Developing Data Systems to Support the Analysis and Development of Large-Scale, On-line Assessment

Chong Ho Yu, Ph.D., MCSE, CNE

Paper presented at the annual meeting of the American Educational Research Association, 2001, Seattle, WA

Abstract

Today many data warehousing systems are data-rich but information-poor. Extracting useful information from an ocean of data to support administrative, policy, and instructional decisions becomes a major challenge to both database designers and measurement specialists. The present paper focuses on the development of a data processing system that integrates multiple types of analyses to efficiently support the maintenance and further development of the large-scale, on-line assessment system. We developed an automation package that involves three components of data processing using a Perl script as the master program: 1.Data parsing: Perl. 2.Data analysis: Iteman, Winsteps, Bilog/Biolog MG, Lisrel, SAS. 3. Data presentation: Xlisp-Stat/DataDesk, Webpage generation. With this online and automated systems, content experts could examine the psychometric attributes of test items in a more efficient manner.

Developing Data Systems to Support the Analysis and Development of Large-Scale, On-line Assessment

Chong Ho Yu, Ph.D., MCSE, CNE

Today many data warehousing systems are data-rich but information-poor (Chen & Frolick, 2000). Extracting useful information from an ocean of data to support administrative, policy, and instructional decisions becomes a major challenge to both database designers and measurement specialists. Further, many data warehousing systems adopt a distributed model, in which storage, analytical tools, and clients scatter in different platforms and locations. A unified interface for accessing data in various forms is also a crucial issue. The Cisco Networking Academy Program (CNAP) is facing the challenge of streamlining a system that presently involves various types of data processing and reporting to inform a variety of stakeholders responsible for various parts of the CNAP system. Everyday thousands of test records are submitted to the data warehousing system. The data must be processed to meet many needs in the CNAP system, including psychometric analysis, instructor feedback, student feedback, and administrative summaries. Presently, accomplishing this requires importing and exporting large amounts of data back and forth across five to six different servers and clients, involving a variety of platforms.

The present paper focuses on the development of a data processing system that integrates multiple types of analyses to efficiently support the maintenance and further development of the large-scale, on-line CNAP system. From a psychometric perspective, there is an urgent need to develop a single interface that provides a variety of information to support refinement of test items based upon tester input. Although various item analysis software packages for both true score theory and item response theory are available, the lack of automation and integration of these tools hinders CNAP test developers from evaluating test items in an efficient manner. To rectify this situation, we are developing an integrated procedure to automate the item analysis process. All final results are uploaded to a Web server so that the psychometric analysts and subject matter experts could access them in one location through the Web browser interface.

We developed an automation package that involves three components of data processing using a Perl script as the master program:

1. Data parsing: Perl
2. Data analysis: Iteman, Winsteps, Bilog/Biolog MG, Lisrel, SAS
3. Data presentation: Xlisp-Stat/DataDesk, Webpage generation

Each component will be explained in detail.

Data parsing

Data parsing is a process of extracting useful data based upon pattern matching. Perl (Wall, Christiansen, & Orwant, 2000) is an interpreted high-level programming language, which has powerful pattern matching capabilities for data parsing. Perl script can be used to read a raw data file and output the cleaned data and the key (correct answer). Since different examinations have different numbers of items, the Perl script is able to determine the appropriate number of items for subsequent analyses. Given certain pattern matching conditions, the Perl script is able to extract the needed data only and reformat them so they are ready for analysis in various statistical packages.

