Improving Efficiency of Cognitive Diagnosis by Using Diagnostic Items and Adaptive Testing

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Kentaro Kato

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

William M. Bart and Ernest C. Davenport, Jr., Advisers

October 2009

Acknowledgements

First of all, I would like to thank my two advisers, Professor William Bart and Professor Ernest Davenport. The basic idea of this thesis stemmed from my work with Professor Bart. He consistently showed me the importance of connecting cognitive psychology and psychometrics for better use of educational assessments for student learning and instruction. He was always supportive and patient with my slow progress. Professor Davenport gave me many helpful advices on planning the studies in this thesis as well as thesis writing.

My appreciation extends to my other examining committee members, Professor Mark Davison and Professor David Weiss. They provided me with helpful and constructive comments and suggestions at every critical moment in the course of completing this thesis.

Hiroshi Watanabe, Professor Emeritus at the University of Tokyo, taught me all basic and advanced skills and knowledge in psychometrics and statistics in my early years as a graduate student. I could not have completed my doctoral program without these skills and knowledge. I also appreciate Professor Jan Boom at Utrecht University, the Netherlands, for kindly providing his Balance Scale data.

In addition to these academic supports, I must note that encouraging words from my colleagues and friends in the U.S. and Japan pushed me toward the goal. Finally, I would like to thank my parents for their continuing moral and material support.

Abstract

Diagnosing student use of problem-solving strategies (cognitive rules) is one of the major concerns of cognitive diagnostic assessment (CDA). With such information, subsequent instruction can be effectively tailored to meet individual student learning needs. For this purpose, student responses to assessment items should be interpretable in terms of cognitive rules. However, most of the current implementations of CDA still treat student responses dichotomously (correct versus incorrect), leading to substantial loss of diagnostic information. Diagnostic items are multiple choice items in which response options are explicitly associated with certain cognitive rules, and considered as a potentially useful tool for more efficient CDA. These considerations lead to the following research questions: (a) to what extent does the use of diagnostic items improve efficiency of cognitive diagnosis from treating student responses dichotomously, and (b) what characteristics of diagnostic items are more responsible for efficiency improvement than others? The present research comprised two studies and approached these questions quantitatively, using (a) a latent class model as a psychometric model for diagnostic items, (b) the number of items administered to reach a diagnosis as a measure of efficiency of cognitive diagnosis, and (c) adaptive testing simulations in which optimal items are sequentially selected for each examinee. Study 1 examined efficiency improvement using simulated responses to hypothetical diagnostic items whose characteristics were varied systematically. Study 2 examined efficiency improvement found in existing real response data on Siegler's Balance Scale Task. Both studies supported the use of diagnostic items in that they substantially improved efficiency of cognitive diagnosis, although several item characteristics had differential effects on the efficiency. Limitations to these studies included validity of the latent class model and simulation settings. Based on these limitations, future research should be directed to further understanding of student response behaviors relevant to cognitive rule usage and the corresponding extension of the current latent class model for diagnostic items.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cognitive diagnostic assessment (CDA) is a type of educational assessment that is "designed to measure specific knowledge structures and processing skills in students so as to provide information about their cognitive strengths and weaknesses" (Leighton & Gierl, 2007c, p. 3). Its goal is to provide instructionally relevant information with which teachers can effectively tailor subsequent instruction to address individual student learning needs.

There are increasing demands for CDA (e.g., Leighton & Gierl, 2007a; National Research Council, 2001; Nichols, 1994; Nichols, Chipman, & Brennan, 1995), and they come from different parts of society such as educators, policymakers, and assessment developers. Societal demands have become more apparent since the introduction of the No Child Left Behind Act of 2001, which requires states to produce diagnostic score reports for parents, teachers, and school principals so that they can understand and address specific learning needs of students. However, a nation-wide survey conducted by Huff and Goodman (2007) revealed that teachers would need more diagnostic information to improve their instructional planning, and such information would be different from what the current large-scale assessments provide. Assessment developers, recognizing these demands from users, are moving toward development of new types of assessments that can offer diagnostic informa-

tion. This movement coincides with the emergence of new assessment design frameworks, psychometric models, and computerized assessment systems that allow delivery of innovative item types and scoring techniques (Embretson, 1999; Leighton & Gierl, 2007a; National Research Council, 2001; Nichols et al., 1995; Stout, 2002; Williamson, Mislevy, & Bejar, 2006).

An even more radical view posits that the shift to assessments that measure student thinking processes is necessary, because assessments have the power to direct what is taught and how it is taught in school (e.g., Partnership for 21st Century Skills, 2005; Resnick & Resnick, 1992). Assessments of thinking processes rather than learning outcomes would direct teachers and students to focus on how students think and how they apply knowledge and skills to real life problems. At present, most assessments just allow one to know if a student got a particular question correct regardless of the reason.

