

Modern Psychometric Theory: Using SAS to Calculate IRT Ability Estimates

Jeffrey D. Kromrey, Cynthia G. Parshall, Walter M. Chason, and Qing Yi

University of South Florida

ABSTRACT

This paper provides a demonstration of code written in SAS/IML® that provides maximum likelihood estimates of examinee ability under the 1, 2, or 3 parameter logistic models used in modern psychometric theory. The program reads a file of examinee scored responses (1/0s) to a set of test items and a file of item parameter. The program implements the Newton-Raphson algorithm to obtain maximum likelihood estimates of examinee ability, then outputs a file of ability estimates and warning flags for examinees with response patterns that did not provide convergence of the algorithm. The program is written so that the convergence criterion and the maximum number of iterations for the estimation may be easily modified.

Modern psychometric theory evaluates examinee responses to test items as probabilistic models based on logistic functions of item parameters (difficulty, discrimination, and guessing) and examinee parameters (abilities). The benefits of such models include computer adaptive testing, advances in test equating, and new developments in item bias analysis.

INTRODUCTION

Item response theory (IRT) represents an application of probabilistic models to indicate the interaction between an examinee's ability (Θ) and the characteristics of items in a test. Although such models may be based on either the normal ogive or logistic functions, the latter have proven to be more useful in most applications. A clear, general introduction to IRT has been provided by Hambleton, Swaminathan and Rogers (1991), and recent advances are summarized in van der Linden and Hambleton (1996). The roots of IRT, in large part, are in the early work of Frederick Lord (e.g., Lord,

1952, 1953a, 1953b).

The logistic models of IRT may be represented by

the number of parameters included:

$$1 \text{ parameter: } P(\Theta) = 1 / (1 + e^{-D(\Theta - b)})$$

$$2 \text{ parameter: } P(\Theta) = 1 / (1 + e^{-Da(\Theta - b)})$$

$$3 \text{ parameter: } P(\Theta) = c + (1-c) / (1 + e^{-Da(\Theta - b)})$$

where

$P(\Theta)$ = the probability of a correct response for an examinee with ability Θ ,

b = the item difficulty parameter,

a = the item discrimination parameter,

c = the lower asymptote or guessing parameter,

D = a scaling factor, 1.702, and

e = the base of the natural logarithms.

These models represent the relationship between examinee ability (Θ) and item characteristics (a , b , c). The IRT model that incorporates 1-parameter includes only the item difficulty parameter (b), the 2-parameter model includes both difficulty and discrimination parameters (a and b), while the 3-parameter model includes a , b , and c . The item parameters and examinee ability are usually estimated through a maximum likelihood approach. Because this paper will focus on the estimation of examinee ability, a brief treatment of the equations related to this estimation will be provided. Hambleton and Swaminathan (1985) provide a lucid treatment of the estimation of the item parameters.

The likelihood of an item response pattern for an examinee with ability Θ is given by

$$L(U|\Theta) = \prod P_i^{U_i} Q_i^{(1-U_i)}$$

where

U = vector of item scores (1/0) with elements U_i

P_i = probability of correct response to item i ,

$Q_i = 1 - P_i$, and

Π = the multiplication operator.

The natural logarithm of this likelihood (i.e., the log likelihood) is given by

$$\ln L(U|\Theta) = \sum [U_i \ln P_i + (1 - U_i) \ln Q_i].$$

Both the likelihood and the log likelihood are maximized when the first derivative of the equation

is equal to zero:

$$\frac{d}{d\Theta} \ln L(U|\Theta) = f'(x) = 0$$

The value of examinee ability that provides the maximum of the likelihood equation can be found using the Newton-Raphson algorithm (Isaacson & Keller, 1966). This algorithm provides successive approximations to the maximum likelihood value of Θ by adjusting the Θ value obtained on iteration K by the ratio of the first derivative to the second derivative, to provide the Θ value for iteration $K + 1$:

$$\Theta_{K+1} = \Theta_K - [f'(x)_K / f''(x)_K]$$

where $f'(x)_K$ and $f''(x)_K$ are the first and second derivatives of the likelihood function at the value Θ_K .

