Predicting Item Exposure Parameters in Computerized Adaptive Testing

By
Shu-Ying Chen
Department of Psychology
National Chung-Cheng University

Shing-Hwang Doong
Department of Information Management
Shu-Te University

This study was supported in part by the National Science Council, Taiwan, R.O.C. (NSC 91-2413-H-194-031) and presented at the 2003 Annual Meeting of the American Educational Research Association, Chicago, IL. USA.

Predicting Item Exposure Parameters in Computerized Adaptive Testing

Abstract

The purpose of this study is to find a relationship formula between item exposure parameters and item parameters in computerized adaptive tests (CATs) by using genetic programming (GP), which is a biologically inspired artificial intelligence technique. Based on the relationship formula, item exposure parameters for new parallel item pools can be predicted without conducting any tedious iterative simulations.

Randomized item selection (e.g., McBride & Martin, 1983) and conditioned item selection (e.g., Sympson and Hetter, 1985) are main approaches used to avoid item overexposure in CATs. Although the conditioned item selection performs better than the randomized item selection on controlling item exposure, finding stabilized item exposure parameters through iterative simulation for the conditioned item selection is very time consuming. Furthermore, the tedious iterative simulations need to be re-conducted whenever there are changes in CAT settings or the examinee population of interest. Since item pools are changed much more often than the other CAT setting when CATs are administered in practice, it would be very useful if the relationship between item exposure parameters and item parameters in a pool can be built. Then, based on the resultant relationship, item exposure parameters for new parallel item pools can be obtained easily without conducting any tedious iterative simulations. This study attempted to find the relationship formula by using GP, which has been shown to be able to rediscover many interesting formulae found in economics or sciences.

Based on the results observed from the study, the relationship formula between item exposure parameters and item parameters in a pool could be built by using GP. The error of prediction was around 0.12 for each item in the pool when item exposure parameters were predicted by using the found formula. The accuracy of the prediction was maintained across parallel item pools. Thus, item exposure parameters could be predicted with moderate errors by using the GP techniques without conducting any tedious iterative simulations.

Keywords: computerized adaptive testing, item exposure control, test security, item exposure parameter, genetic programming.

Predicting Item Exposure Parameters in Computerized Adaptive Testing

One practical advantage of computerized adaptive tests (CATs) is that they can be administered on a flexible schedule rather than at fixed times. The convenience and flexibility for examinees, however, may severely compromise test security if item exposure is not well controlled. To date, randomized item selection (e.g., McBride & Martin, 1983) and conditioned item selection (e.g., Sympson and Hetter, 1985) are main approaches used to avoid item overexposure in CATs (Way, 1998). Although the randomized item selection is simple and easy to implement in CATs, this approach doesn't control item exposure well but shuffle items (Sympsom & Hetter, 1985). On the other hand, the conditioned item selection can control item exposure well such that most items are administered with item exposure rates less than a pre-specified maximum item exposure rate ($r_{\max}$). The conditioned item selection is described in details as follows.

Conditioned Item Selection

The conditioned item selection procedure originally proposed by Sympson and Hetter (1985) was designed to directly control item exposure in a probabilistic fashion, such that most items are administered with frequency less than a pre-specified maximum item exposure rate ($r_{\max}$). Three kinds of probability are defined in this approach: P(S), the probability that an item is "selected" as the best item based on a CAT algorithm; P(A), the probability that an item is actually "administered" to examinees; and P(A|S), the conditional probability that an item is administered, given that it is selected as the best item, also named "item exposure parameter." Since an item must be selected first before it can be administered, the relationship among the three probabilities can be described in terms of a formula,

P(A)=P(A|S)×P(S). In order to meet the requirement that no item has been administered more often than r_{max}, the condition thus becomes P(A)=P(A|S)×P(S) ≤ r_{max} .