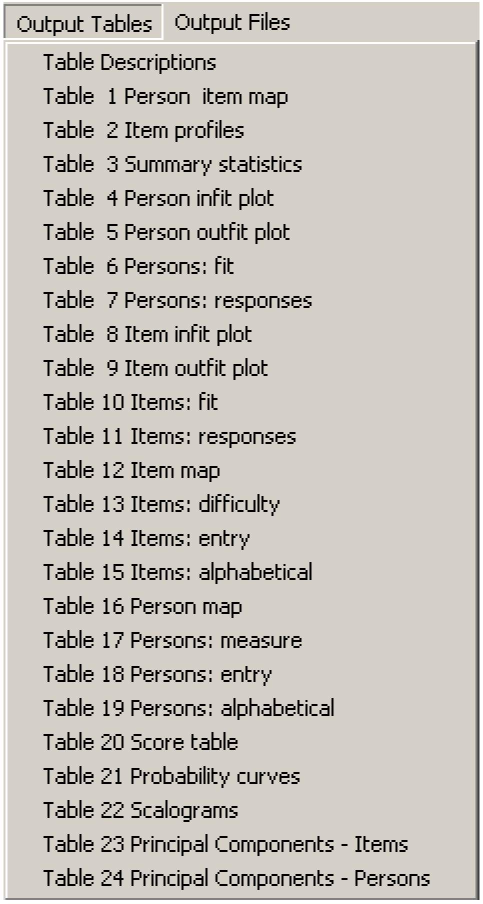
More importantly, Perl works with third-party databases like Oracle, Sybase, Postgres, MySQL, and many others through the abstract database interface called DBI. Currently, due to security reasons, the raw data are exported from the Oracle database server at Cisco in the ASCII format, rather than being accessed from Perl via DBI directly. Nevertheless, Perl is capable of accessing various types of data warehousing systems seamlessly.

Data Analysis

After the data file is cleaned, the Perl script runs Iteman (Assessment Systems Corporation, 2000) to analyze the data. Basically, Iteman produces the classical item analysis results such as the mean, variance, standard deviation, skew, and kurtosis of total (number-correct/keyed) scores, the minimum and maximum score, and median score. It also provides a score frequency distribution as well as a KR-20 estimate of reliability and standard error of measurement for each subtest scale. Although Bilog is capable of providing classical item analysis, Iteman is still indispensable due to several unique features. For example, Iteman can be configured to group examinees based on their overall scores (upper and lower 27%), report the subgroup endorsement rates, and provide a classical upper/lower index of discrimination to identify poorly written items.

Next, the Perl script runs Winsteps (Linacre & Wright, 1998) in a batch mode and output various tables. Winsteps is a program designed for Rasch scaling (see Andrich, 1988), which is equivalent to the one-parameter IRT model. However, the developers of Winsteps assert that equating Rasch scaling to item response theory or logit-linear models is a misclassification. Item response theory and logit-linear models describe data, but Rasch scaling specifies how persons, probes, prompts, raters, test items, and tasks must interact statistically for linear measures to be constructed from ordinal observations (Rasch measurement software and publications, 2001). Discussion of the difference of Rasch scaling and IRT is beyond the scope of this paper.

Certain output tables of Winsteps are very helpful (see Figure 1). For example, the map of students and items illustrates the tester ability and the item difficulty side by side. Using this table, the test developer gains a descriptive picture of the test at one glance. In addition, Winsteps could output the item statistics, the subject statistics, and the residual statistics. The item statistics could be used to determine which items are poorly written. The subject statistics reports the estimated ability (theta) of each tester. The residual statistics indicates the degree of fit between the model and the data.

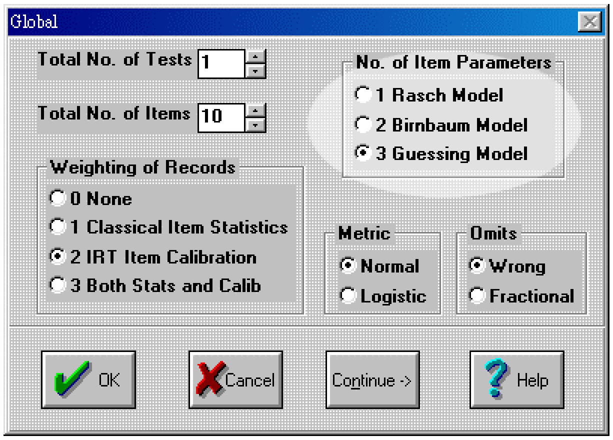


Output Tables | Output Files

Table Descriptions
Table 1 Person item map
Table 2 Item profiles
Table 3 Summary statistics
Table 4 Person infit plot
Table 5 Person outfit plot
Table 6 Persons: fit
Table 7 Persons: responses
Table 8 Item infit plot
Table 9 Item outfit plot
Table 10 Items: fit
Table 11 Items: responses
Table 12 Item map
Table 13 Items: difficulty
Table 14 Items: entry
Table 15 Items: alphabetical
Table 16 Person map
Table 17 Persons: measure
Table 18 Persons: entry
Table 19 Persons: alphabetical
Table 20 Score table
Table 21 Probability curves
Table 22 Scalograms
Table 23 Principal Components - Items
Table 24 Principal Components - Persons