In order to provide deeper understanding about student learning, CDA requires a different design principle from that of traditional assessments. The following remark by Messick (1994) aptly describes the guiding principle of CDA:

> A construct-centered approach would begin by asking what complex of knowledge, skills, or other attributes should be assessed, presumably because they are tied to explicit or implicit objectives of instruction or are otherwise valued by society. Next, what behaviors or performances should reveal those constructs, and what tasks or situations should elicit those behaviors? Thus, the nature of the construct guides the selection or construction of relevant tasks as well as the rational development of construct-based scoring criteria and rubrics. (p. 17)

As a result, development of CDA is made through collaborative work between cognitive psychologists and psychometricians (Fu, 2005; Leighton & Gierl, 2007c; Nichols, 1994; Nichols et al., 1995).

The National Research Council (2001, chap. 2) restated the construct-centered approach by introducing the concept of the assessment triangle, which consists of three interconnected elements: cognition, observation, and interpretation. *Cognition* refers to a cognitive model, that is, a model about "how students represent knowledge and how they develop competence in a subject domain" (National Research Council, 2001, p. 44). A cognitive model provides a description of what should be assessed, but it differs from a general test specification in several aspects. First, a cognitive model specifies not only a narrative definition of the target construct but also cognitive components and processes which constitutes the construct. Second, such detailed specifications make more explicit implications for instructional feedback. Third, these specifications result from a cognitive theory, i.e., an empirically supported model about particular cognitive processes that are relevant to the construct being assessed (Leighton & Gierl, 2007b).

Cognition guides specifications in the other elements of the assessment triangle. *Observation* specifies assessment tasks that elicit student behaviors relevant to the construct. With a more detailed description of the construct, one can be more specific about what context or task situation should be presented to students and what kind of behaviors or performances should be observed in response to the given task situation.

Also, cognition provides a basis for *interpretation*, which specifies how cognitive diagnosis is made based on observed behaviors of students. In this regard, Mislevy (1994, 1996) viewed assessment as a process of evidentiary reasoning, that is, a process of reasoning (or inference) about student knowledge states based on their manifest responses to assessment tasks. A cognitive model provides a frame of reference as to how student responses are cognitively interpreted in terms of providing evidence for a particular configuration of cognitive model variables.

One of the key components in the interpretation part is the psychometric model, which enables systematic and objective inferences from student responses to knowledge states.

Psychometric models used for CDA are called cognitive diagnostic psychometric models (CDPMs). The construct-centered approach requires that a CDPM be compatible with the cognitive model and the type of responses from assessment tasks, that is, it should specify how student responses depend on the components of the cognitive model and vice versa through a probabilistic and mathematical formulation. The selection and validation of a CDPM is crucial in CDA.

There are several assessment design frameworks that operationalize the concept of the assessment triangle such as the one described by Nichols (1994), the evidence centered design (ECD; Mislevy, 1994, 1996; Mislevy, Almond, & Lukas, 2003), and the cognitive design system (Embretson & Gorin, 2001). Although these frameworks differ in their emphases on different parts of assessment design and the level of details, all of them share the above principle of the assessment triangle.

The construct-centered approach to the design of CDA is contrasted with the design of traditional assessment, in which test items are written only with quite general content specifications and test construction is driven by examining psychometric properties of those items (Embretson & Gorin, 2001; Leighton & Gierl, 2007c). Assessments designed in this manner have limited construct validity, because construct validation can be conducted only *after* they are established as having good psychometric properties such as high reliability and scalability, and there is little room where cognitive theory plays its role to establish an interpretive framework of observed scores (Embretson & Gorin, 2001).

In this thesis, several defining features in the design of CDA along the three components of the assessment triangle are first reviewed. The literature suggests the utilization of errors (incorrect responses) in diagnosing student cognitive states and the use of a special type of multiple choice test items termed diagnostic items. While diagnostic items have the potential to improve cognitive diagnosis by providing better interpretability of student responses and better efficiency, there has been no research that clearly showed the degree

of improvement, especially in terms of efficiency improvement. Accordingly, two studies are conducted to examine efficiency improvement brought by diagnostic items. These studies feature a special form of latent class model compatible with diagnostic items and adaptive testing simulations to measure the efficiency of cognitive diagnosis.

The composition of this thesis is as follows. Chapter 2 reviews the literature, identifies current issues, and presents research questions. Chapter 3 describes the methodology. Each of chapters 4 and 5 presents a study that address the research questions. Finally, chapter 6 concludes the studies with discussion, limitations, and future directions.

