DOCUMENT RESUME

ED 309 197 TM 013 686

AUTHOR van der Linden, Wim J., Ed.
TITLE IRT-Based Test Construction. Project Psychometric Aspects of Item Banking No. 15. Research Report 87-2.
INSTITUTION Twente Univ., Enschede (Netherlands). Dept. of Education.
PUB DATE 87
NOTE 83p.; Portions of these papers were presented at the Annual Meeting of the American Educational Research Association (Washington, DC, April 20-24, 1978).
AVAILABLE FROM Mediatheek, Faculteit Toegepaste Onderwijskunde, Universiteit Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS Algorithms; *Computer Assisted Testing; Decision Making; Foreign Countries; Heuristics; *Item Banks; *Latent Trait Theory; *Mathematical Models; Models; Programing; Selection; *Test Construction; Test Items
IDENTIFIERS *Information Function (Tests); Minimax Programming; Zero One Programing

ABSTRACT

Four discussions of test construction based on item response theory (IRT) are presented. The first discussion, "Test Design as Model Building in Mathematical Programming" (T. J. J. M. Theunissen), presents test design as a decision process under certainty. A natural way of modeling this process leads to mathematical programming. General models of test construction are discussed, with information about algorithms and heuristics; ideas about the analysis and refinement of test constraints are also considered. The second paper, "Methods for Simultaneous Test Construction" (Ellen Boekkooi-Timminga), gives an overview of simultaneous test construction using zero-one programming. The item selection process is based on IRT. Some objective functions and practical constraints are presented, the construction of parallel tests is considered, and two tables are provided. The third paper, "Automated Test Construction Using Minimax Programming" (Wim J. van der Linden), proposes the use of the minimax principle in IRT test construction and indicates how this results in test information functions deviating less systematically from the target function than for the usual criterion of minimal test length. An alternative approach and some practical constraints are considered. The final paper, "A Procedure To Assess Target Information Functions" (Henk Kelderman), discusses the concept of an information function and its properties. An interpretable function of information is chosen: the probability of a wrong order of the ability estimates of two subjects. (SLD)

IRT-based Test Construction

Research Report 87-2

W.J. van der Linden (Ed.)

U.S. Department of Education

Division of Educational Measurement and Data Analysis

University of Twente

Colophon

Typing : A. Burchartz, L. Padberg
Cover design : M. Driessen, AV-section, University of Twente
Printed by : Central Reproduction Department, University of Twente

IRT-based Test Construction

Wim J. van der Linden (Ed.)

Contents

Test design as model building in mathematical programming
T.J.J.M. Theunissen

Some methods for simultaneous test construction
Ellen Boekkooi-Timminga

Automated test construction using minimax programming
Wim J. van der Linden

Some procedures to assess target information functions
Henk Kelderman

Test Design as Model Building in Mathematical Programming

T.J.J.M. Theunissen

Abstract

Test design is presented as a decision process under certainty. a natural way of modeling this process leads to mathematical programming. Several models are presented, including information about algorithms and heuristics. Furthermore, notions about the analysis and refinement of test constraints are briefly presented.

TEST DESIGN AS MODEL BUILDING IN MATHEMATICAL PROGRAMMING

Introduction

Programming in the sence of this paper simply means planning. Luce and Raiffa (1957) categorize mathematical programming as belonging to the area of (individual) decision making under certainty and point at the very close relationship between linear programming (one of many forms of mathematical programming) and two-person zero-sum game theory. In decision making under certainty each of the available options, leads invariably to a specific (certain) outcome. Given such a set of options we should choose one that optimizes some index. The programming problem, as described by Luce and Raiffa in a very general way, consists of (1) options, where each option implies the choice of n real numbers, (2) feasibility conditions, where each condition consists of a (linear) equality or inequality constraining the options and (3) an index associated with each option which is a function of the n numbers. This, in actual fact, is the 'model' as used in mathematical programming. The term therefore does not refer to psychometric theory, but to the structure of a decision process. Several types of decision processes occur time and again under different disguises in a large variety of concrete fields of activity. Closer inspection often reveals a common abstract structure, which turns such a class of problems into an abstract class, whose members have essentially the

same structure. One such class is the class of Packing problems, also known as Knapsack test design problems. Further below, it will be shown that many test design problems are members of this class.

Assuming the availability of a pool of I items, calibrated with some I.R.T. model, the total number of possible test is 2^I, generally an extremely large number. 'Finding' a particular test in this universe of tests is usually impossible on practical grounds. Summarizing, to each itembank belongs a deterministically defined test universe; the problem is how to find the desired test. Generally the test universe is reduced in size by practical considerations as, for example, limited testing time for students implying a maximum number of items to be used, or by common sense considerations, e.g. when one does not include easy items in a test for the selection of scholarship students. Furthermore in a psychometrically sophisticated environment quality criteria considerations may also play their role, e.g. for every knowledge item there should be two insight items. Obviously, any test design process is subject to certain constraints. In modeling the decison process (about what items to include in a test or not), these constraints are explicitely included. They play the role of limiting the search process through the test universe. Another way of looking at it is as follows. Imagine an item bank filled with three items. The total number of possible tests is $2^3 = 8$, being one test of 0 items, three tests of 1 item, three tests of 2 items and one test of 3 items. Also imagine, that the three items are represented by three indicator variables x_i, $i = 1,2,3$, with value

1 if the item i is selected for the test and value 0 if it is not. Let there also be a three-dimensional orthogonal coordinate system, where each coordinate represents an x-variable (having only the two values 1 and 0). All possible values of these three x-'s together form the collection of vertices of a cube. It is easy to see, that each vertex of this cube represents a member of the test universe. In this sense, the test universe belonging to an item bank can be seen as the collection of vertices on the hull of a convex body, which body has a dimensionality that is equal to the number of items in the bank. In this case designing a test is like travelling over the hull and checking each vertex against the constraints. Having found a subset of vertices admissable under the constraints, the task is to find that member of the subset giving most satisfaction in some defined sense.

A test always consists of a certain number of items and frequently we have started with one item and have been adding items till we were satisfied. This simple thought suggests that for the index in the sense used before, we could use a simple sum-function of indicator variabels x_i, which by its nature is linear. It remains to formulate conditions constraining us in our search for the 'right' sum-function. Before doing so, another simple sum-function will be presented first, which has proven to be of wide applicability. It will be presented in it's best known and most trivial form, as the Knapsack problem (KP). Subsequently, a form of sensitivity analysis will be presented which can useful in the analysis of test specifications. Next a way to refine test specifi-

cations will be briefly recapitulated. Finally, some problems and solutions related to the choice of algorithms will be mentioned.

General Models.

The KP has many differeent formulations of which the one leading most naturally to the general test design problem will be used. Suppose a hiker is travelling with a number of objects, each object having its own monetary value. He is now arriving at a desert which he will have to cross and knows that he will have to travel as lightly as possible. This means that he will have to leave some of the objects behind. He also knows that once across the desert, he will have to trade with the natives and needs a cerain minimum amount of cash in order to reach his psychometric laboratory again. All objects have a weight not related to their value. In summary, his problem is to minimize the weight of the content. Since an object is either present in the knapsack or not, the objects are represented by binary indicator variables $x_i = 0$ or 1. (Fractional objects have no value and negative objects don't exist). Let w_i be the weights of the objects, v_i their value and V the lower bound on total value of selected objects. The hikers problem can then be presented in formal notation as

$$\text{minimize} \sum_{i=1}^{n} w_i x_i$$

$$\text{subject to} \sum_{i=1}^{n} v_i x_i \geq V$$

and $x_i = 0$ or 1.

Due to the constraint in (1), this KP is known as a binary or zero-one programming problem; if (1) is replaced by an integer-valued variable $x_i \geq 0$, the problem is known as in integer progamming problem. If (1) is formulated as $x_i \leq 0$, it is a general linear programming problem.

The KP is formulated above but has been extended to more general forms as the multiple KP and the multidimensional KP. The multiple KP involves the same problem as above but now putting the objects into m knapsacks. The multidimensional version involves optimization under more than one constraint. In formal notation

$$\text{minimize} \sum_{i=1}^{n} \sum_{j=1}^{m} w_i x_{ij}$$

$$\text{subject to} \sum_{i=1}^{n} v_i x_{ij} \geq V$$

$$\text{and} \sum_{j=1}^{m} x_{ij} \leq 1$$

$$\text{with } x_{ij} = 0 \text{ or } 1,$$

$j = 1,2,\ldots,m$; $i = i,1,2,\ldots,n$, for the multiple KP, and

$$\text{minimize} \sum_{i=1}^{n} w_i x_i$$

$$\text{subject to } \sum_{i=1}^{n} v_i x_i \geq V$$

and, for example,

$$\sum_{i=1}^{n} p_i x_i \geq P$$

with $x_i = 0$ or 1 for the multidimensional KP, where P is the lower bound on some second constraint. A further possibility is the combination of multiple and multidimensional KP.