The purpose of P(A|S) is to adjust P(S) such that P(A) can be less than or equal to r_{max}. For example, when P(S) is greater than r_{max}, P(A|S) should be assigned a value not exceeding r_{max}/P(S) so that P(A) ≤ r_{max}. On the other hand, when P(S) is less than r_{max}, no adjustment is needed, so P(A|S) would be assigned the value of 1. In other words, when an item is selected more frequently, a smaller P(A|S) is needed to protect the item against overadministration. On the other hand, when an item is selected less frequently, no protection against overadministration is needed and the value of P(A|S) can be as large as 1. Thus, P(A|S) ranges from r_{max} to 1, and gives evidence of an item's popularity, where a value at r_{max} indicates that the item is selected for all examinees whereas a value of 1 indicates that the item is selected for less than r_{max} of examinees.

Even though it is simple to decide P(A|S) as long as P(S) is known, determining P(S) is not trivial. Furthermore, when P(A|S) for an item is decided, the P(S) values for the rest of the items in the pool would be changed, as would the P(A|S) values associated with them. Thus, a series of iterative simulations is needed to find stabilized P(S) values and P(A|S) values under which all P(A) values are less than or equal to r_{max}. The procedure for conducting the series of iterative simulations is described in the following paragraphs.

Step 1. Specify the environment for conducting the iterative simulations—the design of a CAT (item pool, item selection criterion, starting rule, termination rule, method of trait estimation, etc.), the distribution of θ (trait) in the population of interest, the desired maximum item exposure rate (r_{max}), and assume the initial P(A|S) is 1 for all items in the item pool (P(A|S) will be denoted henceforth as *k*).

Step 2. Administer CATs to more than 1,000 simulees drawn from the specified population. During the testing course for an examinee, if Item *i* is "selected" given a trait estimate, a random number (*x*) generated from a uniform distribution on the interval (0,1) is compared with the k_i. Item *i* is "administered" if $x \le k_i$. If Item *i* is not administered, select the next best item, Item *j*, and compare another random number with k_j to decide if Item *j* can be administered based on the same rule. The procedure is repeated until an item is administered. Regardless of whether the selected items are administered, remove them from the item pool for the remainder of this examinee's test.

Step 3. Find P(S) and P(A) for each item by computing the proportion of times an item has been selected and administered, respectively. Record the maximum value of P(A) in the item pool and redefine *k* for each item based on P(S) as follows:

If P(S) > $r_{\max}$, then $k = r_{\max}$/P(S); or

If P(S) ≤ $r_{\max}$, then $k = 1.0$.

To guarantee that an examinee will get a complete test before exhausting the item pool, there should be at least *n* items with $k = 1.0$, where *n* is the test length. Thus, setting the *n* largest *k*s equal to 1.0 is necessary after redefining *k*s.

Step 4. Given the redefined *k*s, go back to Step 2. Repeat the iterative simulations until *k*s are stabilized and the maximum value of P(A) is close to $r_{\max}$ and then oscillates in successive simulations.

By using the final stabilized *k*s in the real CATs, most items can be administered with item exposure rates less than the pre-specified $r_{\max}$. Thus, item exposure is well controlled under the approach of conditioned item selection.

Even though the Sympson and Hetter (1985) method seems to perform better than that of McBride and Martin (1983), the iterative simulations required to find stabilized item

exposure parameters (*k*s) are very time consuming (Stocking & Lewis, 1995). Furthermore, the tedious iterative simulations need to be re-conducted whenever there are changes in CAT settings (e.g., item response model, item pool, item selection criterion, starting rule, termination rule, method of trait estimation) or the examinee population of interest. For example, when an item is added to or deleted from an item pool, new iterative simulations are needed; *k*s would be different.

Among the CAT settings, due to the considerations of test security, item pools are changed much more often than the other CAT settings when CATs are administered in practice. Thus, frequently, iterative simulations have to be conducted to find item exposure parameters for new parallel item pools in real CATs. It would be very useful if the relationship between item exposure parameters and item parameters in a pool can be built. Then, based on the resultant relationship, item exposure parameters for new parallel item pools can be obtained easily without conducting any tedious iterative simulations.

Finding the relationship between item exposure parameters and item parameters is not trivial. While it seems difficult to find the relationship analytically, genetic programming, which was originally devised to have computers generate problem solving computer codes automatically in a genetic fashion, may provide a way to solve this problem computationally. Following is a description about the genetic programming.

