\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}\right]' f(\theta|\mathbf{s}_i,\widehat{\boldsymbol{\gamma}})d\theta\right\}, \tag{17}$$

where the relevant first-order and second-order derivatives are easily spotted in Equations 11 and 12 and are evaluated at $\widehat{\boldsymbol{\gamma}}$. The first term in Equation 17 is the Hessian matrix from the M-step with elements defined by Equations 13 – 15. The third term can be referred to as the expected information matrix.

The program produces three sets of standard errors (*SE*s). Let $\mathbf{J}\left(\widehat{\boldsymbol{\gamma}}\right)$ denote the expected information matrix. Then, we can define the following three sets of *SE*s:

$$SE\left(\widehat{\boldsymbol{\gamma}}\right)^{(1)} = \sqrt{\text{diag}\left[\mathbf{I}\left(\widehat{\boldsymbol{\gamma}}\right)^{-1}\right]}, \tag{18}$$

$$SE\left(\widehat{\boldsymbol{\gamma}}\right)^{(2)} = \sqrt{\text{diag}\left[\mathbf{J}\left(\widehat{\boldsymbol{\gamma}}\right)^{-1}\right]}, \tag{19}$$

$$SE\left(\widehat{\boldsymbol{\gamma}}\right)^{(3)} = \sqrt{\text{diag}\left[\mathbf{I}\left(\widehat{\boldsymbol{\gamma}}\right)^{-1}\mathbf{J}\left(\widehat{\boldsymbol{\gamma}}\right)\mathbf{I}\left(\widehat{\boldsymbol{\gamma}}\right)^{-1}\right]}, \tag{20}$$

where the second set is approximated by the approach of Louis (1982) and the third set is a sandwich estimator.

After item parameters are estimated, the program can compute two estimates of θ: the maximum likelihood (ML) estimate and the posterior mean (or expected a posteriori; EAP). The ML estimator of θ for test taker i is

$$\widehat{\theta}_i = \arg\max_{\theta} f\left(\mathbf{s}_i|\theta\right), \tag{21}$$

and the EAP is

$$\overline{\theta}_i = \int \theta f\left(\theta|\mathbf{s}_i,\widehat{\boldsymbol{\gamma}}\right)d\theta. \tag{22}$$

A benefit of computing EAP estimates and their variances is that it becomes straightforward to compute model-based reliabilities of θ (Kim, 2012). The posterior mode (or Bayesian modal estimate) is not computed but is given for completeness:

$$\widetilde{\theta}_i = \arg\max_{\theta} f\left(\theta|\mathbf{s}_i,\widehat{\boldsymbol{\gamma}}\right). \tag{23}$$

Model Fit

To evaluate relative model fit, the program produces the Akaike information criterion (AIC; Akaike, 1974) and the Bayesian information criterion (BIC; Schwarz, 1978). To assess absolute model fit, the program can produce generalized residuals for the ISF. These residuals can be found by defining an *observed ISF* as (Haberman, Sinharay, & Chon, 2013)

$$\widetilde{S}_j(\theta) = \frac{n^{-1}\sum_{i=1}^{n} s_{ij} f\left(\theta|\mathbf{s}_i\right)}{g(\theta)}, \tag{24}$$

where

$$g(\theta) = n^{-1}\sum_{i=1}^{n} f\left(\theta|\mathbf{s}_i\right). \tag{25}$$

Next, the *fitted ISF* is based on Equation 4 with estimated parameters and is denoted here as $\widehat{S}_j(\theta)$. If we let $\Delta_{S_j}(\theta) = \widetilde{S}_j(\theta) - \widehat{S}_j(\theta)$, then a generalized residual can be defined by

$$z\left[\Delta_{S_j}(\theta)\right] = \frac{\Delta_{S_j}(\theta)}{\sigma\left[\Delta_{S_j}(\theta)\right]}, \tag{26}$$

where the variance of the residual can be estimated by (Haberman & Sinharay, 2013, Equation 46)

$$s^2\left[\Delta_{S_j}(\theta)\right] = \left[ng(\theta)\right]^{-2}\sum_{i=1}^{n}\left\{f\left(\theta|\mathbf{s}_i\right)\left[s_{ij}-\widehat{S}_j(\theta)\right]-\left[\mathbf{h}_j(\theta)\right]'\nabla\ell_i\left(\widehat{\boldsymbol{\xi}}\right)\right\}^2, \tag{27}$$

where

$$\mathbf{h}_j(\theta) = \mathbf{J}\left(\widehat{\boldsymbol{\xi}}\right)^{-1}\sum_{i=1}^{n} f\left(\theta|\mathbf{s}_i\right)\left[s_{ij}-\widehat{S}_j(\theta)\right]\nabla\ell_i\left(\widehat{\boldsymbol{\xi}}\right). \tag{28}$$

If the model holds, the distribution of the residual $z\left[\Delta_{S_j}(\theta)\right]$ converges to a standard normal (Haberman et al., 2013). More details and some simulation results are described in van Rijn and Ali (2018).