Figure 1. Output tables from Winsteps

Unlike Winsteps, Bilog (Mislevy & Bock , 1990) is capable of running one-, two-, and three-parameter models. The Perl script runs Bilog with all three models in the batch mode, and then extracts the phase 1, phase 2, and phase 3 data for each parameter from the Bilog output (see Figure 2).



Global

Total No. of Tests 1
Total No. of Items 10

Weighting of Records
○ 0 None
○ 1 Classical Item Statistics
● 2 IRT Item Calibration
○ 3 Both Stats and Calib

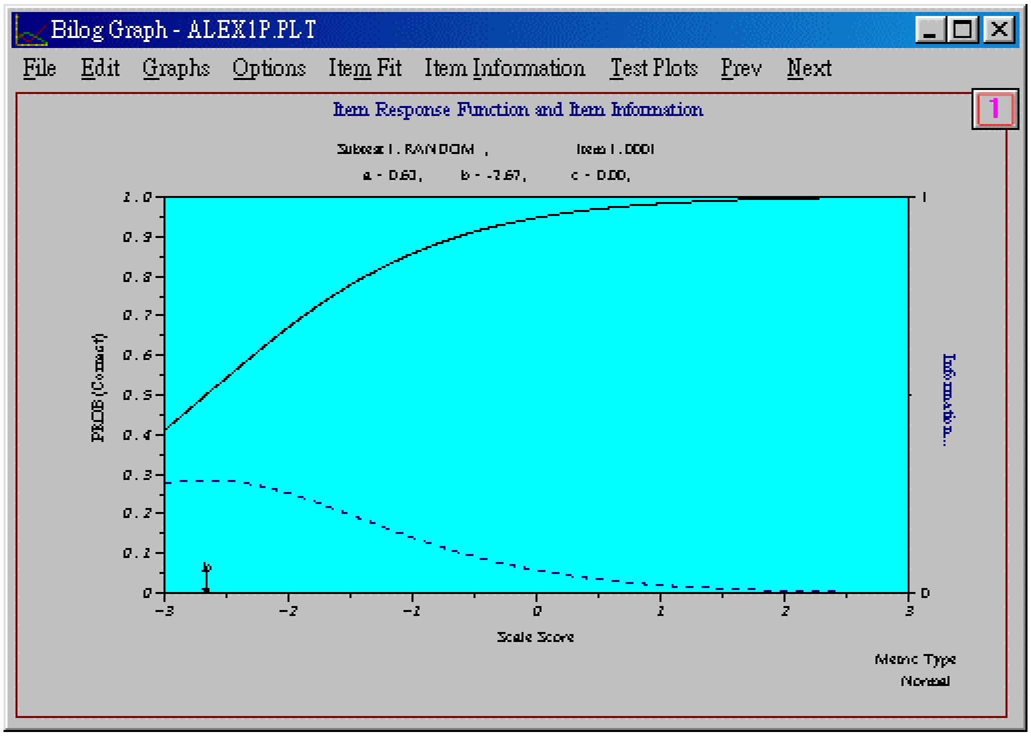
No. of Item Parameters
○ 1 Rasch Model
○ 2 Birnbaum Model
● 3 Guessing Model

Metric
● Normal
○ Logistic

Omits
● Wrong
○ Fractional

Figure 2. 1-3 parameter models in Bilog.

Phase 1 output includes the classical item analysis data such as item difficulty in terms of percentage of correct responses, logits, Pearson coefficients, and bi-serial coefficients. Phase 2 output includes item characteristic curve (ICC)' s parameters such as the low asymptote, the slope, the threshold, and the chi-square statistics. Phase 3 output includes the estimated theta (ability) of each examinee. Although Bilog could display each ICC item by item, it does not overlay the ICCs of all items in one graph (see Figure 3). This shortcoming is remediated by employing Xlisp-Stat/DataDesk in a later step.