Genetic Programming (GP)

The problem of finding relationship between predictor variables and response variables is usually solved by a regression approach in statistics. However, in addition to rigid assumptions, most regression techniques require users to specify the size and shape of the model (e.g., linear regression assumes the relationship between independent variables and dependent variables is linear). In many cases, the most important issue in regression is to

determine the size and shape of the model itself (Koza, 1990).

Genetic programming, a biologically inspired artificial intelligence technique, may help in finding the size and shape of the model (Koza, 1990; Duffy&Engle-Warnick, 1999). In 1975, Holland devised an ingenious algorithm called genetic algorithm (GA) to solve a class of optimization problem. GA is based on Darwinian principle of evolution to provide a probabilistic search method in the solution space. It works on a population of individuals (chromosomes) instead of a single one. Each chromosome is composed of genes, which are simply the parameters used in the target problem. For example, suppose we want to optimize the function *f(x,y,z)*, then we can assume a chromosome of the type [*x,y,z*], a 3-element array (3 genes per chromosome) with the desired data type for each element. Commonly used data types are binary (0 or 1), characters ('A', 'B', 'a', …), integral and real numbers. Each chromosome will be assigned a fitness value as the survival arbitrator in the course of evolution. Usually this value is closely related to the target objective function. Based on the fitness values, the population of individuals is evolved into the next generation according to the following principles:

1. Selection – provided by the fitness proportionate reproduction. Selection operation matches Darwinian "survival of the fittest" principle. A chromosome with a higher fitness value will have a higher probability to survive in the next generation.
2. Crossover- provided by recombination of parts of the parental chromosomes. Each offspring from this operation will have part trait from the father chromosome and part trait from the mother chromosome.
3. Mutation – provided by randomly changing some part of a chromosome. In order to preserve the diversity of individuals in the population, mutation operation will be performed in some probabilistic fashion.

The three genetic operations (Selection, Crossover, and Mutation) tend to promote the elite individuals while preserving the diversity of the individuals at the same time during the course of the search procedure. If we always preserve the best individual in each generation, it can be shown that the best individual in GA will converge to a global optimal solution of the original problem (Rudolph, 1994). GA has been successfully applied in many fields to show its domain-independent characteristics of solving optimization problems (Goldberg, 1989; Holland, 1975).

Based on the very successful experience of GA's application of genetic principles in AI, John Koza of Stanford University proposed a new paradigm named genetic programming (GP) on the same genetic principles to produce good computer codes automatically for solving a problem. GP also works with a population of individuals and evolves them in generations via the same Selection, Crossover and Mutation operations used in GA. However, each individual in GP is a computer program instead of a fixed-length array type chromosome as used in GA. In order to evaluate the fitness of an individual program in GP, we need to set up an environment for the computer program to run and measure its fitness in this environment. There are three steps in a GP design (Koza, Bennett, Andre & Keane, 1999):

1. Randomly create an initial population of individual computer programs
2. Iteratively perform the following sub-steps (called a generation) on the population of programs until the termination criterion has been satisfied:
    (1) Assign a fitness value to each individual program
    (2) Create a new population of individual programs by applying the three genetic operations "Selection, Crossover, Mutation"
3. Designate the individual program as the result solution.

GP has been shown to be able to rediscover many interesting formulae found in economics or sciences (Koza, 1990; Koza, 1992; Duffy&Engle-Warnick, 1999). For

example, in Koza, 1990, a genetic programming paradigm was used to rediscover a well-known multiplicative (non-linear) exchange equation $M=PQ/V$ relating the money supply (M), price level (P), gross national product (Q) and velocity (V) of money in an economy. Following the same methodology, GP may also be used to find a relationship formula between item exposure parameters and item parameters in CATs.

Purpose of the Study

The purpose of this study was to find the relationship formula between item exposure parameters and item parameters for 3P logistic models at various trait levels from -3 to 3 by using genetic programming methodology. The found relationship can then be used to predict item exposure parameters for new parallel item pools without conducting any tedious iterative simulations.