For the 3-parameter logistic model, these derivatives are given by

$$f'(x) = D \sum a_i (U_{ia} - P_{ia}) (P_{ia} - c_i) / P_{ia} (1 - c_i) \text{ and}$$

$$f''(x) = D^2 \sum a_i^2 (P_{ia} - c_i) (U_{ia} c_i - P_{ia}^2) Q_{ia} / P_{ia}^2 (1 - c_i)^2$$

where the summation is over the items on the test.

IRTSCORE.SAS

The program IRTSCORE.SAS uses SAS/IML to produce maximum likelihood ability estimates under the 1-, 2-, or 3-parameter logistic models. The program requires, as input, item parameters (e.g., the difficulty, discrimination and lower asymptote) and a file of examinee item scores (1/0 for each item on the test). The item parameters may be obtained from an item calibration program such as BILOG (Mislevy & Bock, 1990). The program checks for response patterns that are not amenable to maximum likelihood ability estimates (all items answered correctly or all answered incorrectly). In addition, the program flags examinees for whom the algorithm did not converge (assigning an ability estimate of -9). Response patterns in which difficult items are answered correctly, but easy items are missed sometimes yield likelihood surfaces that do not converge with the Newton-Raphson algorithm. The output from the program is a SAS data set containing the ability estimates.

The program IRTSCORE.SAS operates in eight major steps, as follows:

1. *Read IRT item parameters.* The item parameters for each test item are read from an external file, in a SAS data step (DATA ITEM_EST), then passed to SAS/IML. The item parameter file has one record for each test item, and three variables (a, b, c). For the IRT models with 1- or 2-parameters, the c parameter is set to zero.

2. *Read examinee item score file.* As with the IRT item parameter file, the item score file is read from an external file in a SAS data step (DATA ITEMRESP), then passed to SAS/IML. The item score file has one record for each examinee and a separate variable for each item on the test. The variables' values are either 1 (for correct responses) or 0 (for incorrect responses).

3. *Screen for all correct and all incorrect responses.* For each examinee, a check is provided to determine if all items were answered correctly or all answered incorrectly. For these response patterns, the Newton-Raphson algorithm cannot be implemented and maximum likelihood estimates of ability do not exist. By convention, an ability estimate of 4.00 is assigned to examinees if all items are answered correctly and an ability estimate of -4.00 is assigned if all are answered incorrectly. For these examinees, the program skips to step 7 (below). If at least one item is answered correctly and at least one is answered incorrectly, the program proceeds to step 4.

4. *Establish initial ability estimate.* The initial estimate of examinee ability, for use in the Newton-Raphson algorithm is $\ln(N_{\text{right}}/N_{\text{wrong}})$, where N_{right} is the number of items answered correctly, N_{wrong} is the number answered incorrectly, and $\ln$ is the natural logarithm. This ability estimate is placed in a NITEMS X 1 array (THTVEC1).

5. *Execute Newton-Raphson iterations.* Using $\ln(N_{\text{right}}/N_{\text{wrong}})$ as Θ_K , the revised ability estimate, Θ_{K+1} , is obtained as $\Theta_{K+1} = \Theta_K - [f'(x)_K / f''(x)_K]$, where $f'(x)_K$ is the first derivative and $f''(x)_K$ is the second derivative of the log likelihood function. These derivatives are obtained from the LL subroutine described in step 6, below. The change in ability estimates, $ABS(\Theta_{K+1} - \Theta_K)$ is then evaluated. If the change is less than the convergence criterion (in the program illustrated, the convergence criterion is 0.0000001), the iterations stop. If the change is greater than the convergence criterion, then Θ_{K+1} becomes the current ability estimate, and the loop is repeated. If the loop executes 1000 times without convergence, a value of -9 is assigned as the ability

estimate. This value is not interpreted as a valid estimate of ability, but rather serves as a flag that the algorithm did not converge.

6. Derivatives of the log likelihood function. The first and second derivatives of the log likelihood function are computed in the subroutine LL. The inputs to this subroutine are the number of items on the test (the scalar quantity NITEMS), a 1 X NITEMS vector representation of the current estimate of examinee ability (THTVECTR), a vector of ones and zeros that represent correct and incorrect responses to test items (U), and vectors of item difficulty parameters (POPB), item discrimination parameters (POPA), and lower asymptote parameters (POPC). The subroutine returns the value of the log likelihood of the observed item responses, given the current examinee ability estimate and the item parameters (LOGLIKE), and the first and second derivatives of the log likelihood function (FIRSTDER and SECNDDER, respectively).