If the knapsacks in above examples are seen as tests that have to be loaded with items, the stage is reached where we have a simple index, the weighted sum function. If constraints meaningful in a psychometric sense can be formulated, the stage of test design as model building in mathematical programming is reached. In many test applications, a very important issue concerns accuracy of measurement. Since frequently we are only interested in one particular point on the ability continuum (e.g., a cut-off score) or at most in a limited interval of points on the continuum, the notion of concentrating on local accuracy of measurement suggests itself automatically. So frequently, part of the test specification consists of demands as regards local accuracy of estimation for a number of θ-points, expressed as test information for these points. Because of the property that test information is the sum of item information, one could develop the notion that the constraints (test specifications) take the form of linear functions appearing in equalities or inequalities. The V and the P n above KP's can

then be seen as the minimum desired amount of test information specified at two θ-points and v_i and w_i are item informations in item i at these two points. For the test design problem the w_i can be taken to be equal to 1. Writing all this in the more formal I.R.T.-based notation it can easily be seen, that the test specification "find the test of minimum length with a certain amount of information at (for example) two specified θ-points" has exactly the same structure as a KP-formulation. It looks as follows

$$\text{minimize} \sum_{i=1}^{I} x_i$$

$$\text{subject to} \sum_{i=1}^{I} I_i(\theta)x_i \geq I_t(\theta_{k=1})$$

$$\text{and} \sum_{i=1}^{I} I_i(\theta)x_i \geq I_t(\theta_{k=2})$$

$$\text{and } x_i = 0 \text{ or } 1.$$

One can imagine that, in a data base context, each item is accompanied by a string filled with coded information about various properties of the item. By using a pointer, items can be rearranged according to which characteristic is under consideration. For example, items can be coded as regards content and reordered such that x_1 to x_{100} refer to content domain A, x_{101} to x_{200} to domain B, and so forth. This enables us, for example, to add constraints specifying the proportion of items coming from certain domains to the model.

A rather different general model in automated test design involves the use of so-called Matching algorithms. For this model, it is useful to regard items and abilites as nodes in the bipartite graph G(V,E), wher V_1 is the subset of items, V_2 is the subset of abilities and E is the set of edges that connect all elements of both V-subsets with each other. Associated with each edge is a weight with value $I_i(\theta)$ at θ_k, i.e.,the information value of item i at θ_k. A matching is defined as a set of edges, where no two edges have a node in common. One of the questions that could be asked is "What is the maximum weight matching?", i.e. identify the subset in E that is not only a matching, but also has the highest possible weight-sum associated with its edges.

Models like this are useful if a number of tests of equal length that are non-overlapping in the items and of gradually increasing difficulties are to be designed. Details about this approach are presented elsewhere. It can be shown, that the structure of the problem is such that, without specifying this as a constraint in the model, the solution is always integer valued. If an upper bound of 1 to the variables is specified, the solution is automatically a zero-one solution. This implies that standard Linear Programming algorithms can be used, which are readily available.

In order to go to the next section it is necessary to anticipate as regards the section on algorithms and heuristics. For the moment is is sufficient to say that in many practical situations standard Linear Programming (LP) algorithms can also be used in the case of KP-like formulations of test design. This matter will be picked up

again later.

Analysis of Test Design Constraints

LP theory entails a powerful theorem and some very useful techniques. The theorem refered to is the Duality theorem. Regard the following LP problem:

P(1) (primal) maximize $\{c'x\}$

subject to $Ax \leq b$

and $x \geq 0$.

The duality theorem says that associated with P(1) there is an equivalent LP problem P(2), formulated as follows

P(2) (dual) minimize $\{b'u\}$

subject to $A'u \geq c$

and $u \geq 0$.

(Notice the symmetry of both formulations, written in matrix notation.) Proof of this theorem and results stemming from it can be found in the literature (see Papadimitriou and Steiglitz, 1982). For our purpose it is sufficient to know that the optimal value of the target function is the same in P(1) and P(2). An interesting feature is the appearance in the Dual of new variables u, associated with the right-hand side constraints b of the Primal. These new

variables are called the Shadow Prices (SP) of the Primal (the name originates from economic theory). Inspection of shadow prices can yield interesting insights in the original primal constraints. It can be shown, that if $z = b'u$, the following holds

$$dz/db_i = u_i \qquad i = 1,2,\ldots,m,$$

which means that if b_i changes into $b_i + \Delta b_i$, u_i shows the corresponding change in the target function z; z_i changes with u_i b_i. This means that if the SP of a constraint is equal to 0, this constraint is redundant. SP's are standard output in most commerically available LP packages. It should be noted that the above interpretation of SP is only valid for certain ranges of b. This range is known as the right-hand side range. Changes outside the range and changes in several right-hand side coefficients at the same time are sutdied in parametric programming. This matter will not be pursued any further here.

Logical Conditions in Test Design

Considering test design as a problem in zero-one programming (as first formulated in Theunissen, 1985) supplies a natural opening for the introduction of Boolean variables (Theunissen, 1986). These variables are useful if one wants to put logical conditions on test design. Suppose it is stipulated in the test specification, that if item x_1 or x_2 is selected, then at least one of the items x_3, x_4,

x_5 must also be chosen (instead of individual items, the x's may also represent strings of items) If we use as notation '→' for 'if...then' and 'V' for inclusive 'or' (a or b or both) we have two proportions, $(x_1 V x_2)$ and $(x_3 V x_4 V x_5)$, connected as follows:

$$(x_1 V x_2) \rightarrow (x_3 V x_4 V x_5).$$

It is obvious how the separate propositions are entered as constraints:

$$x_1 + x_2 \geq 1$$

and $$x_3 + x_4 + x_5 \geq 1.$$

What remains to be done is to connect these two propositions. We now introduce a new variable d and 'translate' the propositions as follows:

$$x_1 + x_2 - 2d \geq 1 \rightarrow d = 1$$

and $$d = 1 \rightarrow x_3 + x_4 + x_5 \geq 1.$$

This gives rise to the following constraints:

$$x_1 + x_2 - 2d \leq 0$$

and $$-x_3 - x_4 - x_5 + d \leq 0 \text{ (with d as binary variable).}$$

This models our logical condition. Situations can be imagined where the item selection proces is steered in a very detailed way, for example, to avoid depedencies among items. Even for relatively small sets of items this may result in rather long and complex Boolean expressions. It is useful to know that reduction algorithms for such complex Boolean formulations exists. They are used, for example, in the algebra of switching circuits (see e.g., Graham Flegg, 1965). This matter will not be pursued here any further.

In the final section of this paper some aspects of the practical practical implementation of the above models will be treated.

Algorithms and Heuristics

In discrete optimization theory, a useful distinction is the one between algorithms that work in polynomial time (P-algorithms) and those that work in non-polynomial time (NP-algorithms). Working in polynomial time means that the CPU-time necessary for the solution of the problem is a polynomial function of the size of the input. The input of an algorithm is basically a string of symbols. The size is the length of this sequence is the number of symbols in it (Papadimitriou and Steiglitz, 1982). For our type of problem, i.e., many variables (items) and relaively few constraints, the size is strongly determined by the number of items. In NP-algorithms the required CPU-time is generally an exponential function of input-size. This means that there is no guarantee that the solution can be found in reasonable time, although one does not always know this

in advance; for example, it is known (see Papadimitriou and Steiglitz, 1982) that the simplex algorithm and its derivatives in LP are NP-algorithms, but in practice extremely large problems involving thousands of varaibles and constraints have been solved without any time-problems. Integer programming, which searches for solutions that have only integer values and of which binary programming is a particular instance, is known to be a NP-problem (Papadimitriou and Steiglitz, 1982). Here it is known, that time demands may frequently turn out to be excessive in case of moderately-sized problems. A well-known algorithm in binary programming is the Balas algorithm (for details see e.g., Syslo, Deo and Kowalik, 1983). A strategy often taken in practice is that one first finds an approximate solution and uses this as a starting point for the Balas algorithm to find a purely binary solution. The Balas algorithm makes use of Branch and Bound techniques which are extensively used in all sorts of heuristics. A brief recapitulation of the basic ideas of Branch and Bound techniques will therefore be useful. The important point to remember is that, no matter its form, a Branch and Bound technique basically is a strategy to check the vertices on the hull of a convex body (see Introduction). Assuming a start-solution, obtained by LP, the first step is to choose a branch variable x_i, for example, the x with the highest fractional value. The second step is to create two sub-problems, one with $x_i = 0$ and one with $x_i = 1$, both together with all other variables. The value of the target function is now not larger and usually lower, since we have the original LP with more constraints. If the

solution is now completely integer, one stops; if not, one stops; if not, one goes on. Getting lower in the search tree gives steadily lower values for the target function. Finding the first purely binary solution could be defined as producing the first bound. The next step is to go back to a new candidate variable and repeat the process again. Any further development at a branch is stopped when the value of the target drops below the current bound before having reached an integer solution. This process continues till the search tree is complete. The choice of candidate variables (branching) and the definition of the nature of the bounds, determines the nature of the B and B algorithm. Obviously, this type of algorithm can also be used without prior approximate solution. As noted, however, time demands may be excessive. Therefore, now some heuristics approach are presented. The effectiveness of a number of therese heuristics in a test design was recently investigated by Boomsma (1986).

A well-known theorem in mathematical programming states that if we regard a continuous multidimensional KP (which is the same as saying we regard a LP with a general upper bound of 1 for the variables), then the solution for this KP consists of at most a number of fractional values, equal to the number of constraints with all other values integer, 1 or 0. Since in many test design problems the number of constraints will be low relative to the number of variables (items), this is a very useful theorem. In the experience of the author, simply rounding off the fractional values, keeping an eye on the constraints, produces excellent results at a

low price. Boomsma (1986) found excellent results with this heuristic, when he rounded all fractional values upward to 1. This guarantees results fulfilling the constraints. Inserting a backtracks mechanism by which it is checked if setting one of these rounded variables to 0 will improve the solution without violating the constraints was the final embellishment. Since the number of constraints is generally small, the solution found in this way is excellent. However, it is useful to have other heuristics that for their effectiveness are not so dependent on the number of constraints. Another heuristic investigated by Boomsma (1986) is the so-called Lagrangian heuristic. It is mentioned here, because there is some evidence that it performs well in the case of uniform test information functions (Theunissen, 1986), and also because it leads to his best general purpose heuristic, i.e., the heuristic with surrogate constraints. Suppose we have optimization problem

L(1) maximize $v'x$

subject to $Ax \leq b$, and $x_i = 0$ or 1,

then a theorem by Everett (see Salkin, 1975) says that if L is a vector of Lagrange multipliers and x_0 solves for problem

L(2) maximize $v'x - LAx$,

subject to $x_i = 0$ or 1

x_0 will also solve L(1), with b replaced by x_0. Algorithms exist, that systematically vary L, until a vector x_0 is found that aproxi-

mates b as close as possible. The 'surrogate constraints' heuristic is essenstially a Lagrangian procedure, with as multipliers the optimal values of the dual of the continuous version of the original primal. For comparisons as regards the effectiveness, the reader is referred to Boomsma (1986).