Response Time Metric

We would like to point out an important note on changing the metric of response times and the time limits (after the data have been collected). If the 2P-SARM is used, a slope parameter is estimated for each item. This means that, for this model, it does not matter how the time limits or the time metric are chosen, because differences in time limits between items will be compensated by the estimation of the slope parameters. For example, if the time limits are different for different items and half of the item limits (and the associated response times) are multiplied by 2, the slope estimates will simply be divided by 2, and the resulting log-likelihood will be the same. This also holds for using different weights for items and the discrimination parameters in case the 2PLM is fitted to item response data.[2]

For the 1P-SARM, however, changing the time metric between items does impact the estimation, because only a single slope parameter is estimated for all items. Note that if the time limits (and responses times) for all items are multiplied by 2, then the slope estimate will be divided by 2, and the resulting log-likelihood will again be the same. A comparison of different models and different time metrics is discussed in the example.

For these reasons, the SARM program internally rescales the time limits and response times so that the geometric mean of the time limits is equal to 1 (or, equivalently, the mean of the log time limits is 0). Another reason for doing this is to prevent numerical underflow and overflow. After the estimation is completed, the time limits and response times are scaled back to the original metric, as specified in the input (see the following section).

Input Specification

The input consists of at least a data file and a control file that makes use of FORTRAN namelists. The program is named SARM: speed-accuracy response modeling. For Windows operating systems, it is helpful to set the path variable so that it contains the location of the executable. If the input file is input.txt, then the program can be invoked with the following command: sarm < input.txt.

The program can also be invoked through other programs (e.g., Emacs, R) and can be used for batch processing (e.g., for simulations). In the control file, the following namelists are used:

- runtitle
- dataspec
- modelspec
- estimspec
- quadspec
- itemspec
- startspec
- output
- isfspec (optional)

Descriptions of the variables within each namelist follow.

runtitle

This namelist contains two variables:

- title
- filestem

title

This variable is a character variable with maximum length of 80. The default value is "Speed-Accuracy Response Model."

filestem

This variable is a character variable with maximum length of 80. It is used as the stem for naming the output files. The default value is "sarm."

dataspec

This namelist contains four variables:

- ni
- no
- id
- datafile

ni

This integer variable is the number of items and should be larger than 1.

no

This positive integer variable is the number of test takers and should be larger than 1.

id

This logical variable indicates whether test-taker identifications should be read. If it has a value of true, the first column in the data file will be read as a character variable with maximum length of 80.

datafile

This character variable is the data file. Currently the program only reads a comma-separated-values (CSV) file. If *id* is false, the file should contain a $no \times (2 \times ni)$ data matrix. The first *ni* columns should contain the item responses, and the second *ni* columns should contain the response times. If *id* is true, the first column should contain the test-taker identifications, and the data matrix should start in the second column. The item response data are read as integer variables, and the response time data are read as real variables. The default value of datafile is "data.csv."

For item response data, any values other than 0 and 1 are treated as missing. For response time data, any values smaller than 0 or larger than the time limit are treated as missing. If either the response or the response time is missing, the associated score is treated as missing at random.

modelspec

This namelist contains the following variable:

- modelpar

modelpar

This integer variable can take values 1 and 2. If the value is 1, then the 1P-SARM is estimated. If the value is 2, then the 2P-SARM is estimated. The default value is 2.

estimspec

This namelist contains the following variables:

- nr
- maxitnr
- tolnr
- maxitemstart
- tolemstart
- maxitem
- tolem
- maxitmstep
- tolmstep

All convergence criteria are defined in terms of the relative change in log-likelihood, as follows:

$$\frac{\ell\left(\widehat{\boldsymbol{\gamma}}_k\right) - \ell\left(\widehat{\boldsymbol{\gamma}}_{k+1}\right)}{\widehat{\ell(\boldsymbol{\gamma}_{k+1})}}, \tag{29}$$

where k is an iteration index. The program throws warnings if any of the convergence criteria are not reached after the associated maximum number of iterations.

nr

This logical variable indicates whether the NR (true) or EM (false) algorithm is employed. The default value is true. It is well known that NR converges quickly if it is in the neighborhood of the solution (Lindstrom & Bates, 1988). However, if the algorithm starts far from the solution, NR can have problems. Also, EM is known to be stable, but convergence is linear (at best) and thus can be slow. For these reasons, if nr is true, the estimation starts with EM iterations and switches to NR if a somewhat more lenient convergence criterion (tolemstart) or the maximum number of starting EM iterations (maxitemstart) is reached. This approach is efficient when the number of items is not too large, say, fewer than 100. For larger problems, the EM algorithm is preferred, because it involves fewer computations per iteration. EM is used if nr is false.

maxitnr

This integer variable is the maximum number of NR iterations. The default value is 50.

tolnr

This real variable is the convergence criterion for NR iterations. The default value is .00000001.

maxitemstart

This integer variable is the maximum number of starting EM iterations. The default value is 25.

tolemstart

This real variable is the convergence criterion for starting EM iterations. The default value is .0001.

maxitem

This integer variable is the maximum number of EM iterations. The default value is 500.

tolem

This real variable is the convergence criterion for EM iterations. The default value is .00000001.

maxitmstep

This integer variable is the maximum number of iterations in the M-step of the EM algorithm. The default value is 10.

tolmstep

This real variable is the convergence criterion for iterations in the M-step of the EM algorithm. The default value is .00000001.

quadspec

The namelist quadspec contains the specification of the quadrature integration methods. Adaptive GH quadrature using the posterior mean and standard deviation is used to approximate all integrals (Naylor & Smith, 1982). The namelist contains the following variables:

- nquad
- maxitaq
- tolaq

nquad

This integer variable is the number of GH quadrature points. The default value is 10, which should be sufficient for most purposes. The maximum value is 30. If in doubt (e.g., when the number of items is large), it is useful to plot the individual posterior means (EAPs) against the posterior variances (e.g., van Rijn, 2014). If complete data are available, the plot should show a parabola that opens upward.

maxitaq

This integer variable is the maximum number of adaptive quadrature (AQ) iterations. The default value is 10.

tolaq

This real variable is the convergence criterion for AQ iterations. The default value is .00000001.

itemspec

This namelist contains six variables:

- itemname
- w0
- w1
- tl
- useitemfile
- itemfile

itemname

This character vector of length *ni* contains the item names, which can have a maximum length of 80. The default values are "Item_001," "Item_002," and so on.

w0

This real vector of length *ni* contains the weights for incorrect responses. The default values are −1.

w1

This real vector of length *ni* contains the weights for correct responses. The default values are 1.

tl

This real vector of length *ni* contains the item time limits. There is no default.

useitemfile

This logical variable indicates whether item specifications should be read from a file. The default value is false.

itemfile

This character variable indicates the name of the CSV file containing the item specifications. The order and type of the columns should be as follows: itemname (character), w0 (real), w1 (real), and tl (real).

startsspec

This namelist contains three variables:

- startpar
- usestartfile
- startfile

startpar

This real vector contains the starting values for the item parameters. The default value is 0 for intercept parameters and 1 and for slope parameters.

usestartfile

This logical variable indicates whether starting values should be read from a CSV file. The default is false.

startfile

This character variable indicates the name of the CSV file that contains the starting values. If usestartvalue is true, then the default name is "start.csv." The first column is skipped so that the output parameter file can be used directly as input, which enables continuation runs.

output

This namelist provides the output specifications. It contains the following logical variables:

- printoutput

```
&runtitle filestem='listen_1p_sarm' /
&dataspec ni=17 no=9355 datafile='data_list.csv' /
&modelspec modelpar=1/
&paramspec /
&quadspec /
&itemspec useitemfile=T /
&output /
&isfspec /
```

Figure 2 Example of basic control file.

- printparam
- printml
- printeap
- printpost
- printparamcov
- printisf

All output is produced in the form of CSV files. The default filenames are "sarm_output.csv," "sarm_param.csv," and so on (see the filestem section).

printoutput

If this variable is true, then basic output of the estimation is printed to the output file. This includes iteration progress, final log-likelihood, number of estimated parameters, AIC, and BIC. The default value is true.

printparam

If this variable is true, then the estimated parameters and three sets of standard errors are printed to a separate file. The default value is true.

printml

If this variable is true, then the ML estimates and their variances of the latent variable θ are printed to a separate file. The default value is true.

printeap

If this variable is true, then the individual posterior means (EAPs) and their variances of the latent variable θ are printed to a separate file. The default value is true.

printpost

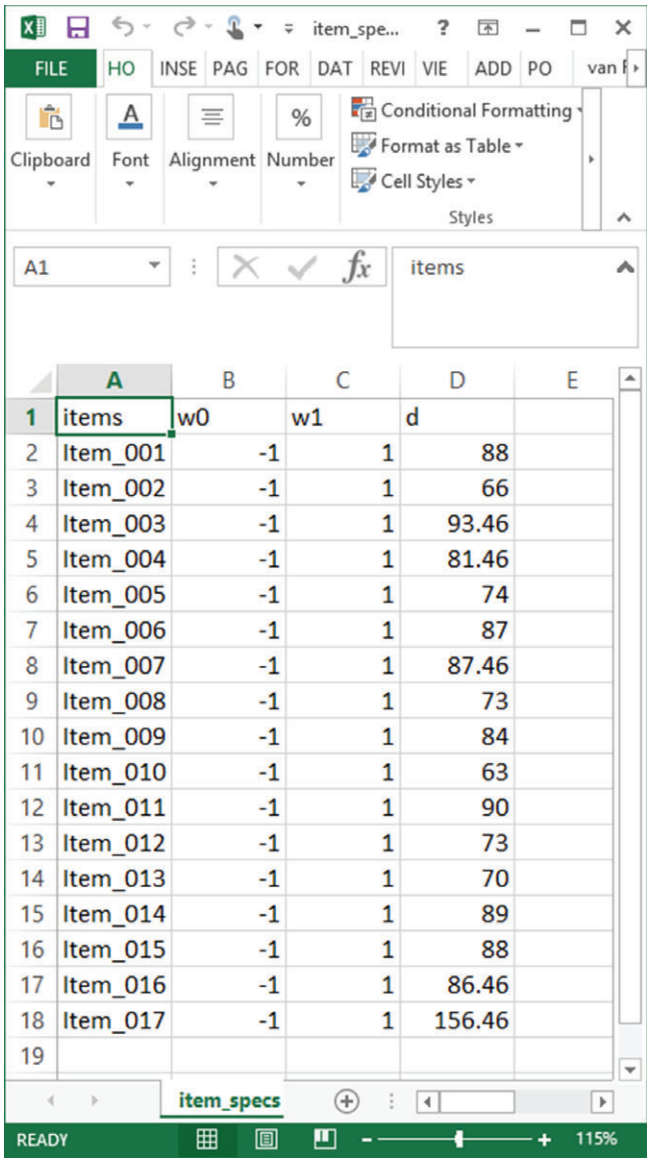
If this variable is true, then the individual posteriors are printed to a separate file. The default value is true.

printparamcov

If this variable is true, then the parameter covariance matrix is printed to a separate file. This matrix is the inverse of the observed information matrix. The default value is true.

printisf

If this variable is true, then the ISFs are printed. The file contains the item number, θ-value, observed score, fitted score, residual score, standard deviation of the residual, and adjusted residual. The default value is true.