Method

A detailed description of the processes to find the item exposure parameters and to discover a relationship formula between the item exposure parameters and item parameters for 3P logistic model at trait level 3 is given below. Similar processes are applied to the other trait levels.

Finding item exposure parameters

Step 1. The environment for conducting the iterative simulations is specified as follows.

Item response model. All simulated data were generated using the three-parameter logistic item response model (3PLM), in which the probability of a correct response, given a trait level (θ), is defined by

$$P_i(\theta) = c_i + \frac{1 - c_i}{1 + e^{-1.7a_i(\theta - b_i)}}, \quad (1)$$

where a_i is the item discrimination parameter, b_i is the item difficulty parameter, and c_i is the pseudo guessing parameter.

Item pool. Three parallel item pools were used in this study with 360 items in each pool. Iterative simulations were conducted to find item exposure parameters for each pool. One of the item pools, named Training was used to discover the relationship between item exposure parameters and item parameters. The other two pools, named Test 1 and Test 2 were used to conduct cross validation.

Initialization. The initial trait estimate was assumed to be zero.

Trait estimation. Expected a posteriori (EAP) estimation with a θ~N(0,1) prior distribution was used for the current study.

Test length. The test length was set at 20 items.

Item selection. The maximization of item information was the criterion used for item selection at each stage.

Regarding the item exposure control, the Sympson and Hetter (1985) procedure was adopted with the maximum item exposure rate set at 0.2. Assuming a true trait level of 3, one thousand simulees were used in each iterative simulation to find stabilized item exposure parameters. An initial value of 1.0 for the item exposure parameter (*k*) for all items in the item pool was assumed.

Step 2. Simulated CATs were administered to 1,000 simulees with $\theta = 3$. Given an EAP trait estimate based on *n* items having been administered so far ($\hat{\theta}_n$), the $(n + 1)^{th}$ item were selected such that $I_j\left(\hat{\theta}_n\right)$ (item information) had the maximum value among all of the items. Given the "selected" item *j*, one random number (*x*) generated from a uniform distribution on the interval (0,1) was compared with k_j. Item *j* was "administered" if

$x \le k_j$. If item *j* was not administered, the next best item *g* was selected. For item *g* another random number was compared with k_g to decide if item *g* could be administered based on the same rule. The procedure was repeated until an item was administered. Regardless of whether the selected items were administered, they were removed from the item pool for the remainder of this examinee's test.

Step 3. P(S) and P(A) were determined for each item by computing the proportion of times an item has been selected and administered, respectively. The maximum value of P(A) in the item pool was noted and *k* was redefined for each item based on P(S) as follows:

If P(S) > 0.2, then k =0.2/P(S); or

If P(S) ≤ 0.2, then k= 1.0.

To guarantee that an examinee would take a complete test before exhausting the item pool, the 20 largest item exposure parameters were set to 1.0.

Step 4. Given the redefined *k*s, returned to Step 2. The iterative simulations were repeated until *k*s stabilized and the maximum value of P(A) was slightly above 0.2 and oscillated in successive simulations.

Finding a relationship formula between item exposure parameters and item parameters

Given 360 pairs of data (a_i, b_i, c_i, k_i), i = 1, 360 obtained from the item parameters and item exposure parameters described above, GP was used to find a relationship formula, where item parameters were independent variables and item exposure parameter was dependent variable. Following Koza's approach (1990), a function set, a terminal set and an environment in the GP paradigm were defined as follow.

1. Function set: Arithmetic operators (+, -, *, /), Exp, Log, and Sqrt were used here. Elements from the function set corresponded to the non-leaf nodes of the parsing tree of an individual program in a GP population.

2. Terminal set: Item parameters (*a, b, c*), probability of a correct response, item information and random real numbers (1, -2.3 and etc.) constituted the terminal set. Elements from the terminal set appeared as the leaf nodes in the parsing tree of an individual program in a GP population.