References

Boomsma, Y. (1986). Item selection by mathematical programming. [Bulletinreeks nr. 47]. Arnhem: Cito.

Luce, R.D., & Raiffa, H. (1957) Games and decisions. New York: Wiley.

Papadimitriou, C.H., & Steiglitz, K. (1982). Combinatorial optimization: Algorithms and complexity. Englewood Cliffs: Prentice-Hall.

Syslo, M.M., Deo, N., & Kowalik, J.S. (1983) Discrete optimization algorithms. Englewood Cliffs: Prentice-Hall.

Salkin, H. (1975) Integer programming. Reading: Addison-Wesley.

Theunissen, T.J.J.M. (1985). Binary programming and test design. Psychometrika, 50, 411-420.

Theunissen, T.J.J.M. (1986). Some applications of optimization algorithms in test design and adaptive testing. Applied psychological measurement, 10, in press.

Methods for Simultaneous Test Construction

Ellen Boekkooi-Timminga

Summary

An overview of simultaneous test construction methods using zero-one programming is given. The item selection process is based on the concept of information from item response theory. Next, some objective functions and practical constraints useful in simultaneous test construction are presented. Then, the special case of constructing parallel tests is considered. The paper ends with a few examples.

Some Methods for Simultaneous Test Construction

Recently, a start has been made with research on test construction from item banks using mathematical programming, in particular zero-one programming. The idea to adopt such an approach to test construction has been presented in a paper by Theunissen (1985). It has been further explored in a series of papers by Boekkooi-Timminga (1986, 1987), Boomsma (1986), Theunissen (1986), Theunissen and Verstralen (1986) and van der Linden and Boekkooi-Timminga (1986, 1987). Some references to operations research methods are Rao (1984), Syslo, Deo, and Kowalik (1983), Wagner (1972), and Williams (1978).

In this paper, methods to construct two or more tests at the same time from an item bank are presented. The possibility of doing so is of great importance whenever tests with a certain relationship to each other have to be constructed, for instance, parallel tests or tests with increasing difficulty levels (Boekkooi-Timminga, 1987).

The actual process of item selection is based on the concept of information from item response theory. All items are assumed to fit the same one-dimensional item response model. Furthermore, maximum-likelihood estimation of the subjects's abilities is assumed, so that the item and test score information functions have the property of additivity. Target values for the test information functions are specified by the test constructor at some prechosen ability levels. A procedure to obtain target values from test

constructors is described by Kelderman (1987).

Simultaneous Test Construction: The General Case

Simultaneous test construction can be viewed at as a generalization of the test construction method proposed by Theunissen (1985). The two models dealt with below clearly illustrate this. The model in (1) - (3) specifies the test construction model for one test described by Theunissen (1985). The model minimizes the number of items in the test subject to the constraints that the actual test information function values should exceed $I_t(\theta_k)$ at all K ability levels considered, where $I_t(\theta_k)$ is the desired test information function value of test t at ability level k. The model in (4) - (6) describes the construction of T tests at the same time. The total number of items over all T tests is minimized, under the constraint that for each test t and each ability level k the actual test information function values should exceed the values $I_t(\theta_k)$. The following definitions will be used: $I_i(\theta_k)$ is the item information function value for item i at ability level k. The decision variables x_i indicate if item i is selected ($x_i = 1$) or not ($x_i = 0$), whereas x_{it} indicates whether or not item i is selected for test t. The total number of items in the item bank is denoted by I.

The model for the construction of one test is as follows

(1) minimize $\sum_{i=1}^{I} x_i$

subject to

$$(2) \qquad \sum_{i=1}^{I} x_i I_i(\theta_k) > I_t(\theta_k) \qquad k = 1,\ldots,K$$

$$(3) \qquad x_i \in \{0,1\} \qquad i = 1,\ldots,I$$

The model for constructing T tests simultaneously is

$$(4) \text{ minimize} \quad \sum_{i=1}^{I} \sum_{t=1}^{T} x_{it}$$

subject to

$$(5) \qquad \sum_{i=1}^{I} I_i(\theta_k) > I_t(\theta_k) \qquad t = 1,\ldots,T; \; k = 1,\ldots,K$$

$$(6) \qquad x_{it} \in \{0,1\} \qquad i = 1,\ldots,I; \; t = 1,\ldots,T$$

Instead of minimizing the number of items many other objective functions may be used (van der Linden & Boekkooi-Timminga, 1987). In simultaneous test construction there are several possibilities. The objective function can consider aspects of all, a few or one of the tests to be constructed. For instance, the total number of items in all or in one of the tests. Some objective functions are exclusively to be used in simultaneous test construction. This is

the case when objective functions consider an aspect taking into account a relationship between all or some of the tests, such as, the difference in the actual test information function values at all or some of the ability levels considered between all or some of the tests to be constructed. In Figure 1 three possible objective functions for simultaneous test construction are presented.

Insert Figure 1 about here

During the optimization process all kinds of constraints can be taken into consideration. An overview of some constraints to be used in both simultaneous test construction and the construction of one test at a time is given in van der Linden and Boekkooi-Timminga (1987), Theunissen (1987) and in van der Linden (1987). Some constraints to be used in simultaneous test construction are listed in Figure 2.

Insert Figure 2 about here

Constructing Parallel Tests

In this section three methods for the simultaneous construction of parallel tests are discussed. Tests are considered to be parallel if their information functions are the same (Samejima, 1977). In addition to this statistical definition, it is possible to guarantee that tests are also parallel as regards content. To achieve this, additional constraints should be added in the test construction model. A discussion of these constraints concludes this section.

A possible approach to constructing parallel tests is a sequential procedure in which tests are selected after each other using a test construction model with the same specifications. However, practical experience with this approach shows that such tests tend to be far from parallel. Parallel tests can be well constructed using simultaneous test construction methods. Three methods for simultaneously constructing parallel tests are described in Boekkooi-Timminga (1986). The first method assigns items to tests. The other two methods match the tests item by item.

The objective function in the first method is based on a measure of difference between the tests to be constructed. For instance, objective function 3 in Figure 1 minimizes the maximum absolute distance between the actual test information functions. With this function, the same target test information function values are required for each test (Figure 2, constraint 1), no overlap of items between the tests is allowed (Figure 2, constraint 7), and,

if necessary, an equal number of items is assigned to each test (Figure 2, constraint 4).

The second and third methods are based on a measure of difference at item level. This measure, c_{ij} for items i and j, may be, e.g., the difference in difficulty level when the Rasch model is considered. Using method 2, items with minimum difference are assigned to different tests, subject to the condition that the test information functions satisfy the target. The third method assumes that the item bank is partitioned into as many as comparable parts as tests to be constructed. Then, the procedure of method 2 is applied selecting one test from each set in the partition. For the construction of two parallel tests, the test construction model for the second method is as follows

(7) minimize $$\sum_{i=1}^{I}\sum_{j=1}^{I} c_{ij}x_{ij}$$

subject to

(8) $$\sum_{i=1}^{I} x_{ij} + \sum_{i=1}^{I} x_{ji} < 1 \qquad j = 1,\ldots,I$$

(9) $$\sum_{i=1}^{I}\sum_{j=1}^{I} \theta_k)x_{ij} > I_t(\theta_k) \qquad k = 1,\ldots,K$$

(10) $$\sum_{i=1}^{I}\sum_{j=1}^{I} I_j(\theta_k)x_{ij} > I_t(\theta_k) \qquad k = 1,\ldots,K$$

$$x_{ij} \in \{0,1\} \qquad i = 1,\ldots,I \quad j = 1,\ldots,I, \tag{11}$$

where c_{ij} is large, compared to the other c_{ij} values, whenever $i = j$. The decision variables x_{ij} are equal to one if items i and j are matched. Items i an j should then be included in the first and second test, respectively. For both tests the same target test information functions are specified in (9) and (10). Constraint (8) indicates that an item may be selected for one test only.

By including some extra constraints in the test construction models, it is possible to assure that the tests are parallel as regards content. Indicator variables p_{is} are used to indicate if item i covers a certain subject matter s ($p_{is}=1$) or not ($p_{is}=0$). Let S be the number of topics of interest during the selection process. Then, (12) gives a set of constraints requiring that the proportions $a_1, a_2, \ldots, a_S$ of items in the test from topics s must be the same for all tests t.

$$a_1 \sum_{i=1}^{I} p_{i1} x_{it} = a_2 \sum_{i=1}^{I} p_{i2} x_{it} = \cdots = a_S \sum_{i=1}^{I} p_{iS} x_{it} \qquad t = 1,\ldots,T \tag{12}$$

Examples

Three examples of parallel tests constructed on basis of their test information functions are given (see also Boekkooi-Timminga,

1986). Examples 1, 2, and 3 were based on the methods described in the previous section. Two parallel tests had to be constructed. The test information function values were considered at the ability levels $\theta = -1, 0, 1$. The target values were the same for each example: $I_t(\theta_k) > 0.4$ at all ability levels considered. An item bank of 14 items was used. In Table 1 the item parameters and item information function values are given. Since the meaning of these examples was to explore the behavior of the three methods only on a bank of 14 items was used. Applications to more realistic domains of items have to wait for solutions to the computational complexity of zero-one programming problems.

Insert Table 1 about here

The algorithm used for solving the problems was a branch-and-bound algorithm developed by Land and Doig (1960) implemented on a DEC-2060 computer.