Chapter 2

Review of Literature

2.1 Cognition in CDA: Several Distinctive Features

A model of cognition is a foundation of CDA that guides the other components of the assessment triangle. A cognitive model builds upon some theory of cognition. Although there is no universal theory of cognition that fits every particular domain of learning, it is still possible to discern what cognitive research suggests in general about student learning.

Snow and Lohman (1993) summarized advances in cognitive psychology up to the early 1990s that are relevant to, and prompt new forms of, psychological testing and psychometric modeling. Their summary was organized under two broad categories: cognitive abilities and cognitive achievement. On the one hand, new views of cognitive abilities included component processing skills required in ability test performance (e.g., stimulus encoding, feature comparison, rule induction, and response justification) and strategy uses and shifts (i.e., students use different strategies and often change and adapt strategies in response to task characteristics as they learn).

On the other hand, analysis of cognitive achievement entailed understanding of declarative and procedural knowledge in terms of its acquisition and structure. Learning invokes

restructuring of existing knowledge; students restructure existing knowledge to form new knowledge as they acquire new information by chunking, elaboration, and connection to the existing knowledge. Snow and Lohman (1993) presented several forms of knowledge structures such as semantic networks, schemata, scripts, prototypes, images, and mental models, and how they change as learning proceeds.

One of the implications in Snow and Lohman's (1993) review is that changes that learning evokes are often *qualitative* rather than quantitative. Accordingly, CDA needs to address qualitatively different cognitive states instead of, or in addition to, quantitative "degree" of the target construct, which has been the main focus of traditional assessments.

The construct-centered approach to CDA entails a second consideration: how these qualitative differences manifest themselves as observable responses to assessment tasks? One of the most straightforward indicators is *errors* (or *incorrect* responses) that students make, "because different kinds of knowledge structures should produce different patterns of responses and particular kinds of errors, tests might be designed to elicit different error patterns from different structures" (Snow & Lohman, 1993, p. 8). This makes a sharp contrast to the main focus of traditional assessments on how many *correct* responses students produce. In fact, several psychometricians also recognized errors as a key informant in cognitive diagnosis. Bejar (1984) stated that examining the pattern of errors (error analysis) is one of the major approaches to developing diagnostic assessments, referring to studies of errors in various subject areas such as reading, mathematics, and writing. In a similar vein, Tatsuoka (1990) stated that incorrect responses have implications for understanding student knowledge states better, and should be investigated more closely.

Gitomer and van Slyke (1988) distinguished three types of errors. The first type is that of rule-based errors, which are direct functions of misconceptions. Rule-based errors are generated by systematically applying incorrect strategies or insufficient knowledge, so one can predict an error response for each given misconception. The second type of error is

termed idiosyncratic. Patterns of errors over a set of tasks are idiosyncratic if they are not consistent with rule-based errors, and thus no misconceptions can be inferred. The third type of error is due to inadequate efficiency. This type of error occurs when students understand all required concepts and skills but are unable to access their knowledge efficiently in a given testing environment (e.g., a limited amount of time to complete the test). As a result, they may fail tasks with relatively high memory or cognitive demands.

Cognitive modeling is concerned primarily with rule-based errors. Bart and Williams-Morris (1990) used the term *cognitive rule* to refer to "a sequence of one or more cognitive operations that permits an individual to generate a response" (p. 147), and such rules include "problem-solving strategies, decision making strategies, and algorithms" (Bart, Post, Behr, & Lesh, 1994, p. 2). A cognitive rule can be defective if inappropriate cognitive operations are applied or the order of applying the operations is wrong, resulting in a specific incorrect response. Analysis of student problem solving behavior identified various strategies used by students in several learning domains. For example, Brown and Burton (1978) analyzed student response behaviors on simple number arithmetic problems. They found that by close examination even seemingly random responses could result from the consistent application of a certain procedure (i.e., cognitive rule), and that "students are remarkably competent procedure followers, but they often follow the wrong procedures" (p. 157).

Thus, one of the goals in cognitive modeling is to identify cognitive rules that students use in the given learning domain and relate them to error responses. Several such attempts are reviewed in the following sections.