   The selection of these two sets is usually part of the domain knowledge required for solving the target problem. Most common arithmetic operators in the function set, and the item parameters (*a, b, c*), probability of a correct response, item information and real numbers as the terminal set elements have been selected for the current study.

3. Environment: The set of the given 360 data (a_i, b_i, c_i, k_i), i = 1, 360 constituted the environment for measuring the fitness of an individual formula. For each formula composed of terminals from the terminal set and functions from the function set, we used the environment to evaluate the fitness value of the individual formula. For example, suppose

$$(a + c - 0.1) * b / (1 + ac) \qquad (2)$$

   is one of the individual formula. For i = 1, 360, we substitute (a_i, b_i, c_i) successively into formula (2), evaluate its value and the difference between the evaluated value and the corresponding k_i, and finally take the sum of the squares of these differences. This is denoted as S*SE,* the sum of squared error for the individual formula (2) in the environment. Then, a fitness value of formula (2) can be defined as

$$\frac{1}{1 + SSE} \qquad (3)$$

   This fitness value is always between 0 and 1 and has the desired property that the larger a fitness value an individual formula has, the better fit it would be.

A public domain GP program was used for the current study.

Evaluation

To evaluate the accuracy of the item exposure parameters predicted by GP, root mean squared error (RMSE) was calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{360} \left(k_i - \hat{k}_i\right)^2}{360}} \tag{4}$$

where k_i is item exposure parameter obtained from the Sympson and Hetter (1985) procedure and $\hat{k}_i$ is item exposure parameter predicted by GP. To investigate the validation of the relationship formula found by GP, two parallel item pools were used to conduct cross validation.

Results

The relationship formula between item exposure parameters and item parameters found by GP for trait level equal to 3 is as follows.

```
k̂ = Exp( Mul( Mul( Sqrt( Add( Rlog( Sub( Add( Arg3
    :     :     :     :     :     :     :     : Rlog( FConst<57.3779> ) )
    :     :     :     :     :     :     : Arg1 ) )
    :     :     :     :     : Arg1 ) )
    :     :     : Sqrt( Add( Rlog( Sqrt( Mul( Arg1
    :     :     :     :     :     :     :     : Arg3 ) ) )
    :     :     :     :     : Div( Exp( Exp( Arg3 ) )
    :     :     :     :     :     : Arg4 ) ) ) )
    :     : Div( Sub( Sub( Sub( Arg3
    :     :     :     :     :     : Arg4 )
    :     :     :     :     : Arg0 )
    :     :     :     : Arg4 )
    :     :     : Sqrt( Sqrt( Exp( Div( FConst<660.5073>
    :     :     :     :     :     :     : Add( Arg0
```

```
:    :    :    :    :    :    :    : Mul( FConst<582.9944>
:    :    :    :    :    :    :    :    : Arg4 ) ) ) ) ) ) ) ) )
```

where $\hat{k}$ is item exposure parameter predicted by GP; Arg0~Arg4 are independent variables, including item parameters (a, b, c), probability of correct response (p) and item information (f) respectively; FConst is a real number; Add, Sub, Mul, Div, Rlog, Exp, and Sqrt are arithmetic operators, +, -, *, /, log, exponential, and square root respectively.

The formula can be simplified based on the definition of each term. For example, part of the formula can be expressed as follows.

```
Sqrt( Add( Rlog( Sub( Add( Arg3
  :    :    :    :    : Rlog( FConst<57.3779> ) )
  :    :    :    : Arg1 ) )
  :    : Arg1 ) )
```

$$= \sqrt{\log((\log 57.3779 + p) - b) + b}$$

Based on the formula found by GP, item exposure parameter for each item in the three parallel item pools was predicted. Table 1 shows the root mean squared error for three parallel item pools at each true trait level from –3 to 3.

Insert Table 1 about here

For the pool of Training, the amount of RMSE ranged from 0.11 to 0.14 across seven true trait levels. That is, the error of prediction was around 0.12 for each item in the pool of Training. Similar results were observed for the two parallel pools, Test 1 and Test 2. The amount of RMSE ranged from 0.12 to 0.15 for Test 1 and from 0.11 to 0.15 for Test 2. At each true trait level, the differences among the three parallel pools with respect to RMSE were

not distinguishable. In other words, the relationship formulae found by GP were valid across parallel item pools.