In the first example objective function 3 from Figure 1 was used. It had two versions: one without (1a) and the other with (1b) the constraint of both tests containing the same number of items. In Examples 2 and 3, the sum of the squared absolute differences in item information over the three ability levels was considered as a measure for the differences between the items. In Example 3 the item bank was divided into two equivalent parts. Part one consisted

of the items 2, 5, 6, 7, 9, 11, 14 and part two of the items 1, 3, 4, 8, 10, 12, 13. The results are summarized in Table 2. For each example: (1) the items selected, (2) the number of items selected, (3) the test information function values, (4) the maximum distance y between the actual and target test information function values, and (5) the maximum distance y* between the actual test information function values of the two constructed tests are given. For Methods 2 and 3, the following item pairs were produced: (1-4), (3-10), (14-12) and (6-13), (7-4), (14-12).

Insert Table 2 about here

It is clear that y* was smallest for Method 1. This result was not unexpected because this method explicitly minimizes the distance between the items. Instead, the value of y is much larger for this method than for the other two methods. Which method should be considered best is mainly a matter of taste depending on which objective the test constructor finds most important to optimize.

Conclusion

In this paper a description of simultaneous test construction methods using zero-one programming was given. First, it was shown

that simultaneous test construction methods are a generalization of the method for the construction of one test proposed by Theunissen (1985). Then, three models for the construction of parallel tests were presented. In these methods, both statistical and content aspects can be taken into consideration. Three examples were given using the methods described for the construction of parallel tests.

Algorithms for solving zero-one programming problems are known, and computer packages in which these algorithms are implemented are amply available nowadays. However, an important problem with zero-one programming problems is their computational complexity (Lenstra & Rinnooy-Kan, 1979). If one test at a time has to be constructed, CPU-time can be reduced by relaxation, which means that the decision variables x_i are allowed to take values between zero and one. However, when simultaneous test construction is involved this is not possible, because it could lead to solutions in which items are partly included in different tests. Before simultaneous test construction methods can be used in every day testing practice, more research on algoritnms and approximations will be needed. Given the large amount of research adressing this topic as well as the number quick approximative methods already obtained, it is expected that fast algorithms will be found before long.

References

Boekkooi-Timminga, E. (1986) Algorithms for the construction of parallel tests by zero-one programming. Manuscript submitted for publication.

Boekkooi-Timminga, E. (1987) Simultaneous test construction by zero-one programming. Methodika, to appear.

Boomsma, Y. (1986) Item selection by mathematical programming. Arnhem, Cito, Bulletinreeks nr. 47.

Kelderman, H. (1987) A procedure to access information functions for the construction of tests measuring multiple traits. [This Research Report].

Land, A.H., & Doig, A. (1960) An automated method for solving discrete programming problems. Econometrica, 28, 497-520.

Lenstra, J.K., & Rinnooy Kan, A.H.G. (1979) Computational complexity of discrete optimization problems. In P.L. Hammer, E.L. Johnson, & B.H. Korte (Eds.), Discrete optimization I. New York: North-Holland.

Rao, S.S. (1984) Optimization: Theory and Applications (2nd ed.). New Delhi: Wiley Eastern Limited.

Samejima, F. (1977) A use of the information function in tailored testing. Applied Psychological Measurement, 1, 233-247.

Syslo, M.J., Deo, N., & Kowalik, J.S. (1983) Discrete optimization algorithms. Englewood Cliffs, New Jersey: Prentice-Hall.

Theunissen, T.J.J.M. (1985) Binary programming and test design. Psychometrika, 50, 411-420.

Theunissen, T.J.J.M. (1986) Some applications of optimization algorithms in test design and adaptive testing. Applied psychological measurement, 10, in press.

Theunissen, T.J.J.M. (1987) Test design as model building in mathematical programming. [This Research Report].

Theunissen, T.J.J.M., & Verstralen, H.H.F.M. (1986) Algoritmen voor het samenstellen van toetsen [Algorithms for test construction] In W.J. van der Linden (Ed.), Moderne methoden voor toetsgebruik- en constructie. Lisse: Swets en Zeitlinger.

van der Linden, W.J. (1987) Automated test construction using (generalized) minimax programming. [This Research Report].

van der Linden, W.J., & Boekkooi-Timminga, E. (1986) A zero-one programming approach to Gulliksen's matched random subtests method. Manuscript submitted for publication.

van der Linden, W.J., & Boekkooi-Timminga, E. (1987) Algorithmic test design with practical constraints. Enschede, The Netherlands: Department of Education, University of Twente.

Wagner, H.M. (1972) Principles of operations research: with applications to managerial decisions. Englewood Cliffs, New Jersey: Prentice-Hall.

Williams, H.P. (1978) Model building in mathematical programming. New York: John Wiley & Sons.

Figure 1. Some Objective Functions for Simultaneous Test Construction

---

1. Minimizes the total number of items in all tests:

$$\min \sum_{i=1}^{I} \sum_{t=1}^{T} x_{it}$$

2. Minimizes the sum of the distances between the target test information functions and the actual test information functions at the ability levels considered:

$$\min \sum_{i=1}^{I} \sum_{t=1}^{T} \sum_{k=1}^{K} x_{it} I_i(\theta_k)$$

subject to

$$\sum_{i=1}^{I} x_{it} I_i(\theta_k) \geq I_t(\theta_k) \qquad \begin{matrix} t = 1,\ldots,T \\ k = 1,\ldots,K \end{matrix}$$

3. Minimizes the maximum absolute distance y between the information functions of test 1 and 2 at the ability levels considered:

min y

subject to

$$\sum_{i=1}^{I} x_{i1} I_i(\theta_k) - y - \sum_{i=1}^{I} x_{i2} I_i(\theta_k) \leq 0 \qquad k = 1,\ldots,K$$

$$-\sum_{i=1}^{I} x_{i1} I_i(\theta_k) - y + \sum_{i=1}^{I} x_{i2} I_i(\theta_k) \leq 0 \qquad k = 1,\ldots,K$$

Figure 2. Some Constraints for Simultaneous Test Construction

1. Target test information function values:

$$\sum_{i=1}^{I} x_{it} I_i(\theta_k) \gtrless I_t(\theta_k) \qquad \begin{matrix} t = 1,\ldots,T \\ k = 1,\ldots,K \end{matrix}$$

2. The number of items desired for each of the tests t:

$$\sum_{i=1}^{I} x_{it} \gtrless n_t \qquad t = 1,\ldots,T$$

3. Total number of items in all tests:

$$\sum_{i=1}^{I} \sum_{t=1}^{T} x_{it} \gtrless n$$

4. Proportions of items selected for each test given by $b_1, b_2, \ldots, b_T$:

$$b_1 \sum_{i=1}^{I} x_{i1} = b_2 \sum_{i=1}^{I} x_{i2} = \ldots = b_T \sum_{i=1}^{I} x_{iT}$$

5. Item i must be excluded from all tests:

$$x_{it} = 0 \qquad t = 1,\ldots,T$$

6. Item i must be included in precisely one of the tests:

$$\sum_{t=1}^{T} x_{it} = 1$$

7. Each item must be included in at most one test:

$$\sum_{t=1}^{T} x_{it} \leq 1 \qquad i = 1,\ldots,I$$

8. Proportions of items selected from each topic given by $a_1, a_2, \ldots, a_S$:

$$a_1 \sum_{i=1}^{I} p_{i1} x_{it} = a_2 \sum_{i=1}^{I} p_{i2} x_{it} = \ldots = a_S \sum_{i=1}^{I} p_{iS} x_{it} \qquad t = 1,\ldots,T$$

Table 1

Item Parameters and Information Function Values

Item	b_i	a_i	$I_i(-1)$	$I_i(0)$	$I_i(1)$
1	0.576	0.695	0.091	0.116	0.118
2	-0.442	1.109	0.280	0.290	0.172
3	-0.824	0.823	0.168	0.151	0.101
4	0.254	0.609	0.080	0.092	0.088
5	0.419	1.213	0.189	0.345	0.326
6	-0.017	1.138	0.240	0.324	0.236
7	-0.245	0.549	0.072	0.075	0.067
8	1.828	1.171	0.047	0.129	0.273
9	1.109	0.892	0.091	0.157	0.198
10	-0.080	0.879	0.165	0.193	0.155
11	-1.708	1.384	0.380	0.151	0.043
12	0.016	0.909	0.168	0.207	0.170
13	-0.264	1.299	0.339	0.410	0.229
14	0.063	0.936	0.173	0.219	0.182

Table 2

Results for Test Construction Methods 1 - 3

	Selected	n	$I_t(\theta)$	$I_t(0)$	It(0)	y	y*
1a	9-10-12-14	4	0.597	0.776	0.705	0.376	
	1-3-4-5-7	5	0.600	0.779	0.700	0.379	0.005
1b	3-7-9-13-14	5	0.843	1.012	0.777	0.612	
	1-2-4-6-10	5	0.856	1.015	0.769	0.615	0.013
2	1-3-14	3	0.432	0.486	0.401	0.086	
	4-10-12	3	0.413	0.492	0.413	0.092	0.019
3	6-7-14	3	0.485	0.618	0.485	0.218	
	13-4-12	3	0.587	0.709	0.487	0.309	0.102

n: number of selected items

y: maximum distance between the actual and target test information function values

y*: maximum distance between the test information function values of both parallel tests

Author's Note

This research was supported by a grant from the Dutch Organization for Pure Research (Z.W.O.) through the Foundation for Psychological and Psychonomic Research in the Netherlands (PSYCHON).