Subtest 1 : RANDOM , Item 1 : 0001

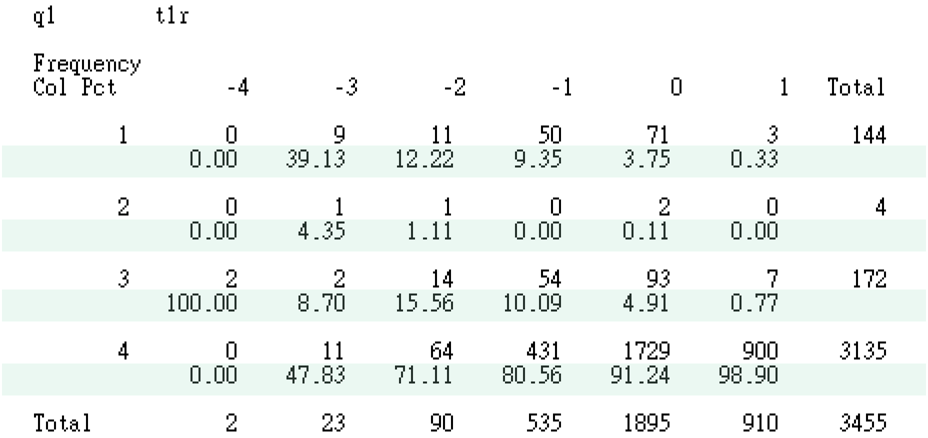
a = 0.63, b = -2.67, c = 0.00,

Figure 3. ICC from Bilog

When field test items are inserted into multiple forms, the Perl script runs Bilog-Multiple Group (MG) (Zimowski, Muraki, Mislevy, & Bock, 1996) to analyze the data. Field test items are new items and thus their psychometric properties are unknown. These items are inserted into real examinations but scores of these items are not counted toward the final grade of testers. Since only a small number of field test items could be presented along with non-field-test items, multiple forms are needed to accommodate testing of these new items. For example, when forty new items are released, four alternate forms are used and ten field test items are included in each form. Field test items are analyzed as a subset and reliability in terms of alternate forms is estimated. Initially we experimented with Bilog for multiple-form analysis. However, it is difficult for Bilog to accept multiple keys and thus Bilog MG was adopted for this specific task.

Item response theory assumes all test items measure a single trait. Therefore, it is important to examine whether the test is unidimensional or multidimensional. For this purpose, the Perl script runs Lisrel (Joreskog & Sorbom, 1993) to perform a factor analysis with the tetrachoric correlation matrix. Since the data are dichotomous, the tetrachoric correlation matrix, instead of the Pearson correlation matrix, is used for factor analysis. It is arguable whether there is a significant difference between running factor analysis with the tetrachoric matrix and performing factor analysis with Pearson matrix. Using CNAP test data, we found that fewer numbers of factors are extracted when analyzing the tetrachoric correlation matrix. Running factor analysis with the tetrachoric matrix requires two steps. First, the Perl script runs a SAS macro program to generate the matix. Although PROC FREQ in SAS can produce the tetrachoric correlation coefficient in a pairwise manner, the SAS macro named “polychoric” (SAS Institute, 2000) is a more efficient approach since it outputs the entire matrix (all possible pairs). Next, this matrix is imported into Lisrel for factor analysis. One may wonder why Lisrel instead of SAS is used when PROC FACTOR in SAS can also factor analyze the tetrachoric matrix. When the number of test items is less than twenty-six, SAS has no problem running factor analysis with the tetrachoric matrix. However, when the number of test items increases, SAS has difficulties.