2.1.1 Cognitive Development

Siegler (1981) examined the cognitive development of understanding of the physical concept of torque in the balance scale task, in which students predict which side will go down for a

Table 2.1: Cognitive Rules for Siegler's Balance Scale Task

Rule	Description
1	Look at the weights only. Predict that the side with the greater weight will go down. If both sides have the same weight, the beam will balance.
2	Same as Rule 1 when the distances are the same. If the weights are equal but the distances are not, the side with the greater distance will go down. If the weights and distances are both equal, then predict that the beam will balance.
3	Same as Rule 2, but when one side has more (less) weight with a shorter (longer) distance than the other, children "muddle through," possibly leading to random guessing.
4	Compare the torques from both sides by computing the *product* of weight and distance on each side of the scale, and predict that the side with larger torque will go down. This is the correct rule, that is, it always produces a correct response.

given configuration of weights and their locations on a balance beam (see Figure 2.1). Based on his research and Piagetian theory of cognitive development, Siegler (1981) developed a cognitive model on what strategies students use to solve the balance scale tasks. His model consisted of the four cognitive rules shown in Table 2.1. Rules 1 through 3 are defective rules that typically appear in the course of development. Rule 4 is the correct rule indicating the full understanding of the torque concept; it always generates a correct response.

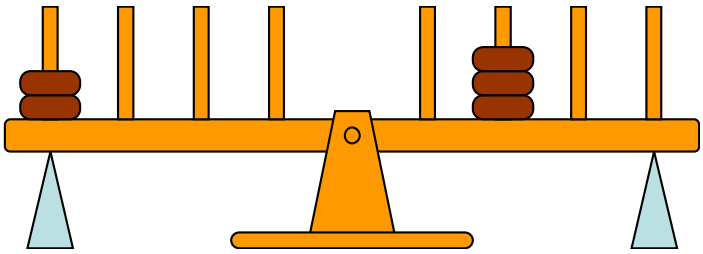

Figure 2.1: Sample balance scale problem

Each of these cognitive rules generates a particular response(s) to a given balance scale

problem. By systematically changing the configuration of weights and their locations on the balance scale, Siegler (1981) developed several types of items that distinguish all four rules by their expected response patterns. Table 2.2 shows 20 balance scale items and their expected responses for each of the four rules described in Table 2.1. These items are classified into four types of items: Distance (D), Conflict Weight (CW), Conflict Distance (CD), and Conflict Balance (CB). The distance items have equal weights on both sides but at different distances from the fulcrum. All conflict type items involve unequal weights at unequal distances from the fulcrum. For the conflict weight items, the correct response is the side with more weight. For the conflict distance items, the correct response is the side with weight placed at a greater distance from the fulcrum. For the conflict balance items, the correct response is "balance."

Expected responses that the four rules generate systematically differ by item type. In Table 2.2, for example, if students follow Rule 2, then their responses would be "right" for all CD items (i.e., items 11 through 15) and "left" for the other item types. Rule 4 is the correct rule, so responses different from those in the Rule 4 column are all incorrect responses within each row. Although the defective rules can be distinguished simply by whether they are correct or incorrect (with respect to the correct rule), even different incorrect responses relate to different defective rules as in items 11 through 20. In item 16, for example, response "left" implies Rules 1 through 3 while "right" implies Rule 3.

2.1.2 Facets and Facet Clusters

Facets and facet clusters (Minstrell, 2001; Minstrell & Kraus, 2007) describe student thinking and ideas in various learning situations and are intended to help teachers make instructional decisions. A facet of student thinking is defined as "a construction of one or more pieces of knowledge, reasoning, or procedures that students use to explain a situation or solve a problem" (Minstrell & Kraus, 2007, p. 1). Facets represent qualitatively different

Table 2.2: Expected Responses for the Balance Scale Items

Item	Type	Left	Right	Rule 1	Rule 2	Rule 3	Rule 4
1	D	(2, 2)	(1, 2)	B	L	L	L
2	D	(3, 2)	(2, 2)	B	L	L	L
3	D	(4, 3)	(2, 3)	B	L	L	L
4	D	(4, 2)	(3, 2)	B	L	L	L
5	D	(4, 1)	(2, 1)	B	L	L	L
6	CW	(2, 3)	(4, 1)	L	L	L/B/R	L
7	CW	(3, 2)	(4, 1)	L	L	L/B/R	L
8	CW	(3, 3)	(4, 2)	L	L	L/B/R	L
9	CW	(2, 4)	(4, 1)	L	L	L/B/R	L
10	CW	(1, 4)	(3, 1)	L	L	L/B/R	L
11	CD	(3, 1)	(1, 2)	R	R	L/B/R	L
12	CD	(4, 2)	(1, 4)	R	R	L/B/R	L
13	CD	(4, 1)	(1, 3)	R	R	L/B/R	L
14	CD	(3, 2)	(1, 3)	R	R	L/B/R	L
15	CD	(4, 3)	(2, 4)	R	R	L/B/R	L
16	CB	(1, 3)	(3, 1)	L	L	L/B/R	B
17	CB	(2, 3)	(3, 2)	L	L	L/B/R	B
18	CB	(3, 4)	(4, 3)	L	L	L/B/R	B
19	CB	(1, 4)	(2, 2)	L	L	L/B/R	B
20	CB	(2, 2)	(4, 1)	L	L	L/B/R	B

Note. D = Distance; CW = Conflict Weight; CD = Conflict Distance; CB = Conflict Balance; L = left; R = right; B = balance. Numbers in parentheses indicate the distance from the fulcrum and the number of weights, respectively.