Automated Test Construction Using Minimax Programming

Wim J. van der Linden

Abstract

The use of the minimax principle in IRT-based test construction is proposed. It is shown how this results in test information functions deviating less systematically from the target function than for the usual criterion of minimal test length. Next, an alternative minimax approach is presented. Under this approach, the test constructor specifies only relative target values which serve as constraints subject to which the algorithm maximizes the information in the test. In the final part of the paper, some practical constraints are considered (e.g., test composition, administration time, mutually exclusive items, and curriculum differences), and a description of how these constraints can be included in the optimization model is presented.

Automated Test Construction
Using (Generalized) Minimax Programming

Although in IRT-based test construction a target information function for the test is specified, the actual item selection procedure usually has a different entity as its objective function. Theunissen (1985), for instance, has proposed a binary programming model for test construction in which the objective function consists of minimization of the test length. In his model, a branch-and-bound algorithm selects a test of minimal length subject to the condition that, at a number of ability points chosen in advance, the test information function lies above the target function.

Practical experience in using models with minimization of test length as the objective function shows that, for the usual item response models, the information functions usually have a large peak in the middle of the ability interval. The explanation of this phenomenon is simple. Let θ_k ($k = 1,\ldots,K$) be the values of the ability parameter considered in the model. Since the target values for the information function at these points have to be met by a minimum number of items, the algorithm will select items with the "bulk of their information" in the interval $[\theta_1,\theta_K]$. However, for the one- and two-parameter logistic models the item information functions are symmetric about their difficulty parameter values. Hence, a tendency exists to select items located in the middle of the interval. (Due to the presence of a guessing parameter, the

item information functions corresponding to the three-parameter model are skewed to the left and items somewhat to the left of the interval are preferred.) This tendency will be observed for most target functions in use in test construction. An exception is the case of a U-shaped target with large values at the extremes; then, obviously, the test will tend to contain items not in the middle but at the ends of the interval.

The above phenomenon is not only less elegant but may also nave some practical consequences. For example, the fact that all items tend to concentrate at a single point and not to be distributed over the entire interval may be less desirable as regards test content. Also, in case new tests for the same interval have to be selected on a regular basis, the supply of items in the middle of the interval may quickly be exhausted. Then the procedure no longer meets the ideal of producing short tests.

This paper is based on a twofold goal. The first goal is to propose an objective function of the minimax type to solve the above problem. Although other remedies are possible, this objective function has two other favorable properties: First, as will be shown below, the minimax principle has a generalization that suggests a simple experiment to elicit target information functions from test constructors. It is believed that this experimental approach provides a major advance in the attempt to solve the awkward problem of specifying a target information function. Second, an objective function of this type does not contain any test parameters. Therefore, the properties of the test may be

completely controlled by manipulating appropriate constraints in the model. It is the second goal of the paper to exemplify the use of this model under a variety of practical constraints.

A Minimax Test Construction Model

The purpose of having a target information function for a test is that at each of the points θ_k ($k = 1,\ldots,K$) the information about the ability parameter will be close to some prespecified value. Let $I_t(\theta_k)$ and $I(\theta_k)$ denote the actual test information at θ_k and its target value, respectively. As the test information function may approach the target values from below as well as from above, a choice needs to be made. It is henceforth assumed that the target function specifies the minimum amount of information required from the test and that $I_t(\theta_k)$ must approximate $I(\theta_k)$ from above. It follows that the relevant quantities are the (non-negative) values $\{I_t(\theta_k)-I(\theta_k);\ k=1,\ldots,K\}$ and that the objective function in the item selection model must guarantee that they are minimal in some sense.

A direct attack on the problem of peaks in test information functions is to minimize the largest deviation from the target function subject to the condition that all deviations are non-negative. This leads to the following criterion:

$$(1) \qquad \text{minimize } [\max_k \{I_t(\theta_k)-I(\theta_k);\ k=1,\ldots,K\}].$$

Although the minimax criterion specified in (1) seems to result in non-linear optimization, it is a standard transformation in mathematical programming to modify (1) so that it is in a linear form (e.g., Wagner, 1975, sect. 14.8). Let y denote an arbitrary upper bound to the set $\{I_t(\theta_k)-I(\theta_k);\ k=1,\ldots,K\}$ and let $I_i(\theta_k)$ be the value of the information function of item i ($i = 1,\ldots,I$) at the point θ_k. Now, if x_i is the decision variable as to whether ($x_i=1$) or not ($x_i=0$) to include item i on the test, a linear programming model minimizing the largest deviation may be specified as follows:

(2) minimize y

subject to

$$(3) \qquad \sum_{i=1}^{I} I_i(\theta_k)x_i - y \leq I(\theta_k) \qquad k = 1, \ldots, K$$

$$(4) \qquad \sum_{i=1}^{I} I_i(\theta_k)x_i \geq I(\theta_k) \qquad k = 1, \ldots, K$$

$$(5) \qquad x_i \in \{0, 1\} \qquad i = 1, \ldots, I.$$

The constraint in (3) requires the deviation of $I_t(\theta_k) = \sum_{i=1}^{I} I_i(\theta_k)$ from $I(\theta_k)$ to be no larger than the upper bound y; the constraint

in (4) stipulates that these deviations are non-negative. By minimizing the upper bound y in (2) the test information function tends to conform to the target function. Consequently, a test information function with the smallest possible peak is produced and the items in the test are spread out over the interval $[\theta_1,\theta_k]$. The model specified (2) through (5) can be solved for $(y, x_1, \ldots, x_I)$ by one of the branch-and-bound algorithms available for integer programming problems (Wagner, 1975, chap. 13).

It should be noted that the objective function specified in (2) is just a dummy variable introduced to cast the minimax criterion into a linear model. Hence, it does not contain any item or test parameters. This provides the test constructor with the potential for controlling any feature of the test that can be modeled as a linear constraint. Examples of such modelling will be provided below.

An Alternative Minimax Model

In IRT-based test construction it is assumed that the test constructor is able to specify a target information function. Although in general the target function of a test should be derived on the basis of its intended use, the specification of such a function is by no means an easy task. This section of the paper describes a simple experiment that may be used to elicit information about target functions from test constructors. An alternative minimax model is then presented in which elicited

information is used in item selection. An other approach to the problem of specifying target information functions is given by Kelderman (1987).

The suggested experimental approach consists of the following steps. First, the test constructor is faced with the ability scale underlying the item bank. This can be done by offering him or her a line displaying the contents of items with locations at some well-chosen points. The same practice is used in scale-score reporting of assessment data (e.g., Pandey, 1986). Then, the constructor is asked to select a number of scale points he or she wants to consider. The number of points and their spacing are free. Next, he or she is given, say, 100 chips and requested to distribute them over the scale points such that they reflect the relative distribution of information wanted from the test. The final step then is to ask the test constructor for the desired length of the test. The answer to this question can be facilitated by providing some statistics about the time typically needed by the group of examinees to complete items in the bank.

Let r_k be the numbers of chips the test constructor puts at point θ_k ($k = 1,\ldots,K$). Now the idea is to characterize the relative target information function by a series of lower bounds (r_1y, ..., r_Ky) in which y is a dummy variable to be maximized subject to the constraint that test length is equal to the value n specified by the test constructor. This leads to the following model:

(6) maximize y

subject to

$$(7) \qquad \sum_{i=1}^{I} I_i(\theta_k)x_i - r_k y \geq 0 \qquad k = 1, \ldots, K$$

$$(8) \qquad \sum_{i=1}^{I} x_i = n$$

$$(9) \qquad x_i \in \{0, 1\} \qquad i = 1, \ldots, I.$$

The constraints in (7) set a series of lower bounds, $r_k y$, to the test information $I_t(\theta_k) = \sum_{i=1}^{I} I_i(\theta_k)x_i$ at each of the points θ_k. The common factor y in these bounds is maximized in (7). The constraint in (8) sets the test length equal to n.

Just as in the previous model, the present model also tends to prevent the items in the test from concentrating in the middle of the ability interval. The reason is simply that for each test with an information function showing a large deviation from the target function at one of the points θ_k, it is likely that a test with a series of uniformly larger lower bounds $r_k y$ could be found by distributing the items more in accordance with the relative weights $(r_1, \ldots, r_k)$.

A comparison between (2) through (5) and (6) through (9) shows

that the latter has K-1 less constraints. Nevertheless, it has the additional potential for controlling the length of the test.

Some Practical Constraints

For automated test construction to be practical, it is necessary to provide control of features of the test other than just the information function and the number of items. Since the previously presented models are linear programming models, they can easily be extended through the use of additional constraints, provided these can be modeled as linear (in)equalities. In this section some practical constraints are discussed. Throughout the discussion it is assumed that (6) through (9) is the basic model.

Test Composition

As already noted, for a sufficiently large bank of items, the constraint in (8) controls the length of the test. The same principle can be applied at the level of possible subtests providing the test constructor with the ability to control the composition of the test. Let V_j ($j = 1,\ldots,J$) be a subset of items in the bank from which the test constructor wants $n_j \leq n$ in the test. This is attained if the following equality is added to the model:

$$(10) \qquad \sum_{i=1}^{I} x_i = n_j \qquad\qquad i \in V_j$$

It is important to note that for a series of such constraints the sets V_j ($j = 1,\ldots,J$) do not need to be disjoint. This provides the opportunity for controlling the composition of the test simultaneously with respect to several dimensions. For example, an item bank for English could be partitoned not only with respect to its content (e.g., vocabulary, grammar, or reading comprehension), but also to a behavioral dimension (e.g., knowledge of facts, application of rules, or evaluation) or the format of its items (e.g., multiple choice, completion, or matching). For each set in these partitions the constraint in (10) is incorporated within the model, with the restriction that the n_j's are specified such that the sum over all sets in the same partition is equal to n. If this option is used, the constraint in (8) is redundant and may be dropped.