In addition to running SAS for outputting the tetrachoric matrix, the Perl script runs another SAS program in the batch mode with phase 1, phase 2, and phase 3 data for each parameter, computes the curve height for each ICC conditioning upon the estimated theta, and writes all data to a space-delimited text file, as well as generating frequency tables of distracter X ability (Figure 4). The frequency tables are generated by Output Delivery System (ODS) in SAS so that the tables are HTML ready. The information regarding the portion of each option selected by testers of different levels of ability could help the test developer find out which distracter is confusing to even testers of high ability, and which distracter is so unconvincing that even examinees of very low ability did not give it consideration, as well as other anomalies (van der Linden & Hambleton, 1990). It is important to note that the row percentage, which is the portion of choosing a particular option conditional on the theta (highlighted in green), rather than the frequency count, is the focal interest of the analysis.



```
q1        t1r

Frequency
Col Pct        -4       -3       -2       -1        0        1    Total

       1        0        9       11       50       71        3      144
             0.00    39.13    12.22     9.35     3.75     0.33

       2        0        1        1        0        2        0        4
             0.00     4.35     1.11     0.00     0.11     0.00

       3        2        2       14       54       93        7      172
           100.00     8.70    15.56    10.09     4.91     0.77

       4        0       11       64      431     1729      900     3135
             0.00    47.83    71.11    80.56    91.24    98.90

Total           2       23       90      535     1895      910     3455
```

Figure 4. Distracter X ability table

Data presentation

Next, the Perl script runs an Xlisp-Stat program to import the space-delimited file formatted by the SAS program and generate ICCs and fit statistics for 1-P, 2-P, and 3-P models. Currently this procedure is still under development. As a temporary solution, DataDesk is used to generate graphics. Since the graphic panels have the capability of linking and brushing, the test developer could examine the ICC and the fit statistics (chi-square/degree of freedom) in an interactive manner (see Figures 5). For example, in the one-parameter ICC plot shown in Figure 5, one test item displays a relatively large fit statistic. The subject expert could click on the data point on the left panel to highlight the ICC of that item on the right panel to assess how likely it would be for examinees of different thetas to get the item correct.