Table 2.3: Facet Cluster for "Explaining Falling Bodies"

Facet	Description
340	Gravitational pull by earth on falling object and mass of object compensate for each other. The resistance by the medium through which the object is falling increases with speed and will decrease the rate of acceleration.
341	$(F_g - F_r)$/mass is the acceleration (instantaneous rate) of fall. With no resistance, near the earth, things fall, accelerating at about 9.8m/s^2.
342	Gravitational pull and mass compensate, but greater air resistance on the lighter object, making it fall behind.
...	...
349	Weight makes things fall. The more weight, the faster they fall.
349+	When you let things go, they fall.
349++	Things fall down.

Note. Excerpt from Table 3 in Minstrell (2001). Only the first and last three facets are shown.

understandings and are derived from empirical research on student thinking and from classroom observations by teachers. In addition, Minstrell (2001) emphasized the relevance to instruction; facets are described at an appropriate level of detail for instructional purposes so that each facet corresponds to a different remedial instruction.

A set of facets organized for a particular learning topic is called a facet cluster. Within a facet cluster, relevant facets form a sequence in an approximate order of development from least to most problematic. Table 2.3 shows an example of a facet cluster for how students explain falling bodies (Minstrell, 2001). The facet at the top of a cluster usually describes a learning goal, which may be specified in learning standards, and the following facets represent increasingly naive understandings, each of which would require a different instruction.

Facets provide a basis for designing assessment tasks and interpreting student responses. Assessment questions are designed to differentiate a small number of facets at a time, and multiple questions may be given to a student until the student's facet is diagnosed. Although he did not show the actual questions, Minstrell (2001) provided a guideline

for how to design questions to diagnose student facets. His recommendations included (a) making questioning contexts seductive to eliciting some problematic facets, (b) writing specific answers associated with relevant facets in multiple choice format or rubrics for facet diagnosis in the open-response format, and (c) leaving an option for learners to respond with an unanticipated answer. The second point states the importance of relating facets to responses as shown in Siegler's example.

2.1.3 Construct Maps

Similar to facets are *construct maps* (Briggs, Alonzo, Schwab, & Wilson, 2006), which are lists of descriptions of students' understandings at different levels of a certain construct to be measured. Each level "reflects a hierarchical stage through which students pass as they gain a qualitatively richer understanding about a given construct" (Briggs et al., 2006, p. 38).

Table 2.4: Construct Map for "Properties of Light"

Score Level	Description
4	Students conceive of light as a distinct entity in space. Understands the relationship between a light's source, its motion and path, the objects it encounters along the way and the effect it produces.
3	Student conceives of light as a distinct entity in space, traveling in a straight line. Lacks an understanding of how light interacts with objects.
2	Student understands limited cause and effect relationships between a light's source (bulb), state (brightness), and the effect it produces (patch of light).
1	Student identifies light solely with respect to its source or its effect. Light is not understood apart from its effects. Student defines light in relation to dark.

Note. Taken from Briggs et al. (2006, p. 38).

Table 2.4 shows an example of a construct map for the construct "Properties of Light." This construct map consists of four different levels of understanding. The state at the top

indicates the highest level of understanding with respect to the construct, and is given the highest score level ("4"). States that are given lower scores represent lower levels of understanding.

Development of construct maps is based on thorough investigation of various existing standards and empirical research as in facets and facet clusters, but construct maps and facet clusters are different in several aspects. First, descriptions in a construct map have a strict order along the intended unidimensional construct to represent developmental levels, even though they describe qualitatively different cognitive states. Second, construct maps are more oriented to item development and scoring than to instructions as for facets. As Briggs et al. (2006) described, construct maps provide a basis for designing ordered multiple choice items, in which each response option reflects each level of understanding specified in a construct map. Then, the same construct maps are used as scoring rubrics for those items (i.e., higher scores are assigned to responses corresponding to higher levels).