Administration Time

In a computerized testing environment, the time needed to solve the items in the bank by the population of examinees of interest can easily be monitored. Let t_i be, e.g., the 95th percentile of the distribution of time for item i in the population. Instead of fixing the length of the test, the selection of the items could also be based on the time limit, T, in force for the examinees. In that case (8) is replaced by

$$(11) \qquad \sum_{i=1}^{I} t_i x_i \leq T.$$

Analogous to (10), the composition of the test can be controlled by introducing time limits at the subtest level.

Selection on Item Parameters

Let c_i be a positively valued numerical parameter representing a feature of the items in the bank. Then it is possible to restrict the selection of the items to those with $c_i \in [c_\ell, c_\mu]$ by including the following set of inequalities in the model:

$$c_i x_i \leq c_u \qquad i = i, \ldots, I \tag{12}$$

$$c_i^{-1} x_i \leq c_\ell^{-1} \qquad i = 1, \ldots, I\ , \tag{13}$$

where $c_u > c_\ell$.

Unlike (10), these constraints do not fix the length of subtests. Therefore they can be used to give all items in the test the same properties. At the same time, (10) can be used to compose the test with different item properties.

If the frequency of administration of the items in the bank is monitored, the constraints in (12) through (13) can be used to restrict the selection of the items to certain frequencies. For example, if the intention is to obtain uniform usage of items in the bank, (12) can be used to set an upper bound for item use thus restricting the selection of items to those with lower usage.

It is also possible to substitute one of the parameters in the item response model for c_i. In this way, the constraints can be

used, for example, to select items with values for the difficulty parameters in a certain interval. For the Rasch model, this allows for the selection of items based on their probabilities of success: Let θ_0 be the a priori known average ability of the group of examinees, and let $[p_\ell, p_u]$ be the interval to which the probabilities of success for the "average" examinees are restricted. It follows that the items must have the values of the difficulty parameter, b_i, in the interval $[b_\ell, b_u]$ determined by $p(\theta_0;b_\ell) = p_u$ and $p(\theta_0;b_u) = p_\ell$, where $p(.)$ is the logistic function specified in the Rasch model. Selecting items based on their probabilities of success for given examinees may be desirable for instructional reasons.

Group-dependent Item Parameters

If the item bank has to serve distinct groups of examinees, items may have different properties for different groups. In such cases it is obvious to consider the parameter c_i in (12) - (13) as group dependent. In school settings, for instance. the recording of the date of the final administration of item i to group g = 1, ..., G may be useful. The constraint in (13), with c_{gi} instead of c_g, then allows the selection of items for one group that have not been used after a given date for other groups. Such strategies may be instrumental in solving the problem of test security.

If c_{gi} is allowed to take only the values zero and one, it can be used to adapt tests to curriculum differences between groups. Let c_{gi} indicate whether (c_{gi}=1) or not (c_{gi}=0) item i covers a

part of the curriculum of group g. Then the following constraint automatically suppresses the administration of items to group g on topics for which instruction is absent:

$$x_i \leq c_{gi} \qquad i = 1, \ldots, I. \tag{14}$$

Inclusion or Exclusion of Individual Items

For some personal reason the test constructor may want to include or exclude certain items from the test. As already noted by Theunissen (1985) and Boekkooi-Timminga (1986), the following constraints can be used for this purpose:

$$x_i = 1 \qquad i \in V_j \tag{15}$$

$$x_i = 0 \qquad i \in V_{j_o}, \tag{16}$$

with $V_j \cap V_{j_o} = \emptyset$.

Inter-item Dependencies

In some practical situations certain items are not allowed to be administered on the same test. For instance, this may be the case if some items contain a cue with respect to the solution of other items. Suppose $i_o = 1, \ldots, I_o$ indicates a set of mutually exclusive items in the bank. Then, the following multiple-choice constraint allows the selection of at most one item from this set:

$$(17) \qquad \sum_{i_o=1}^{I_o} x_{i_o} \leq 1$$

The opposite case occurs if the selection of one item entails the necessity to select other items as well. This may occur if the content of some items builds on that of other items. (The question if one of the current response models could fit such items is deliberately omitted.) It is also possible to model the presence of this dependency between test items as a linear constraint. Let $i_1 = 1, \ldots, I_1$ represent a set of dependent items in the bank. The following equality guarantees the simultaneous inclusion or exclusion of these items from the test:

$$(18) \qquad x_{i_1} = x_{i_1+1} \qquad i_1 = 1, \ldots, I_1-1.$$

The last two constraints differ from those previously specified in that they represent dependencies among items in the bank that hold for all possibly generated tests. Therefore, they should be specified when the item bank is designed and automatically inserted in the model each time a test is constructed. Another approach to the problem of inter-item dependencies, using Boolean algebra, is given by Theunissen (1986).

Discussion

From (6) through (8), it is clear that the basic model in this paper always has a feasible solution for $n \leq I$: The constraint specified in (8) stipulates that n items are selected; from all

possible selections of this length, (6) through (7) result in the choice of the one with an information function for which the lower bounds $(r_1y,\ldots,r_Ky)$ are maximized. Thus, a sufficient condition for a model extended with additional constraints to have a feasible solution is that the intersection of their solution spaces is non-empty. This should be taken into account when specifying constraints in (10) through (18) as an addition to the basic model. For example, if (10) is used in combination with (18), it should be specified such that the lengths of the subtests are consistent with the equality constraints in (18).

Strictly speaking, a solution to the models in the paper is just a collection of test items. To make them into a test, the items should be put into an appropriate order of administration. This again can be considered as a problem of optimization subject to constraints with respect to, e.g., item difficulty, administration time, or topic structure. How this problem can be solved using a linear programming model is the subject of another paper.

As a final comment it is noted that in a computerized test system the models in this paper can also be used in an interactive mode. In doing so, the system selects a test and requests the user to indicate which items are appropriate and which are not. In the next stage, the model is used to select a new version of the test, but now with $x_i = 1$ for the items that have to be retained and $x_i = 0$ for those that were rejected. The process is repeated until all items are considered appropriate. Interactive use of the models in this paper is recommended since it allows test construction to be based on possible remaining constraints of interest that can not be modeled as linear (in)equalities.

References

Boekkooi-Timminga, E. (1987). Simultaneous test construction by zero-one programming. Methodika, 1, to appear.

Kelderman, H. (1987). Some procedures to assess target information functions. [This report].

Pandey, T.N. (1986). State of the art of large-scale assessment in the United States. In W.J. van der Linden & J.M. Wijnstra (Eds.), Ontwikkelingen in de methodologie van het onderwijs-onderzoek [Developments in the methodology of educational research]. Lisse: Swets & Zeitlinger.

Theunissen, T.J.J.M. (1985). Binary programming and test design. Psychometrika. 50, 411-420.

Theunissen, T.J.J.M. (1987). Test design as model building in mathematical programming. [This report].

Wagner, H.M. (1975). Principles of operations research. London: Prentice-Hall.

A Procedure to Assess Target Information Functions

Henk Kelderman

Abstract

To construct a test from an item bank, items are selected from the bank so that the test has a certain test information function. In this paper procedures to assess target information functions for the test are described. The probability that a certain student of ability θ_1 will erroneously obtain a higher estimate than a more able student with ability θ_2 can be derived from the information function of the particular test. The procedures to obtain information function are based on the reverse relation; from the probabilities of wrong-order mistakes (WOM), information-function values are obtained. In a dialogue between the test constructor and a computer, the procedure can be used to obtain the information functions on one or more scales.

Some Procedures to Assess Target Information Functions

Item banks are used increasingly in educational testing (Choppin, 1976, 1981; Wright & Bell, 1984). An item bank contains a large number of test items relevant for a particular curriculum. From an item bank a teacher may select a set of items to measure the ability of a group of students. In this selection proces two problems are encountered.

The first problem is that, in general, an item bank will not consist of a single homogeneous set of items fitting a one dimensional item response theory model. Typically, a bank will contain different homogeneous scales measuring the effects of particular elements of the curriculum. Bock, Mislevy and Woodson (1982) call these elements indivisible curricular elements. In selecting the items to be used in a test, the teacher has to decide which indivisible curricular elements have to be represented in the test and with what weight this has to be done. This is a problem of content validity (Thorndike 1982, ch. 7).

Secondly, for each indivisible curricular element it has to be decided how many items of what levels of difficulty must be included in the test. For example, if for a particular element only a low level of ability is required, easy items have to be included in the test.

Birnbaum (1968) has pointed out that information functions can be used for test construction. If, for a given latent trait, a tar-

get information function is known, the desired test can be constructed by selecting items in such a way that the information function for the test approximates the target function. Lord (1980), Theunissen (1985) and Boekkooi-Timminga (1986) describe methods to do this. Theunissen (1985) describes a procedure based on integer programming, a special branch of linear programming. Boekkooi-Timminga (1986) uses integer programming to construct several tests simultaneously starting with several information functions for different tests.

To employ these methods, for each of the scales a target information function must be known. To date, however, no satisfactory method to specify target information functions is available.

In this paper, first the concept of an information function and its properties are described. Some interpretations of this concept are discussed and an interpretable function of information is chosen: the probability of a wrong order of the ability estimates of two subjects.

Using this interpretation, a paired comparison experiment is proposed that yields the values of the information function for different scale points by comparing their wrong order probabilities. These experiments can be used in an interactive procedure to specify information functions for scales.