	A	B	C	D	E
1	items	w0	w1	d	
2	Item_001	-1	1	88	
3	Item_002	-1	1	66	
4	Item_003	-1	1	93.46	
5	Item_004	-1	1	81.46	
6	Item_005	-1	1	74	
7	Item_006	-1	1	87	
8	Item_007	-1	1	87.46	
9	Item_008	-1	1	73	
10	Item_009	-1	1	84	
11	Item_010	-1	1	63	
12	Item_011	-1	1	90	
13	Item_012	-1	1	73	
14	Item_013	-1	1	70	
15	Item_014	-1	1	89	
16	Item_015	-1	1	88	
17	Item_016	-1	1	86.46	
18	Item_017	-1	1	156.46	
19					

Figure 3 Example of item specifications file.

isfspec

If the variable printisf is true, then the namelist isfspec is read from the control file. There are three variables:

- nthetas
- mintheta
- maxtheta

nthetas

This integer variable is the number of θ values used to compute the ISF. The default value is 31.

mintheta

This real variable is the minimum value of θ. The default value is −3.

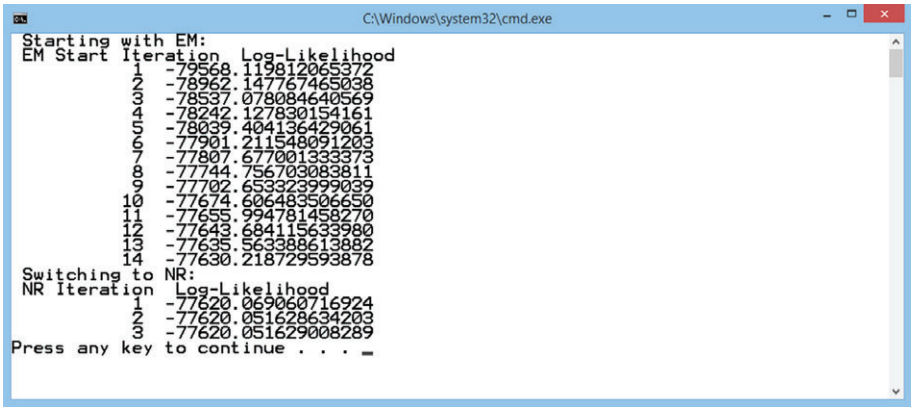


Figure 4 Hybrid expectation – maximization and Newton – Raphson iterations for listening example.

maxtheta

This real variable is the maximum value of θ. The default value is 3.

Example

As an example, we discuss the analysis of the item responses and response times of 9,385 test takers to 17 four-choice items that are part of a listening section of an international assessment of English language skills. Although there are no item-specific time limits, test takers have 10 minutes to answer all 17 items in this section. We used the default scoring rule, which assigns a weight of −1 to incorrect responses and a weight of 1 to correct responses. Because no item-level time limits were used in the administration, the time limits were set at the 99th percentile of the response time distribution of each item. To keep the full sample, the top 1% of response times were truncated at the corresponding item time limit (so that the score is 0, not missing).

If default settings are used, specification of the control file is quite minimal. Figure 2 shows an example of a control file for fitting the 1P-SARM to the listening data. The program can be invoked in two different ways: (a) using the command prompt and (b) using another program (e.g., R). Using the first option, the program can be started by

sarm < list_1p_sarm.txt.

Using the second option with R, the program will start by entering shell ("sarm < list_1p_sarm.txt") in R. Note that in R, the working directory needs to be set. In addition, as noted before, in Windows, it is needed to set the path variable so that it includes the path to the executable.

Figure 3 shows the item specifications file. Note that the order in which columns are read is fixed.

Figure 4 shows the NR iterations with EM start iterations when the program is called from a Windows command prompt. The algorithm uses 14 EM start iterations and 3 NR iterations. If *nr* is set to false (*nr* = F) in the namelist &estimspec, then the program will use EM iterations only. The EM iterations are shown in Figure 5. It can be seen that, with 35 iterations, the EM algorithm converges more slowly than the hybrid EM-NR algorithm with a total of 17 iterations (14 + 3).

Table 1 shows the estimated item parameters for the 2P-SARM in the rescaled metric. Although the three *SE*s are similar for most items, there are items for which they are quite different. For example, Item 17 has the highest slope estimate, and the sandwich *SE* is almost twice as large as the Louis *SE* (.045 vs. .023). Large differences in the different types of *SE*s can indicate model misfit (e.g., White, 1982). In this case, the sandwich *SE* would be more robust than the other two.

Table 2 shows relative model fit statistics of different models that were fitted to the listening data. For comparison reasons, we also fitted the 1PLM and 2PLM to the item response data using the MIRT software developed by Haberman (2013), the results for which are shown in the first two rows. The AIC is better for the 2PLM than for the 1PLM, but the BIC is not. This is related to the large sample size, which affects the penalty term used in the BIC but not in the AIC.