2.1.4 Cognitive Attributes

In some domains, student knowledge states can be decomposed into a set of basic skills, which is termed *attributes* in general. For example, Tatsuoka (1990) analyzed strategies that students use to solve mixed fraction subtraction problems and identified five basic attributes: (a) basic fraction subtraction, (b) simplify/reduce, (c) separate whole number from fraction, (d) borrow one from whole number to fraction, and (e) convert whole number to fraction (see Table 2.5). The cognitive model is then a set of these attributes, and each knowledge state is represented by a pattern of mastery/non-mastery of these attributes.

Test items were also characterized by these attributes and represented by an item-by-attribute matrix, termed the Q matrix (Tatsuoka, 1991). An element of a Q matrix is 1 if the attribute is required to successfully solve the item, and 0, otherwise. Thus, each row of a Q matrix indicates what attributes are required to successfully solve the item, from which

Table 2.5: Q Matrix for Mixed Fraction Subtraction Items

		Attribute				
Item	Text	1	2	3	4	5
4	$3\frac{1}{2} - 2\frac{3}{2}$	1	1	1	1	0
6	$\frac{6}{7} - \frac{4}{7}$	1	0	0	0	0
7	$3 - 2\frac{1}{5}$	1	1	1	1	1
8	$\frac{3}{4} - \frac{3}{8}$	1	0	0	0	0
9	$3\frac{7}{8} - 2$	1	0	1	0	0
10	$4\frac{4}{12} - 2\frac{7}{12}$	1	1	1	0	0
11	$4\frac{1}{3} - 2\frac{4}{3}$	1	1	1	1	0
12	$\frac{11}{8} - \frac{1}{8}$	1	1	0	0	0
...	...			...		

Note. Attributes represent (1) Basic fraction subtraction; (2) Simplify/reduce; (3) Separate whole number from fraction; (4) Borrow one from whole number to fraction; (5) Convert whole number to fraction. The table is excerpted from Mislevy (1995, p. 58). Only the first eight items are shown.

one can predict whether a correct or incorrect response is expected for a given attribute pattern. Table 2.5 shows an example of a Q matrix derived from Tatsuoka's (1990) task analysis. For example, mastering only attributes 1 and 2 leads to correct responses to items 6, 8, and 12 and incorrect responses to the other items.

Cognitive attribute modeling is becoming more popular in the design of CDA, because such modeling can produce attribute (skill) mastery profiles, which are appealing for both summative and formative purposes. Such modeling also provides an efficient way to represent a large number of knowledge states by combinations of a smaller number of attributes.

Although the cognitive modeling examples reviewed above are based on different assumptions and representations of cognitive processes, a common feature is that they all make clear how different cognitive states lead to different error patterns. This should be reflected in the design of assessment tasks and interpretation of student responses.

2.2 Observation in CDA: Diagnostic Items

Observation in the assessment triangle sets up a context or situation in which student behaviors relevant to the target construct are elicited and recorded. This involves the design of assessment tasks and determination of what kinds of student behaviors are taken as responses on those tasks.

One of the concerns in the observation part is to determine the type of responses. While theoretical and technological advances in educational measurement have broadened the possibility of implementing more complex (and computerized) assessments such as essay or open-ended questions, measuring response time, and motion/solution tracking (e.g., Williamson et al., 2006), the multiple choice format is still one of the most widely used test formats, especially in large-scale assessments. In a typical multiple choice item, several response options are presented following the item stem, and each examinee chooses one response option as the answer. One of the response options is a correct answer, and the others are incorrect.

Multiple choice items are generally believed to measure low-level cognitive skills (e.g., mere recall of facts) or abilities that are defined under general specifications (e.g., "reading ability"). In the usual development of multiple choice items, incorrect response options are "chosen to reflect a range of typical incorrect answers observed in actual student responses," or selected from lists of alternatives generated by test development experts (Briggs et al., 2006, p. 34) without much consideration of how students reach those incorrect answers. Resnick and Resnick (1992) criticized the use of multiple choice items for not measuring reasoning processes on the ground that they discourage students from actively engaging in assessment tasks and merely facilitate guessing of the right answers.

However, multiple choice items have the potential to tap higher-level cognitive processes so that they can provide diagnostic information if carefully designed (National Research Council, 2001, p. 194; Osterlind, 1998, p. 163). Mislevy (1993) mentioned the possibility

and necessity of distinguishing among incorrect responses in multiple choice items; considerations of incorrect responses can provide a way to stronger inferences about student knowledge states. In addition, strengths of the multiple choice format in general include its flexibility for accommodating a variety of contents and valid test score interpretations (Osterlind, 1998, chap. 5). Also, the multiple choice format saves time and cost required for their development, administration, and scoring, and generally maintains higher objectivity and reliability than open-ended formats (Briggs et al., 2006; Haladyna, 2004; Sadler, 1998).