Test Information Functions

Consider a test measuring q traits, where each trait r (=1,...,q)

is measured by n_r items with responses $\mathbf{U} = (U_{r1},\ldots,U_{rn_r})$. Each item response u_{ri} can take values 0 (wrong, negative) or 1 (right, positive). It is assumed that the responses are locally independent

$$P(\mathbf{U}_r=\mathbf{u}_r \mid \theta^{(r)}) = \prod_{i=1}^{n_r} P(U_{ri}=u_{ri} \mid \theta^{(r)}), \tag{1}$$

that is, the dependence between the item responses is wholly explained by their dependence on an unidimensional latent trait $\theta^{(r)}$. Several item response models are proposed for $P(U_{ri}=u_{ri} \mid \theta^{(r)})$. For example Rasch (1980) gives the model

$$P(U_i=u_i \mid \theta) = \frac{\exp(u_i(\theta-\delta_i))}{1+\exp(\theta-\delta_i)}, \tag{2}$$

where δ_i is a parameter for the difficulty of item i and the index r denoting the trait is dropped.

The amount of information about θ contained in the subtest $\mathbf{U}_r$ is defined as (Kendall & Stuart, 1978, p. 10):

$$I(\theta) = E_\theta\left[\left(\frac{\partial}{\partial\theta}\log P(\mathbf{U}_r|\theta)\right)^2\right] \tag{3}$$

If the items are locally independent (1), we have (Lord, 1980):

$$I(\theta) = \sum_{i=1}^{n} I_i(\theta), \tag{4}$$

where

$$I_i(\theta) = E_\theta\left[\left(\frac{\partial}{\partial\theta}\log P(U_i|\theta)\right)\right] \tag{5}$$

is the information about θ in the response to item i. For example, in the Rasch model the item information function is

$$I_i(\theta) = [2 + \exp(\delta_i - \theta) + \exp(\theta - \delta_i)]^{-1} \qquad (6)$$

If t is an unbiased estimator of some function $\tau(\theta)$ of θ, the Cramer-Rao inequality (Kendall & Stuart, 1978, p.10; Lord, 1980, p.71) states that

$$\mathrm{Var}(t|\theta) \geq \{\tau'(\theta)\}^2 / I(\theta) \; . \qquad (7)$$

If $\hat{\theta}$ is a maximum likelihood estimator of θ we have asymptotically (Lord, 1980, p. 71):

$$\mathrm{Var}(\hat{\theta}|\theta) = I^{-1}(\theta) \qquad (8)$$

Furthermore, $\hat{\theta}$ has asymptotically a normal distribution with expectation θ and variance $I^{-1}(\theta)$ (Oosterloo, 1984).

Interpretations of Test Information

To obtain a target information function $I(\theta)$ from a test constructor, some suitable interpretation of both the latent trait value θ and its associated information value $I(\theta)$, must be available. Unfortunately, neither θ nor $I(\theta)$ have a straightforward interpretation. Before a procedure for obtaining information functions can

be constructed, we must therefore relate θ and $I(\theta)$ to quantities that do have an interpretation familiar to the test constructor.

Interpretation of Ability Level

Firstly, the ability level can be expressed in terms of the expected observed score (true score) of some subtest with which the test constructor is familiar. Let A be the set of items of this test, then

$$\tau_1(\theta) = \sum_{i \in A} P(U_i=1|\theta) \qquad (9)$$

is the true score, where $P(U_i=1|\theta)$ is some IRT model. In general, however, it may not be expected that test constructors are very familiar with specific tests, let alone homogeneous subtests pertaining to indivisible curricular elements.

A second way to give an interpretation to the ability level is to relate θ to percentiles for a reference population with which the test constructor is familiar, i.e.

$$\tau_2(\theta) = 100.F(\theta) \quad , \qquad (10)$$

where $\tau_2(\theta)$ is the percentile point for ability level θ and F is the cumulative density function of θ in the population of interest, e.g. students of a certain grade level in a certain school type.

In general, it may be expected that the test constructors are more familiar with subpopulations than which specific tests.

Consequently percentile points may be the prefered way to express ability level. In the application of percentile points, however, a good description of the trait to be measured should be given. This trait can be described verbally in terms of teaching materials or in terms of test items.

A third way is to give an interpretable representation of different ability levels is to provide test items with difficulty levels corresponding to the particular θ level. Instead of test items also mean ability levels of particular reference groups can be used.

Interpretation of Information

A more difficult problem is the one of interpreting test information. For functions τ of θ we can use the relation

$$I^{1/2}(\theta) = [\tau'(\theta)] \; / \; \mathrm{var}(t|\theta)^{1/2} \qquad (11)$$

which can be interpreted as "the slope of the regression of t (i.e. the observed value of $\tau(\theta)$, on θ relative to the standard error of measurement of t for fixed θ" (Lord, 1980, p. 67). This interpretation can be used for both true scores (9) and percentiles (10), but it has the drawback of still referring to a θ scale which is not interpretable. Moreover the 'standard error of measurement' is not directly a very easily interpretable quantity either.

A second way to get to an interpretable quantity from which information values can be obtained is to use the property that for a

latent trait value θ the estimator $\hat{\theta}$ has an asymptotic normal distribution with variance $I^{-1}(\theta)$. The asymptotic confidence interval has length $I^{-1/2}(\theta)$ so that the .95 confidence interval is $\hat{\theta} \pm 1.96\, I^{-1/2}(\theta)$. The interpretation of a confidence interval "The probability that the interval covers the true value is .95" is not easy to comprehend for test constructors who are unfamiliar with statistics.

A third way to derive an interpretable quantity from test information is as follows. Consider two individuals with true ability levels θ_1 and θ_2, respectively, where the second individual is more able than the first. If the values of the target information function for θ_1 and θ_2 are small, the variances of the estimated ability levels $\hat{\theta}_1$ and $\hat{\theta}_2$ are large. In that case, the probability that the first individual is erroneously estimated to be more able than the second individual becomes also large.

The probability of such a wrong order mistake (WOM) can be derived as follows. Because $\hat{\theta}_1$ and $\hat{\theta}_2$ are ability scores obtained from different persons who respond independently to the test (Kreyszig 1970, p. 173)

$$\mathrm{Var}(\hat{\theta}_1 - \hat{\theta}_2) = \mathrm{Var}(\hat{\theta}_1) + \mathrm{Var}(\hat{\theta}_2) \qquad (12)$$

Furthermore, from the fact that $\hat{\theta}_1$ and $\hat{\theta}_2$ have an asymptotic normal distribution with mean θ_1 and θ_2 and variance $I^{-1}(\theta_1)$ and $I^{-1}(\theta_2)$, respectively, we have

$$P(\hat{\theta}_1-\hat{\theta}_2) = N(\theta_1-\theta_2, I^{-1}(\theta_1) + I^{-1}(\theta_2)) \ , \qquad (13)$$

so that the probability of putting both persons in the wrong order of ability is:

$$P(\hat{\theta}_1>\hat{\theta}_2) = P(\hat{\theta}_1-\hat{\theta}_2 > 0) =$$

$$= P\left(\frac{(\hat{\theta}_1-\hat{\theta}_2) - (\theta_1-\theta_2)}{I^{-1}(\theta_1) + I^{-1}(\theta_2)} > \frac{-(\theta_1-\theta_2)}{I_1^{-1}(\theta_1) + I^{-1}(\theta_2)}\right) =$$

$$= 1 - \Phi\left(\frac{-(\theta_1-\theta_2)}{I^{-1}(\theta_1)+I^{-1}(\theta_2)}\right) =$$

$$= \Phi\left(\frac{\theta_1 - \theta_2}{I^{-1}(\theta_1) + I^{-1}(\theta_2)}\right) , \qquad (14)$$

where Φ is the cumulative normal distribution function. From (14),

$$I^{-1}(\theta_1) + I^{-1}(\theta_2) = (\theta_1-\theta_2)\{\Phi^{-1}(P(\hat{\theta}_1>\hat{\theta}_2)\}^{-1} \qquad (15)$$

so that the sum of the reciprocals of the information values for two known scale points can be obtained if the wrong-order probabilities are known. Furthermore, from (15) we have:

$$I(\theta_2) = 1/\{(\theta_1-\theta_2)\ [\Phi^{-1}(P(\hat{\theta}_1>\hat{\theta}_2))]^{-1} - I^{-1}(\theta_1)\} \qquad (16)$$

If one of the information values is known, the wrong-order probability can be used to determine the other. If both information values are unknown but can be assumed to be approximately the same we have

$$I_2(\theta) = 2/\{(\theta_1-\theta_2)[\Phi^{-1}(P(\hat{\theta}_1>\hat{\theta}_2))]^{-1}\} \qquad (17)$$

Since measurement in education is concerned with comparisons between persons on particular traits and the fairness of these comparison is important to most test constructors, we will choose for an interpretation of information functions in terms of wrong-order mistakes.

Assessment of Information through Wrong-order Mistakes

The above relationship can be used to give a sensible interpretation to an information function for the precision of measurement in terms of wrong-order probabilities. It can also be used the other way around. That is, the desired measurement precision may be specified in terms of wrong-order probabilities to obtain the associated target information function.

To obtain a target information function for a certain scale, a number of equidistant points may be chosen for which the information values will be determined. Three to five of such target information values suffice for the construction of a test using integer programming (Theunissen, 1985). For each pair consecutive pair of scale points, a test constructor, typically a teacher, is then asked to specify the wrong order probability that (s)he is willing to accept for that particular pair. The target information function is then calculated using formula (16). Since formula (16) supposes that one information value is already known, one more pair of scale

points must be judged to remove the indeterminacy.

Although the idea of wrong order is easy to grasp, the procedure is not entirely satisfactory. Firstly, producing a probability is still a difficult task. The teacher might not be willing to accept any order mistakes at all! But on the other hand he might not be willing to pay the price of a very long test, or not believe that that is the price to pay.

Secondly, this procedure is restricted to one scale only. We need however a procedure that simultaneously yields information functions for a number of homogeneous subscales. The procedure must give reliable information concerning the relative heights of these information functions so that a composite test can be constructued. In this section of the paper, some procedures to be presented are based on paired comparisions of wrong-order mistakes that avoid asking for probabilities and can be used to obtain information about the relative heights of information functions for different traits. This can be done through a procedure based on the comparison of two paires of items: The pair completion experiment.