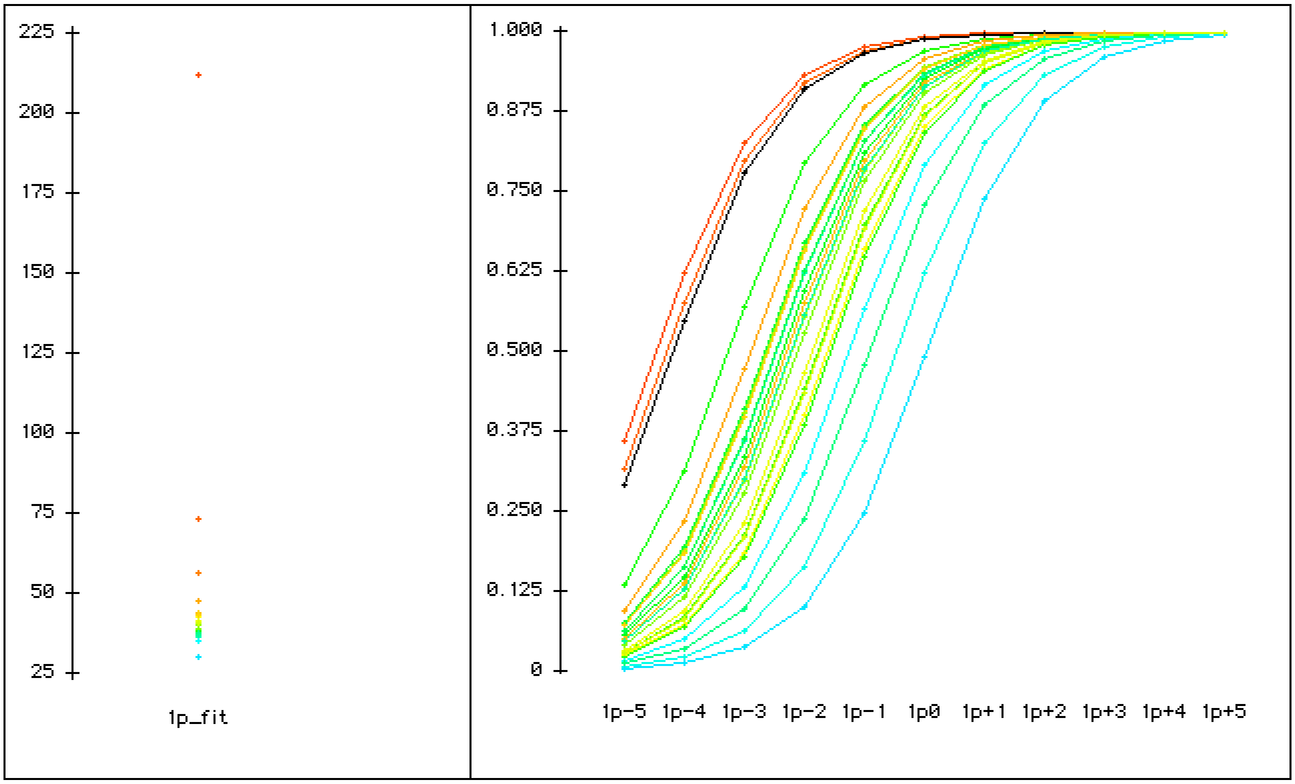
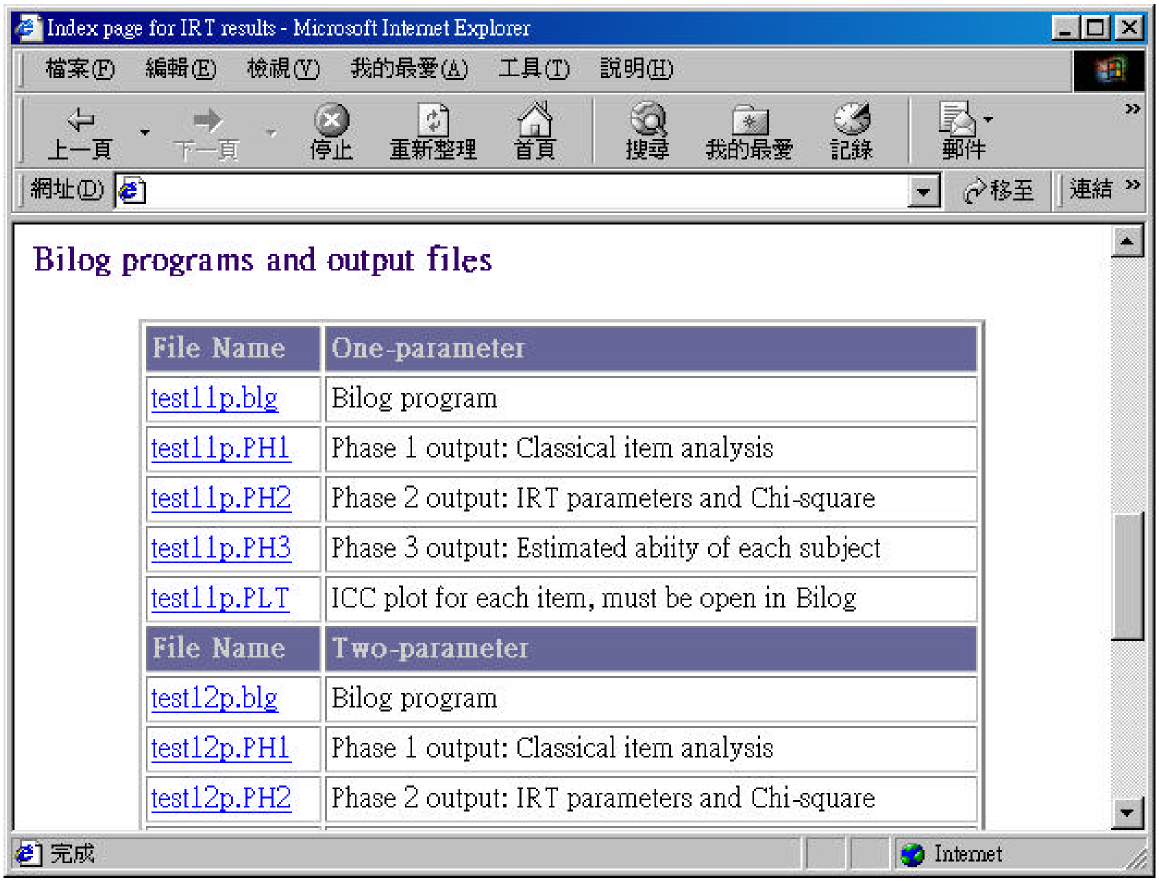


Figure 5. Fit statistics and ICCs for the one-parameter model.

