

# ST

## A Computer Program for IRT Scale Transformation

Version 1.0

April 1995

Brad Hanson and Lingjia Zeng

American College Testing

P.O. Box 168, Iowa City, IA 52243

ST is a program written in the C language for computing the coefficients of an Item Response Theory (IRT) scale transformation function. The  $\theta$ -scale in IRT is indeterminate in regard to location and spread. Any linear transformation of an IRT  $\theta$ -scale, with a corresponding transformation of the item parameters, will result in an identical fit to any data. The scale transformation function gives the linear transformation from one  $\theta$ -scale to another  $\theta$ -scale. This linear transformation is determined by two sets of item parameter estimates for a common set of items. The two sets of item parameter estimates define two  $\theta$ -scales which are related by a linear function. The basic IRT model used in ST is the three-parameter logistic model with  $D=1.7$  as described in Lord (1980).

Two categories of techniques for computing the scaling transformation function are computed in ST: 1) techniques based on the mean and standard deviations of the item parameters; and 2) techniques based on minimizing a loss function involving item characteristic curves. For the characteristic curve techniques ST uses a general loss-minimization technique to compute the transformation function. The scale transformation function is found by minimizing the transformation error which is defined by a criterion function. ST uses criterion functions introduced by Haebara (1980) and Stocking and Lord (1983). The transformation errors are computed in either of the two ways: cumulating the error over a set of discretized ability ( $\theta$ ) values, or numerically integrating the error over a distribution of  $\theta$  values.

### Program Operation

This manual describes the Macintosh version of ST. This description of the program assumes familiarity with IRT item parameter scaling methodology and familiarity with basic Macintosh operations.

When the program begins only the menu bar is visible. To compute an item parameter scaling select the "Compute Scaling" item from the File menu . A standard file dialog is presented to select an input data file to read (the format of the input file is described below).

After the input file has been selected a standard file dialog is presented for specifying an output file in which to write the scaling results. The scaling results are then computed and written to the output file. After the computations are complete another item parameter scaling can be computed by selecting Compute Scaling from the File menu, or the program can be terminated by selecting Quit from the File menu.

The output file produced by the program is a text file that can be read by any text editor or word processor. The output file contains the name of the input file from which the data were read, means and standard deviations of the common item parameters, and the slopes and intercepts of the scaling transformations computed using the Stocking-Lord, Haebara, Mean/Mean, and Mean/Sigma methods.

The algorithms used to computing the item parameter scalings are described in Kolen and Brennan (1995) and Zeng and Kolen (in preparation). To minimize the loss functions for the characteristic curve methods the routine DFPMIN from *Numerical Recipes in C* (Press, Teukolsky, Vetterling, & Flannery, 1992) is used.

### Format of Input Data File

The input data file should contain two required pieces of information: 1) the item parameter estimates to be scaled (referred to as the new form item parameters), and 2) scaled item parameter estimates for some of these new form items (referred to as old form item parameters). The program computes scale transformations to convert the new form item parameters to be on the same scale as the old form item parameters. The data file can also contain two optional pieces of information which are used in some item parameter scaling methods: 1) a set of new form thetas, or quadrature values and weights, and 2) a set of old form thetas, or quadrature values and weights.

Each of the four possible pieces of information in the data file begins with a information line followed by the appropriate data. The information line consists of a dollar sign in the first column followed by one or more blanks or tabs and a string describing the information to follow. For example, a data file containing six new form item parameters, three old form item parameters, and three theta values for the new and old form is given as follows:

```
This is an example data file
$ Item_Parameters   Form A
3      0.4328      -1.1013      0.1168
6      0.5727      -1.1425      0.1168
9      0.7910      -0.1419      0.1168
12     0.7409      -0.0380      0.1168
15     1.0911      0.8233      0.2859
18     1.0685      0.0506      0.0134
$ Link_Item_Parameters   Form B
3      0.4561      -1.4908      0.0791
6      0.5902      -1.5075      0.0791
```

```

9      0.6758   -0.7280    0.0791
$ Theta_Distribution   Form A
-1.0   .25
0.0    .50
1.0    .25
$ Link_Theta_Distribution   Form B
-1.3   .125
-0.78  .25
0      .25
0.80   .25
1.3    .125

```

The first line is a comment. The program will skip over any lines at the top of the file that do not have a dollar sign as the first character in the line – these lines are treated as comments. The second line has a dollar sign as the first character. This indicates that the line is a information line preceding information to be read in by the program. The text "Item\_Parameters" after the dollar sign in the second line indicates that item parameters for the new form will follow starting on the next line. The text after "Item\_Parameters" on the second line is descriptive information that is ignored by the program (any text on the line after the string "Item\_Parameters" is ignored by the program). The ninth line begins with a dollar sign indicating the end of the new form item parameters and the beginning of another piece of information to be read. The text "Link\_Item\_Parameters" in the ninth line indicates that item parameters for the old form will follow. Each line following the "Item\_Parameters" and "Link\_Item\_Parameters" information lines contains one string and three numbers: 1) an string identifying an item (maximum of eight characters), 2) the slope item parameter (a) for that item, 3) the threshold item parameter (b), and 4) the lower asymptote item parameter (c). The string and three numbers must be separated by one or more blanks or tabs.

The strings identifying the items are very important for the operation of the program as they identify the common items used in the item parameter scaling. Any item parameters with the same identifying string under both the "Item\_Parameters" information line and the "Link\_Item\_Parameters" information line are assumed to belong to the same item. The pairs of item parameters estimates corresponding to the same items are used to compute the scale transformation function.