Recently, researchers have suggested that *diagnostic items* be used in formative classroom assessment (Ciofalo & Wylie, 2006; Wylie & Ciofalo, 2006; Wylie & Wiliam, 2007). Ciofalo and Wylie (2006) defined a diagnostic item as one in which each incorrect response "not only provides information that a student does not clearly understand a particular topic; it also provides specific insight into *what it is that the student does not understand*—in other words, the nature of his/her misconceptions" (para. 6). More specifically, diagnostic items "are single, multiple choice questions connected to a specific content standard or objective. They have one or more answer choices that are incorrect but related to common student misconceptions regarding that standard or objective" (Ciofalo & Wylie, 2006, para. 9).

Prior to the recent rise of diagnostic items, several psychologists had written about the same idea (Bart & Williams-Morris, 1990; Bart et al., 1994; Powell, 1968, 1977; Powell & Shklov, 1992). In particular, Bart and Williams-Morris (1990) introduced an analytic method of diagnostic items termed refined digraph analysis, which is a method to visualize and quantify the diagnostic capability of diagnostic items. Bart and Williams-Morris (1990) proposed a response-by-rule matrix to make explicit how cognitive rules relate to different responses in a diagnostic item. Suppose that there are J multiple choice items and L cognitive rules, and item j has K_j response options. The response-by-rule matrix for item

j is a $K_j \times L$ matrix and denoted by $\mathbf{A}_j = \{a_{kl}^{(j)}\}$, where

$$a_{kl}^{(j)} = \begin{cases} 1, & \text{if the } l\text{th rule generates the } k\text{th response for item } j, \\ 0, & \text{otherwise.} \end{cases} . \tag{2.1}$$

For example, the response-by-rule matrix for item 16 in Table 2.2 is

$$\mathbf{A}_{16} = \begin{array}{c} \\ \text{Left} \\ \text{Balance} \\ \text{Right} \end{array} \begin{array}{c} \begin{array}{cccc} \text{Rule 1} & \text{Rule 2} & \text{Rule 3} & \text{Rule 4} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}. \tag{2.2}$$

In addition to explicating the relationship between cognitive rules and responses, several indices that summarize different aspects of diagnostic capability of an item are readily available from the response-by-rule matrix. Bart and Williams-Morris (1990) introduced two such indices: response interpretability and response discrimination. *Response interpretability* (RI) refers to the degree to which each response is interpreted by a rule, and is defined as the proportion of responses in an item that are associated with at least one cognitive rule. *Response discrimination* (RD) refers to the degree to which each response is interpreted by only one rule, and is defined as

$$\text{RD}_j = \frac{1}{K_j} \sum_{k=1}^{K_j} \text{RD}_{jk}, \qquad \text{where } \text{RD}_{jk} = \begin{cases} \left(\sum_{l=1}^{L} a_{kl}^{(j)} \right)^{-1}, & \text{if } \sum_{l=1}^{L} a_{kl}^{(j)} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

Both indices range from 0 to 1 with 1 denoting the maximum diagnostic power. We have $\text{RI}_{16} = 1.00$ and $\text{RD}_{16} = .61$ for the above item. These figures indicate that all responses are interpreted by at least one rule, but on average more than one rule is associated with each response.

As a further psychometric formulation of diagnostic items, Bart et al. (1994) presented the concept of a *semi-dense* item. Semi-density means an ideal property of a diagnostic item in which each response option corresponds to one and only one cognitive rule. It is not always possible to achieve semi-density in practice. Nonetheless, the response-by-rule matrix provides a way to evaluate diagnostic capability of items *prior to* collecting data (Nichols, 1994). As a general guide, responses should be made to differentiate cognitive rules as much as possible (Wylie & Wiliam, 2007).

Diagnostic items have mainly two implications for CDA. First, diagnostic items make CDA more *efficient* than simply scoring items as either correct or incorrect. Different incorrect responses imply different cognitive states in diagnostic items, but such information would be lost when one only considers whether the response is correct or incorrect, which is a common practice even in the current CDA design. In more efficient assessments, one can reach a diagnosis with a fewer number of diagnostic items (i.e., shorter testing time) while maintaining the same level of diagnostic accuracy. Shorter testing time means that teachers can use more time for instruction and students also have more time to do their own work This is an important consideration especially when the CDA is embedded in regular instruction or an automated learning environment. Also, shorter tests also save cost for administration and minimize the unnecessary exposure of items.