For the 1P-SARM and 2P-SARM, we made use of the original time metric with different time limits for each item based on the 99th percentile of the associated response time distribution and a rescaled time metric so that the time limits could all be fixed at 1. This rescaling was done to prevent large differences in item time limits to have an impact

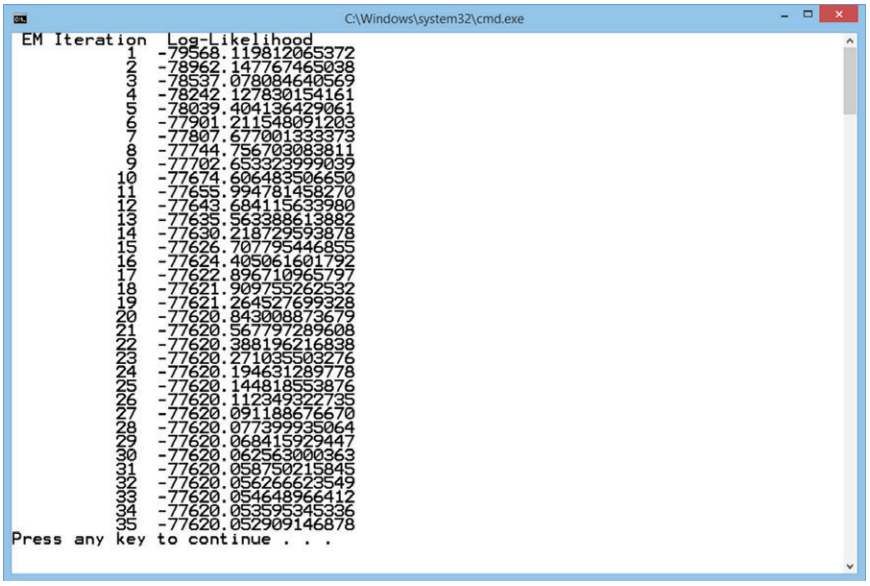


Figure 5 Expectation – maximization iterations for listening example.

on the modeling. As noted before, these different time metrics should have an impact on the 1P-SARM but not on the 2P-SARM. This is confirmed by the last four rows of Table 2. The 1P-SARM shows better fit statistics when the original time metric is used than when the rescaled metric is used. This means that, overall, the items with a higher time limit have somewhat better discriminating power. In the last two rows of the table, it can be seen that using different time metrics does not impact the fit of the 2P-SARM. Note that the reliability of the 2P-SARM is considerably larger than that of the 2PLM, which is due to the added information of response times under the 2P-SARM (Maris & van der Maas, 2012, Figure 6).

Figures 6 and 7 show observed and fitted ISFs in the rescaled metric for the 1P-SARM and 2P-SARM, respectively. The observed ISF is indicated by a dashed line, and the thinner dashed lines indicate a 95% confidence interval, whereas the solid line indicates the ISF as determined by the estimated model parameters. Note that the confidence intervals are quite narrow owing to the large sample. For the 1P-SARM, 78% of the ISF residuals is significant at a nominal alpha of .05 (in comparison, 63% of the residuals associated with the item response functions are significant for the 1PLM). For the 2P-SARM, this percentage is 60 (compared to 51 for the 2PLM). Comparing the two figures, it can be seen that the 2P-SARM leads to considerable improvement in fit over the 1P-SARM, most notably for Items 1 and 13. However, comparing the residuals for the 2PLM (51% significant) and the 2P-SARM (60% significant), the 2P-SARM performs worse. It should be noted that test takers were not informed that response time would be used for scoring, and the extent to which this may affect model fit cannot be determined with the current data. Finally, although adding response times seems to add to measurement precision, illustrated by the higher θ reliabilities, it also adds to misfit.

As a final illustration, we compare the reliabilities of θ for the 2P-SARM under different scoring rules. We fitted this model with four different weights for incorrect responses: −1, −1/2, −1/3, −1/4. The weights for the correct responses were fixed at 1. Note that we cannot compare relative model fit statistics AIC and BIC, because the different weights lead to different data on which the models are fitted. Table 3 shows the θ reliabilities for the four different rules. It can be seen that the reliabilities decrease as the weights for incorrect responses increase. However, even with a penalty of 1/4 for incorrect responses, the θ reliability is somewhat larger for the 2P-SARM than for the 2PLM (.761 vs. .754).

Discussion

In this report, we illustrated how to estimate speed-accuracy response models with the computer program SARM. Several features were discussed, and data from a listening test were analyzed. We would like to finish with some limitations and potentially useful additions.