7. Create SAS data set. The ability estimate for each examinee is placed in the vector THT1. When ability estimates are computed for all examinees in the item score file, this vector is transferred to a SAS data set (WORK.THETAEST) and given the variable name THETAHAT.

8. Identify examinees for whom the algorithm did not converge. The data set WORK.THETAEST is partitioned into two data sets. WORK.OOPS will contain examinees (if any) for whom the algorithm did not converge (i.e., assigned a value of -9 in step 5, above). The data set WORK.Scores contains the ability estimates for all other examinees.

PROGRAM CODE

```
data item_est;
* +-----+
  These are item parameter estimates
  (a,b,c) for each item on the test
  +-----+;

infile 'c:\itemests.out'
input ahat bhat chat;
data itemresp;

* +-----+
  This is the file of item scores
  (1,0) for each examinee on each
  item on the test
  +-----+;
```

```
infile 'c:\itemresp.dat';
input @1 (rsp1 - rsp80)(1.);
```

```
proc iml;
  use item_est;

* +-----+
  Reading in the vectors of item parameter estimates
  +-----+;

  read all var {ahat} into popa;
  read all var {bhat} into popb;
  read all var {chat} into popc;
  nitms=nrow(ahat);

* +-----+
  Define the subroutine to calculate the log-
  likelihood given item parameter estimates,
  an examinee response vector, and an estimate
  theta (tht).
  +-----+;

start LL (thtvectr, U, nitms, popa, popb, popc,
loglike, firstder, secnnder);

  matdev = thtvectr - popb`;

* +-----+
  The following yields a vector of probabilities of
  correct responses on each item (pi) and
  probabilities of incorrect responses (qi).
  +-----+;

  pi = popc` + ((1 - popc`) # ((1 + exp(matdev *
diag(-1.7*popa))))#(-1));
  qi = J(1,nitms) - pi;

* +-----+
  Log likelihood
  +-----+;

  lnpi=log(pi);
  lnqi=log(qi);
  u1 = J(1,nitms) - u;
  loglike = (u*lnpi`)+(u1*lnqi`);

* +-----+
  First derivative
  +-----+;

  ex = exp(matdev * diag(-1.7*popa));
  ex2 = exp(matdev * diag(1.7*popa));
  fder1=1.7#popa`#(1 - popc`);
  fder2=(ex2+2+ex)##(-1);
```

```

fder= fder1#fder2;
fder = (u-Pi)#(fder)#((Pi#Qi)##(-1));
firstder = fder*J(nitems,1);

* +-----+
Second derivative
+-----+;

sder1=((1.7##2)*(popa')##2)#(1 - popc');
sder2= (popc +ex2)#(1+ex)##2;
sder = (sder1)#(sder2)##(-1);
secndder = sder*J(nitems,1);
finish;

use itemresp;

* +-----+
Reading in the item score file (0,1)
+-----+;

read all var {rsp1 rsp2 rsp3 rsp4 rsp5 rsp6 rsp7 rsp8
rsp9 rsp10 rsp11 rsp12 rsp13 rsp14 rsp15 rsp16 rsp17
rsp18 rsp19 rsp20 rsp21 rsp22 rsp23 rsp24 rsp25
rsp26 rsp27 rsp28 rsp29 rsp30 rsp31 rsp32 rsp33
rsp34 rsp35 rsp36 rsp37 rsp38 rsp39 rsp40 rsp41
rsp42 rsp43 rsp44 rsp45 rsp46 rsp47 rsp48 rsp49
rsp50 rsp51 rsp52 rsp53 rsp54 rsp55 rsp56 rsp57
rsp58 rsp59 rsp60 rsp61 rsp62 rsp63 rsp64 rsp65
rsp66 rsp67 rsp68 rsp69 rsp70 rsp71 rsp72 rsp73
rsp74 rsp75 rsp76 rsp77 rsp78 rsp79 rsp80}
into Uall;

* +-----+
Analyze each examinee response vector
+-----+;

do I=1 to nrow(Uall);
Ui = Uall[i,];
ncorrect=Ui*J(nitems,1);
nwrong=nitems - ncorrect;
if (nwrong ^= 0 & ncorrect ^=0) then do;
thetal=log(ncorrect/nwrong);
thtvec1 = thetal*j(nitems,1);
do iter = 1 to 1000
while (max(abs(chg))>.0000001);
run LL(thtvec1,Ui,nitems,popa,popb,popc,
LL1,deriv1,deriv2);
if (deriv2<0 | deriv2>0) then
thetam1 = thetal + (deriv1/deriv2);
if deriv2=0 then thetam1= thetal;
chg = thetam1-thetal;
thetal = thetam1;
if thetal > 4 then thetal = 4;
if thetal < -4 then thetal = -4;