The pair-completion experiment is as follows. The test constructor is confronted with three appropriately anchored scale points. (S)he must now provide a fourth scale point so that the wrong-order mistake in scale points one and two is equally serious as a wrong-order mistake in scale points three and four. We will show now that if the information values of the first three scale points are known, the information value of the fourth scale point can be calculated.

The basic idea is that two wrong order mistakes of the same seriousness are allowed to occur with the same probability, that is

$$P(\hat{\theta}_{11}>\hat{\theta}_{12}) = P(\hat{\theta}_{21}>\hat{\theta}_{22}), \tag{18}$$

where $\hat{\theta}_{11}$ and $\hat{\theta}_{12}$ are the estimated latent trait values of the first pair and $\hat{\theta}_{21}$ and $\hat{\theta}_{22}$ are the estimated latent trait values of the second pair.

From (18) and (14) and the fact that the cumulative normal distribution function has an inverse, we have

$$I_2^{-1}(\theta_{21}) + I_2^{-1}(\theta_{22}) = \frac{\theta_{21}-\theta_{22}}{\theta_{11}-\theta_{12}} (I_1^{-1}(\theta_{11})+I_1^{-1}(\theta_{12})), \tag{19}$$

so that

$$I_2(\theta_{22}) = 1/[\frac{\theta_{21}-\theta_{22}}{\theta_{11}-\theta_{12}} (I_1^{-1}(\theta_{11})+I_1^{-1}(\theta_{12}))-I_2^{-1}(\theta_{21})], \tag{20}$$

where I_1 and I_2 are the information functions for the first and the second pair, respectively. These information functions are indexed because the first and the second pair of scale values may be from different latent traits.

If the information values of the given scale points θ_{11}, θ_{12} and θ_{21} are known, the information value of the scale point θ_{22} that is provided by the test constructor (through his choice of an item or a percentile point) can be calculated using (20). If the information values in the second pair are assumed to be

approximately the same, we have from (19):

$$I_1(\theta_{21}) \doteq I_2(\theta_{22}) \doteq 2\,\frac{\theta_{11}-\theta_{12}}{\theta_{21}-\theta_{22}}\;\;\frac{1}{I_1^{-1}(\theta_{11})+I_1^{-1}(\theta_{12})}\;. \qquad (21)$$

If the information values are assumed to be approximately the same in both members of the first pair i,e. $I_1(\theta_{11}) = I_1(\theta_{12})$, we have

$$I_2(\theta_{22}) \doteq 1/[2\,\frac{\theta_{21}-\theta_{22}}{\theta_{11}-\theta_{12}}\,I_1^{-1}(\theta_{11}) - I_2^{-1}(\theta_{21})]\;, \qquad (22)$$

and if the information values within both pairs are approximately the same, i.e. $I_1(\theta_{11}) = I_1(\theta_{12})$ and $I_2(\theta_{21}) = I_2(\theta_{22})$, we have

$$I_2(\theta_{22}) \doteq \frac{\theta_{11}-\theta_{12}}{\theta_{11}-\theta_{12}}\;\;I_1(\theta_{11})\;. \qquad (23)$$

A Procedure to Determine Multiple Information Functions

The pair-completion experiment can be used repeatedly to obtain information values for a number of points on one or more scales. It is supposed that the test constructor is seated in front of a terminal. The terminal presents the scale points appropriately anchored by item content and/or percentile points on a line. The following steps are now taken to determine the information functions.

Step 1 The test constructor is asked to select an interval on one of the scales around the point (s)he is most interested in. The items corresponding to the end points of the

interval are highlighted.

Step 2 The test constructor is requested to give the WOM probability for the end points of the interval. The end points are assumed to have the same information values.

step 3 A new point on the scale scale is found using the pair completion experiment. In this experiment, the points of the first pair are the end points of the interval and the first point of the second pair is one of the end points of the interval. The information value of the new point is calculated by formula (22).

Step 4 In the same way as in step three, information values are determined for points at the right (left) side of the scale, using the two rightmost (leftmost) points as points with known information values. The information value of the new point is calculated by formula (20). At each step, the test constructor may decide to stop adding points.

Step 5 The test constructor is asked to select another scale and a point on that scale (s)he is most interested in.

Step 6 The information value of this point and another point on the new scale is found by executing the pair completion experiment. In this experiment, the points of the first pair are the end points of the interval on the old scale and the first point of the second pair is the the point chosen on the new scale. Assuming that the information values of both points on the new scale are the same, the information value of the new points are calculated by

formula (23).

Step 7 Step 4 is repeated for the new scale.

step 8 Step 5 through 7 is repeated another new scale until all relevant scales are processed.

In this fashion information values are obtained for a number of points on the on a number of scales. Note that the test constructor can determine how many points on the scale (s)he wants to have. In many applications this number will be small. so that the procedure need not take too much time.

Discussion

The procedure presented in this paper yields a number of information functions to be used for constructing a test measuring a number of homogeneous traits.

The procedure to determine information functions is carried out only once. No replications are made to check the reliability of the judgments. Checks could be built into the procedure but would require a lot of time from the test constructor and may make the method impractical.

The unit of precision by which the functions are measured is chosen by the test constructor in step 5. This might be hard to be do no. The size of this unit also has a large effect on the total duration of the procedure. Therefore the unit might be chosen by the system to limit the duration of the procedure.

In this paper a pair completion experiment is performed. In some cases this will not be feasible because there might not be a scale point available for which the WOM in the second pair is as serious as a WOM in the first pair. For example, if it is considered much more important to measure one trait than another trait, a WOM for the standard pair in the first scale may be more serious than a WOM for any pair of scale points in the second trait. As a result the test constructor is not able to indicate a new point on the second scale. In that case another experiment may be devised where all four points are furnished by the system. The test constructor is then asked to distribute a number of dollars (say 100) over the two pairs in proportion to the seriousness of a WOM in each of the pairs. The relative probability of a WOM is now taken inversely proportional to the relative seriousness of a WOM, so that the expected value of WOM seriousness is equal in both pairs. For example, if a WOM in the second pair is considered twice as serious, the probability of a WOM in the second pair becomes half the probability of a WOM in the first pair. Formulas for determining the information values of the points similar to (20) through (22) can be derived along the lines of the present paper.

References

Birnbaum, A. (1968). Some latent trait models. In F.M. Lord & M.R. Novick, Statistical theories of mental test scores. Reading MA: Addison-Wesley.

Bock, R.D., Mislevy, R. & Woodson (1982). The next stage in educational assessment. Educational Researcher, 11, 4-11.

Boekkooi-Timminga, E. (1986). Simultaneous test construction by zero-one programming. Enschede, The Netherlands: Twente University of Technology.

Choppin, B.H. (1976). Recent developments in item banking. Advances in Psychological and Educational Measurement. New York: Wiley.

Choppin, B.H. (1981). Educational measurement and the item bank model. In Lacey, C. and D. Lawton (Eds.) Issues in evaluation and accountability.

Kendall, M. & Stuart, A. (1978). The advanced theory of statistics, Vol. 2, London: Charles Griffin & Co.

Kreyszig, E. (1970). Introductory mathematical statistics: Principles and Methods. New York: Wiley.

Lord, F.M. (1980). Applications of item response theory to practical testing problems. Hillsdale, N.J.: Lawrence Erlbaum.

Oosterloo, S.J. (1984). Confidence intervals for test information and relative efficiency. Statistica Neerlandica, 38, 37-54.

Rasch, G. (1980). Probabilistic models for some intelligence and attainment tests. Chicago: The University of Chicago Press.

Theunissen, P. (1985). Binary programming and test design. Psychometrika, 50, 411-420.

Thorndike. R.L. (1982). Applied Psychometrics. New York: Houghton Mifflin.

Wright, B.D., & Bell, S.R. (1984). Item banks: what, why, how. Journal of Educational Measurement, 21, 331-346.

Authors' Note

Portions of these papers were presented in the Symposium IRT-based Test Construction at the Annual Meeting of the American Educational Research Association, Washington, D.C. April 20-24, 1978.

T.J.J.M. Theunissen is at the National Institute of Educational Measurement, Arnhem, The Netherlands.

Titles of Recent Research Reports

RR-86-1 W.J. van der Linden, The use of test scores for classification decisions with threshold utility

RR-86-2 H. Kelderman, Item bias detection using the loglinear Rasch model: Observed and unobserved subgroups

RR-86-3 E. Boekkooi-Timminga, Simultaneous test construction by zero-one programming

RR-86-4 W.J. van der Linden, & E. Boekkooi-Timminga, A zero-one programming approach to Gulliksen's random matched subtests method

RR-86-5 E. van der Burg, J. de Leeuw, & R. Verdegaal, Homogeneity analysis with k sets of variables: An altenating least squares method with optimal scaling features

RR-86-6 W.J. van der Linden, & T.J.H.M. Eggen, An empirical Bayes approach to item banking

RR-86-7 E. Boekkooi-Timminga, Algorithms for the construction of parallel tests by zero-one programming

RR-86-8 T.J.H.M. Eggen, & W.J. van der Linden, The use of models for paired comparisons with ties

RR-86-9 H. Kelderman, Common Item Equating Using the Loglinear Rasch Model

RR-86-10 W.J. van der Linden, & M.A. Zwarts, Some Procedures for Computerized Ability Testing

RR-87-1 R. Engelen, Semiparametric Estimation in the Rasch Model

RR-87-2 W.J. van der Linden (Ed.), IRT-based Test Construction

Research Reports can be obtained at costs from Mediatheek, Faculteit Toegepaste Onderwijskunde, Universiteit Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

A publication by
the Department of Education
of the University of Twente,
P.O. Box 217,
7500 AE Enschede,
The Netherlands

Department of Education