Table 1 Item Parameter Estimates and Standard Errors of 2P-SARM for Listening Data

Parameter	Estimate	*SE*	*SE*_Louis	*SE*_Sandwich
Item_001_Intercept	1.211	0.021	0.020	0.022
Item_002_Intercept	2.443	0.036	0.042	0.031
Item_003_Intercept	0.142	0.018	0.018	0.019
Item_004_Intercept	0.621	0.023	0.023	0.023
Item_005_Intercept	1.777	0.029	0.034	0.025
Item_006_Intercept	0.038	0.020	0.020	0.021
Item_007_Intercept	0.507	0.021	0.021	0.022
Item_008_Intercept	1.878	0.029	0.035	0.025
Item_009_Intercept	1.075	0.023	0.029	0.020
Item_010_Intercept	2.408	0.037	0.051	0.028
Item_011_Intercept	0.567	0.021	0.025	0.018
Item_012_Intercept	1.478	0.027	0.030	0.025
Item_013_Intercept	1.214	0.029	0.031	0.028
Item_014_Intercept	0.274	0.019	0.020	0.020
Item_015_Intercept	0.876	0.021	0.023	0.020
Item_016_Intercept	0.992	0.022	0.021	0.023
Item_017_Intercept	0.487	0.020	0.018	0.023
Item_001_Slope	0.508	0.024	0.023	0.025
Item_002_Slope	1.246	0.037	0.046	0.032
Item_003_Slope	0.712	0.022	0.021	0.024
Item_004_Slope	0.988	0.027	0.027	0.028
Item_005_Slope	1.016	0.031	0.036	0.028
Item_006_Slope	0.861	0.024	0.024	0.025
Item_007_Slope	0.927	0.025	0.024	0.027
Item_008_Slope	0.943	0.031	0.036	0.028
Item_009_Slope	0.905	0.026	0.032	0.022
Item_010_Slope	1.317	0.038	0.052	0.030
Item_011_Slope	0.913	0.024	0.031	0.019
Item_012_Slope	0.924	0.030	0.032	0.029
Item_013_Slope	1.308	0.034	0.037	0.032
Item_014_Slope	0.745	0.023	0.023	0.024
Item_015_Slope	0.784	0.024	0.025	0.024
Item_016_Slope	0.799	0.026	0.024	0.028
Item_017_Slope	1.323	0.031	0.023	0.045

Table 2 Relative Model Fit for Listening Data

Model	Time metric	Parameters	Log-likelihood	AIC	BIC	Reliability (θ)
1PLM	–	18	−86,854.9	173,746	173,874	0.747
2PLM	–	34	−86,445.6	172,959	173,202	0.754
1P-SARM	Original	18	−77,620.1	155,276	155,405	0.794
1P-SARM	Rescaled	18	−77,933.9	155,904	156,032	0.787
2P-SARM	Original	34	−77,051.5	154,171	154,414	0.802
2P-SARM	Rescaled	34	−77,051.5	154,171	154,414	0.802

Note. 1PLM = one-parameter logistic model. 1P-SARM = one-parameter speed-accuracy response model. 2PLM = two-parameter logistic model. 2P-SARM = two-parameter speed-accuracy response model. AIC = Akaike information criterion. BIC = Bayesian information criterion.

Currently the program only deals with a standard normal latent variable, whereas other types of latent variables could be useful (e.g., discrete, skewed). In addition, it would be useful to add the possibility of linear constraints on the parameters, so that other model identifications can be used. Note that the addition of constraints would also enable the study of differential item functioning. Another useful addition would be the inclusion of predictors, so that latent regression models could be estimated. It should be mentioned that work is in progress to add other models to the software. For example, the hierarchical model by van der Linden (2007) would be a useful extension. Somewhat more complex would be the addition of multidimensional latent variables and, particularly, of such extensions with mixed model types. For example,