```

```

if thetal=4 then chg=0;
if thetal=-4 then chg=0;
thtvec1=thetal*j(nitems,1);
end;

```

```
end;
```

```

* +-----+
Ability estimates assigned
in three special situations:
1. Nonconverge of estimates
2. All items correct
3. All items incorrect
+-----+;

```

```

if iter=1000 then thetal = -9;
if nwrong = 0 then thetal = 4;
if ncorrect = 0 then thetal = -4;

```

```

* +-----+
Build vector of theta estimates
+-----+;

```

```

if I = 1 then tht1 = thetal;
if I > 1 then tht1 = tht1 // thetal;
end;

```

```

* +-----+
Create SAS data set WORK.THETAEST
+-----+;

```

```

create thetaest from tht1
(COLNAME={'Thetahat'});
append from tht1;
quit;

```

```

* +-----+
Partition WORK.THETAEST into those
observations that showed nonconvergence
(WORK.OOPS), if any, and those for
whom valid ability estimates were
computed (WORK.Scores)
+-----+;

```

```

data oops;
set thetaest;
if thetahat = -9;
data scores;
set thetaest;
if thetahat > -9;
run;

```

CONCLUSION

This program has been useful to the authors in our simulation studies investigating the properties of various IRT models (see, for example, Parshall, Kromrey, Chason & Yi, 1997). The examinee ability estimates obtained with IRTSCORE.SAS correlate on the order of $r = 0.99$ with the ability estimates provided by BILOG on data files used in item calibration. The program illustrates the advantages that may be realized by combining the flexibility that is available in the SAS system (especially the flexibility related to the conduct of simulation studies) with the item parameter estimation algorithms of other programs (e.g., BILOG).

REFERENCES

- Hambleton, R. K. & Swaminathan, H. (1985). *Item response theory: Principles and applications*. Boston: Kluwer.
- Hambleton, R. K., Swaminathan, H. & Rogers, H. J. (1991). *Fundamentals of item response theory*. Thousand Oaks, CA: Sage Publications.
- Isaacson, E. & Keller, H. (1966). *Analysis of numerical methods*. New York: Wiley.
- Lord, F. M. (1952). A theory of test scores. *Psychometric Monograph*, 7.
- Lord, F. M. (1953a). An application of confidence intervals and maximum likelihood to the estimation of an examinee's ability. *Psychometrika*, 18, 57-75.
- Lord, F. M. (1953b). The relation of test score to the trait underlying the test. *Educational and Psychological Measurement*, 13, 517-548.
- Mislevy, R. J. & Bock, R. D. (1990). *BILOG3: Item analysis and test scoring with binary logistic models* [Computer program]. Chicago, IL: Scientific Software.
- Parshall, C. G., Kromrey, J. D., Chason, W. M. & Yi, Q. (1997). *Evaluation of parameter estimation under modified IRT models and small samples*. Paper presented at the annual meeting of the Psychometric Society, Gatlinburg, TN.
- van der Linden, W. J. & Hambleton, R. K. (1996, Eds.). *Handbook of modern item response theory*. New York: Springer-Verlag.

SAS/TML is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

CONTACT INFORMATION

The authors can be contacted at the University of South Florida, Department of Educational Measurement and Research, FAO 100U, 4202 East Fowler Avenue, Tampa, FL 33620, by telephone at (813) 974-3220, or Jeff can be contacted by e-mail: kromrey@typhoon.coedu.usf.edu