Conclusions

CATs cannot be implemented effectively in practice unless item exposure is well controlled. Among the methods used to avoid item overexposure in CATs, the Sympson and Hetter (1985) procedure is a typical one and has commonly used in practice. Although the Sympson and Hetter (1985) procedure performs well on controlling item exposure, finding stabilized item exposure parameters through iterative simulation for the Sympson and Hetter (1985) procedure is very time consuming. Moreover, the tedious iterative simulations need to be re-conducted whenever there are changes in CAT settings or the examinee population of interest. GP was proposed in this study to explore a relationship formula between item exposure parameters and item parameters such that item exposure parameters can be obtained efficiently without conducting any tedious iterative simulations.

The results showed that the relationship formula between item exposure parameters and item parameters in a pool could be built by using GP. The amount of RMSE was around 0.12 when item exposure parameters were predicted by using the found formula. Although the amount of RMSE was not large, the individual error of prediction could be as large as 0.5, which is too big to be acceptable in practice. Thus, the GP approach should be improved such that not only RMSE is small but also no individual error is greater than 0.1 or 0.2 before it can be implemented in practice.

The accuracy of the prediction of item exposure parameters was maintained across parallel item pools. That is, the relationship formula found by GP did function well on predicting item exposure parameters for items in the new parallel item pools. Thus, item

exposure parameters could be predicted with moderate errors by using the GP techniques without conducting any tedious iterative simulations.

GP is a powerful technique from computer science and has been used successfully in many fields. It could be applied in the field of educational measurement to do effective data prediction.

References

Duffy, J., & Engle-Warnick, J. (1999, March). *Using symbolic regression to infer strategies from experimental data.* Society for computational Economics, Computing in Economics and Finance 1999.

Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading.

Holland, J. (1975). *Adaptation in natural and artificial systems.* University of Michigan press, Ann Arbor.

Koza, J. (1990, August). *A genetic approach to econometric modeling.* Sixth World Congress of the Econometric Society, Barcelona, Span.

Koza, J. (1992). *Genetic programming: On the programming of computers by means of natural selection*, MIT press.

Koza, J., Bennett, F., Andre, D., & Keane, M. (1999, April). *Genetic programming: biologically inspired computation that creatively solves non-trivial problems*. Proceedings of the AISB 1999 Symposium on AI and Scientific Creativity, Edinburgh, UK.

McBride, J. R., & Martin, J. T. (1983). Reliability and validity of adaptive ability tests in a military setting. In D. J. Weiss (Ed.), *New horizons in testing* (pp. 223-236). New York, NY: Academic Press.

Rudolph, G.. (1994). Convergence analysis of canonical genetic algorithms, *IEEE transactions on neural networks*, *5*, 96-101.

Stocking, M. L., & Lewis, C. (1995). *A new method for controlling item exposure in computerized adaptive testing* (ETS Research Report RR-95-25). Princeton, NJ: Educational Testing Service.

Stocking, M. L., & Lewis, C. (1998). Controlling item exposure conditional on ability in computerized adaptive testing. *Journal of Educational and Behavior Statistics, 23,* 57-75.

Sympson, J. B., & Hetter, R. D. (1985, October). Controlling item-exposure rates in computerized adaptive testing. *Proceedings of the 27th Annual Meeting of the Military Testing Association* (pp. 973-977). San Diego, CA: Navy Personnel Research and Development Center.

Way, W. D. (1998). Protecting the integrity of computerized testing item pools. *Educational Measurement: Issues and Practice, 17*, 17-27.

Table 1

Root mean squared error for three parallel item pools at each true trait level (θ)

	θ						
Pool	-3	-2	-1	0	1	2	3
Training	0.12	0.12	0.13	0.11	0.11	0.14	0.13
Test 1	0.12	0.13	0.13	0.12	0.12	0.14	0.15
Test 2	0.13	0.13	0.13	0.11	0.12	0.15	0.15