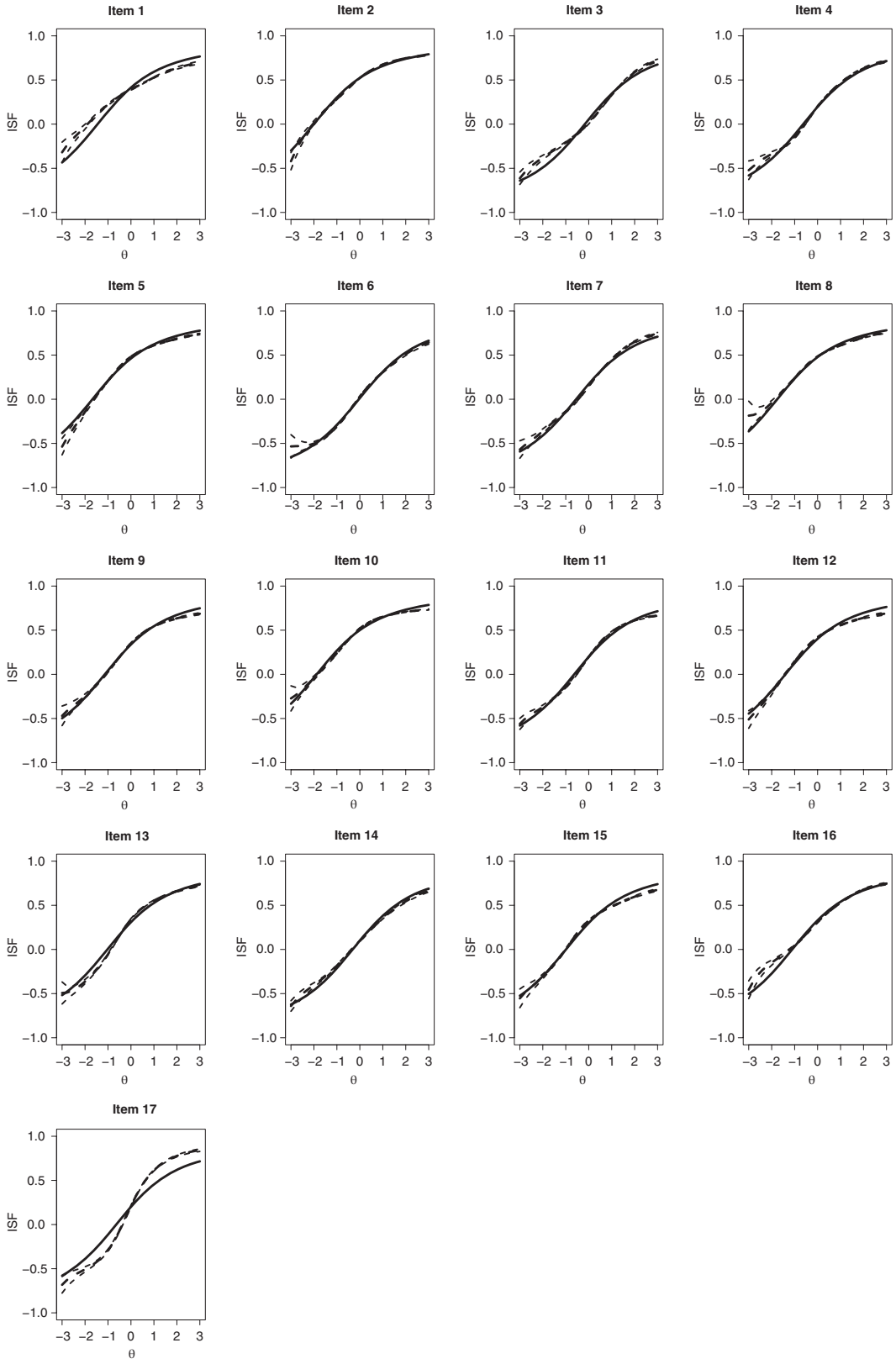


Figure 6 Item fit for one-parameter speed-accuracy response model.

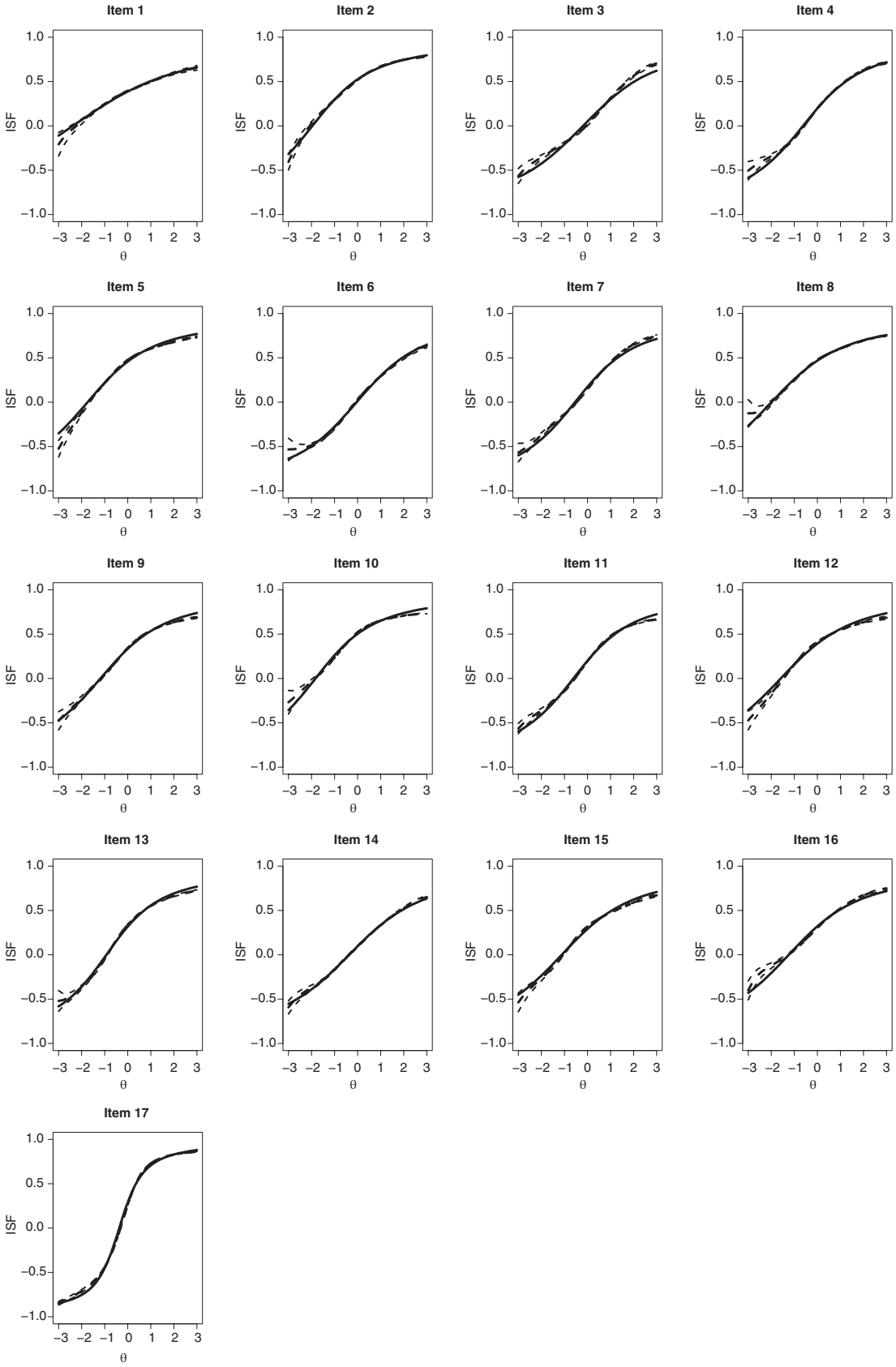


Figure 7 Item fit for two-parameter speed-accuracy response model.