For example, in the sample data given above there are items with identifying strings of 3, 6, and 9 for under both the "Item\_Parameters" information line (new form item parameters) and the "Link\_Item\_Parameters" information line (old form item parameters). These three pairs of item parameter estimates will be used to compute the transformation from the new form item parameter scale to the old form item parameter scale. The additional new form item parameters that are not common items are not used by the program. All old form items must be common items (there must be a corresponding new form item for each old form item).

The optional information concerning the theta distributions in the data file is identified by the "Theta\_Distribution" and "Link\_Theta\_Distribution" information lines. The value of the thetas given under "Link\_Theta\_Distribution" must be on the scale of the old form item parameters. The value of the thetas given under "Theta\_Distribution" must be on the scale of the new form item parameters. There are two formats that the theta distributions can be entered in: the summation format, and the integration format. In the summation format there is a single number on each line. This number is a theta estimate for a single examinee.

In the integration format there are two numbers on each line. The first number is the value for one quadrature point and the second number is the weight for that quadrature point. The program distinguishes between these two format by the number of values on the line following the "Theta\_Distribution" and "Link\_Theta\_Distribution" information lines – one value indicates summation format and two values indicates integration format. The integration format is used in the example given above. If the summation format is used a spaced sample of 200 theta values is used in computing the Stocking-Lord transformation (if there are fewer than 200 theta values all of them are used), and the theta values are grouped into 50 equally spaced intervals between -5 and 5 for computing the Haebara transformation.

It is possible to specify other values for the size of the spaced sample or number of intervals by using the integration format for the thetas. For example, if one wanted a spaced sample of 300 theta values, the 300 theta values could be used as quadrature values with quadrature weights of  $1/300$  for every theta. The thetas would be read in quadrature format but the result would be the same as using the 300 individual theta values in computing the Stocking-Lord scaling.

It is possible for the file to have old form thetas (the "Link\_Theta\_Distribution" information line), but no new form thetas (the "Theta\_Distribution" information line). If new form thetas are present, old form thetas must also be present, and both must be of the same format. If neither the old form nor the new form thetas are present then the Stocking-Lord and Haebara scaling methods are not computed. If only the old form thetas are present the Stocking-Lord item parameter scaling is computed, but the Haebara item parameter scaling is not computed.

The four pieces of information in the input data file may appear in any order. For example, theta distributions could be before item parameters in the file. The input data file must at a minimum contain old and new form item parameters.

## **Scripting Support**

Operation of ST can be controlled via a scripting system such as AppleScript. This allows batch operation of the program. The two commands in the File menu (Computing Scaling and Quit) have corresponding AppleScript commands. Documentation of the AppleScript commands is available by examining the scripting dictionary for ST using the AppleScript Script Editor (the

Script Editor is part of the standard AppleScript software). To view the dictionary for ST select the Open Dictionary item from the File menu of the Script Editor. Select the ST application icon in the standard file dialog that is displayed to open the ST scripting dictionary. The entry from the dictionary concerning the **compute** command is given below.

**compute** file specification -- *File containing input data*  
          [**output file** file specification] -- *File output is written to*  
          [Result: string] -- *Text containing scaling results*

The direct parameter of the **compute** command is a specification of the file containing the input data. There is an optional **output file** parameter that can contain a specification for an output file. If the **output file** parameter is not present the text containing the scaling results is returned by the compute command, otherwise an alias of the file containing the output text is returned. The following is an example of an AppleScript script for ST using the **compute** command.

```
set infile to alias "Hard Disk:IRT scaling:Scale input"
set outfile to a reference to file "Hard Disk:IRT scaling:Scale out"
tell application "ST"
    compute infile output file outfile
end tell
```

### Issues in Using the Program

ST requires Macintosh system software version 7.0 or later. There are two versions of ST - a version that will run on any Macintosh (a "fat" version) and a version for 680X0 Macintoshes with a math coprocessor (an FPU version) . The fat version runs on both Macintoshes with 680X0 processors and Macintoshes with PowerPC processors (the program runs native on Power Macintoshes). The FPU version will only run on Macintoshes with 680X0 processors and a math coprocessor. The FPU version of the program will run considerably faster than the fat version on 680X0 Macintoshes that have a math coprocessor.

There are no limits on the number of items parameters or theta values read, other than the amount of memory allocated to the program. The memory partition might need to be increased to read a large number of theta values. To change the memory partition select the ST application in the Finder and choose "Get Info" from the File menu. Editing the text next to "Current size" will change the amount of memory the program uses.

The program will report an error if the input file is not in the correct format. Things to check for if the program reports an error in reading the input data are: 1) both new form and old form

item parameters must be present, 2) the first character on an information line must be a dollar sign (there cannot be spaces before the dollar sign), 3) the text identifying the information to be read on an information line is case sensitive (it must match exactly), 4) the strings identifying the items can not be longer than 8 characters, 5) Each old form item must have a corresponding new form item with the same item identification string.

## References

- Kolen, M. J., & Brennan, R. L. (1995). *Test Equating methods and practices*. New York: Springer-Verlag.
- Haebara, T. (1980). Equating logistic ability scales by a weighted least square method. *Japanese Psychological Research*, 22, 144-149.
- Lord, F. M. (1980). *Application of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992) *Numerical recipes in C* (2nd edition). Cambridge University Press.
- Stocking, M. L., & Lord, F. M. (1983). Developing a common metric in item response theory. *Applied Psychological Measurement*, 7, 201-210.
- Zeng, L. & Kolen, M. J. (in preparation). *Computing IRT scale transformation function using numerical integration*.