Finally, the Perl script generates an html file as a front end interface for the subject experts to view or download different output files via the internet (see Figure 6). Since Perl is capable of interacting with Web server' s common gateway interface (CGI), the Perl script could authenticate Web users and response to user query with certain search criteria. This provides a user-friendly method for a variety of users to access summaries of psychometric information.



Bilog programs and output files

File Name	One-parameter
test11p.blg	Bilog program
test11p.PH1	Phase 1 output: Classical item analysis
test11p.PH2	Phase 2 output: IRT parameters and Chi-square
test11p.PH3	Phase 3 output: Estimated abiity of each subject
test11p.PLT	ICC plot for each item, must be open in Bilog
File Name	Two-parameter
test12p.blg	Bilog program
test12p.PH1	Phase 1 output: Classical item analysis
test12p.PH2	Phase 2 output: IRT parameters and Chi-square

Figure 6. User-friendly HTML front end for data presentation

Discussion

Extracting meaningful information from data and presenting the information in a unified interface is a serious challenge for both database programmers and test developers. Besides functionality, flexibility should also be considered in developing data processing systems. This module is highly flexible. Like a vehicle consisting of inter-changeable parts, this module involves multiple programs that could be swapped as needed. For example, when we found that SAS is incapable of performing factor analysis with a tetrachoric matrix of more than 25 items, a Lisrel program was implemented for this task. When Bilog failed to accept multiple keys for multiple-form analysis, Bilog-MG was introduced into the module. In addition, the Perl script is interpreted rather than compiled. The source code can be seen by users, and thus different users could modify the source code to accommodate different needs such as parsing data for a different format, or skipping steps such as Bilog MG analyses when multiple forms are not needed. This flexible and comprehensive test analysis module will be helpful to test developers. Our next step is to collect feedback from users (subject matter experts) for further enhancement of the module.

Acknowledgement

We would like to thank Vikas Wadhwani for his assistance with Perl programming, Cody Ding for his help with Bilog MG programming, and Marilyn Thompson for her help with Lisrel analyses.

References

Andrich, D. (1988). Rasch models for measurement. Newbury Park: Sage Publications.

Assessment Systems Corporation (2000). Iteman [Computer software]. [On-line] Available: URL: http://www.assess.com/iteman.html

Chen, L., & Frolick, M. N. (2000). Web-baded data warehousing. Information Systems Management. 17, 80-87.

Mislevy, R., Bock, R. D. (1990). Bilog: Item analysis and test scoring with binary logistic models [Computer software]. Mooresville, IN: Scientific Software.

Joreskog, K. G., & Sorbom, D. (1993). Lisrel 8 [Computer software]. Chicago, IL: Scientific Software International.

Linacre, J. M., & Wright, B. D. (1998). A user's guide to Bigsteps/Winsteps. Chicago, IL: Mesa Press,.

Rasch measurement software and publications. (2001). Rasch measurement, tools, techniques, and people. [On-line] Available URL: http://www.winsteps.com/

Ryan, J. (1983). Introduction to latent trait analysis and item response theory. In W. E. Hathaway (Ed.). Testing in the schools: new directions for testing and measurement, no. 19 (pp. 49-65). San Francisco, CA: Jossey-Bass.

SAS Institure. (2000). Polychoric macro [Computer software]. [On-line] Available: URL: http://www.sas.com

van der Linden, W. J., & Hambleton, R. K. (Eds.). (1997). Handbook of modern item response theory. New York: Springer.

Wall, L., Christiansen, T., & Orwant, J. (2000). Programming Perl (3rd Ed.). New York: O'Reilly.

Zimowski, M. F., uraki, E., Mislevy, R. J., & Bock, R. D. (1996). Bilog-MG: Multiple group IRT analysis and test maintenance for binary items [Computer software]. Chicago, IL: Scientific Software.