Table 3 Reliabilities for Two-Parameter Speed-Accuracy Response Model With Different Scoring Rules for Listening Data

Scoring rule	Reliability (θ)
$w_0 = -1$	.802
$w_0 = -1/2$	.785
$w_0 = -1/3$	.771
$w_0 = -1/4$	.761

a between-item model with a 2PLM in one dimension and a 2P-SARM in another dimension could be implemented. This could be useful if tests are constructed in which item-level time limits are used in some sections but not in others. Finally, research would be needed on such combined modeling approaches.

Acknowledgment

The authors are indebted to Shelby J. Haberman for sharing source code for matrix inversion and for assistance with computing generalized residuals.

Notes

1 The executable associated with this software manual is freely available for noncommercial use upon request via *sarm@ets.org*.
2 That is, for the 2PLM, item weights have no impact on model fit or measurement precision.

References

Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control, 19*, 716–723.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*, 1–38. https://doi.org/10.1.1.133.4884

Fahrmeir, L., & Tutz, G. (2001). *Multivariate statistical modelling based on generalized linear models* (2nd ed.). New York, NY: Springer. https://doi.org/10.1007/978-1-4757-3454-6

Haberman, S. J. (2013). *A general program for item-response analysis that employs the stabilized Newton–Raphson algorithm* (Research Report No. RR-13-32). Princeton, NJ: Educational Testing Service. https://doi.org/10.1002/j.2333-8504.2013.tb02339.x

Haberman, S. J., & Sinharay, S. (2013). Generalized residuals for general models for contingency tables with application to item response theory. *Journal of the American Statistical Association, 108*, 1435–1444. https://doi.org/10.1080/01621459.2013.835660

Haberman, S. J., Sinharay, S., & Chon, K. H. (2013). Assessing item *t* for unidimensional item response theory models using residuals from estimated item response functions. *Psychometrika, 78*, 417–440. https://doi.org/10.1007/s11336-012-9305-1

Kim, S. (2012). A note on the reliability coefficients for item response model-based ability estimates. *Psychometrika, 77*, 153–162. https://doi.org/10.1007/s11336-011-9238-0

Lindstrom, M. J., & Bates, D. M. (1988). Newton–Raphson and EM algorithms for linear mixed effects models for repeated-measures data. *Journal of the American Statistical Association, 83*, 1014–1022. https://doi.org/10.1080/01621459.1988.10478693

Louis, T. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society, Series B, 44*, 226–233. https://doi.org/10.2307/2345828

Maris, G., & van der Maas, H. L. J. (2012). Speed-accuracy response models: Scoring rules based on response time and accuracy. *Psychometrika, 77*, 615–633. https://doi.org/10.1007/s11336-012-9288-y

Naylor, J. C., & Smith, A. F. M. (1982). Applications of a method for efficient computation of posterior distributions. *Applied Statistics, 31*, 214–225. https://doi.org/10.2307/2347995

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*, 461–464.

van der Linden, W. J. (2007). A hierarchical framework for modeling speed and accuracy on test items. *Psychometrika, 72*, 287–308. https://doi.org/10.1007/s11336-006-1478-z

van Rijn, P. W. (2014). Reliability of multistage tests using item response theory. In D. Yan, A. A. von Davier, & C. Lewis (Eds.), *Computerized multistage testing: Theory and applications* (pp. 251–263). Boca Raton, FL: Chapman and Hall/CRC.

van Rijn, P. W., & Ali, U. S. (2017). A comparison of item response models for accuracy and speed of item responses with applications to adaptive testing. *British Journal of Mathematical and Statistical Psychology, 70*, 317–345. https://doi.org/10.1111/bmsp.12101

van Rijn, P. W., & Ali, U. S. (2018). A generalized speed-accuracy response model for dichotomous items. *Psychometrika*, *83*, 109–131. https://doi.org/10.1007/s11336-017-9590-9

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica, 50*, 1 – 25. https://doi.org/10.2307/1912526
Yuan, K.-H., Cheng, Y., & Patton, J. (2014). Information matrices and standard errors for MLEs of item parameters in IRT. *Psychometrika, 79*, 232 – 254. https://doi.org/10.1007/S11336-013-9334-4

Suggested citation

van Rijn, P. W., & Ali, U. S. (2018). *SARM: A computer program for estimating speed-accuracy response models for dichotomous items* (Research Report No. RR-18-15). Princeton, NJ: Educational Testing Service. https://doi.org/10.1002/ets2.12200

Action Editor: Shelby Haberman

Reviewers: Tim Davey and Paul Jewsbury

ETS, the ETS logo, and MEASURING THE POWER OF LEARNING are registered trademarks of Educational Testing Service (ETS). All other trademarks are property of their respective owners

Find other ETS-published reports by searching the ETS ReSEARCHER database at http://search.ets.org/researcher/