Second, it is possible to "upgrade" multiple choice items in existing large-scale assessments to diagnostic items by replacing distractors by rule-interpretable responses. Although its feasibility would likely depend on how the target construct is defined (e.g., a broadly defined construct can be associated with too many different sets of cognitive rules, making the test difficult to diagnose all of those rules), this would lead to assessments that can provide diagnostic information while maintaining high objectivity and reliability and serving their original summative purposes (Briggs et al., 2006). Serving multiple purposes is one of the characteristics expected for current and future educational assessments as rec-

ommended by the National Research Council (2001). Also, considering rule-interpretable responses would reduce arbitrary selection of response options by item writers. It facilitates more systematic generation of response options from the current, state-of-art nature of item writing. In fact, Graf and Ohls (2006) proposed a method to automatically generate response options based on cognitive rules.

2.3 Interpretation in CDA: Cognitive Diagnostic Psychometric Models

Interpretation in the assessment triangle specifies a mechanism that diagnoses student cognitive states from observations on assessment tasks. For this purpose, psychometric models provide a formal way to synthesize evidence from multiple observations on assessment tasks to make a cognitive diagnosis (Mislevy et al., 2003). Psychometric models that are used for CDA are called *cognitive diagnostic psychometric models* (CDPMs). There are recent comprehensive reviews about CDPMs (Fu, 2005, part I; Fu & Li, 2007; Junker, 1999; Rupp, 2007). However, there is variability as to the definition of a CDPM, and models included in these reviews also varied. For example, Rupp (2007) provided a detailed definition of CDPMs as follows:

> CDPMs are probabilistic confirmatory multidimensional latent-variable models with a complex loading structure. They are suitable for modeling categorical response variables and contain categorical latent predictor variables that generate latent classes. They enable multiple criterion-referenced interpretations and feedback for diagnostic purposes that are referenced to a cognitively-grounded theory of response processes at a fine grain size. (p. 8)

This definition resulted in models with discrete latent variables.

In contrast, Fu (2005, part I) defined CDPMs as "statistical models developed to determine each examinee's diagnostic status with respect to cognitive components and/or each item's measurement of those cognitive components" (p. 3). With this broader definition, his review eventually included more than 60 psychometric models, most of which are variations of multivariate item response theory (IRT) models that include univariate IRT models and latent class models as special cases.

The review in this section limits its scope to CDPMs that can be used for diagnostic items, that is, models which explicitly express how student discrete cognitive states lead to different observed responses. This results in the latent class model (LCM; Goodman, 1974; Lazersfeld & Henry, 1968) and its variations, in which cognitive states are represented by a single or multiple latent categorical variables. Most LCMs in this category were developed for dichotomous test items (i.e., items scored as either correct or incorrect). In the following, LCMs for dichotomous items are reviewed first, and then their recent extensions to multiple choice items are presented.

2.3.1 Models for Dichotomous Responses

Suppose that there are J dichotomous items and L cognitive states. Let a response to item j denoted by $X_j \in \{0, 1\}$, $j = 1, \ldots, J$, and a vector of responses to the J items by $\mathbf{x} = (x_1, \ldots, x_J)$. Student cognitive states are represented by a latent categorical variable $\phi \in \{\phi_1, \ldots, \phi_L\}$, where latent state ϕ_l corresponds to the lth cognitive state.

LCM is a model about the probability of item responses $\mathbf{x}$, which systematically varies as a function of latent state ϕ. In the most basic (unconstrained) form of LCM, the probability of a correct response $X_j = 1$ to item j (termed the item response probability) is represented by the conditional probability given latent state, ϕ_l: $\pi_{jl} = P(X_j = 1|\phi = \phi_l)$. The latent state follows a marginal multinomial probability distribution: $\omega_l = P(\phi = \phi_l)$, $l = 1, \ldots, L$. It is assumed that the L latent states are mutually exclusive and exhaustive,

and thus $\sum_{l=1}^{L} \omega_l = 1$. Another fundamental assumption is local independence, which implies that item responses are independent of each other given a state. Thus, if $\phi = \phi_l$ is given, the probability of response $\mathbf{x}$ to J items is given by the product of item response probabilities:

$$P(\mathbf{x}|\phi = \phi_l) = \prod_{j=1}^{J} {\pi_{jl}}^{x_j} (1 - \pi_{jl})^{1-x_j}. \tag{2.4}$$

Finally, the probability of observing responses to J items is a mixture of Equation 2.4 with respect to the latent state probability ω_l:

$$P(\mathbf{x}) = \sum_{l=1}^{L} \omega_l \prod_{j=1}^{J} {\pi_{jl}}^{x_j} (1 - \pi_{jl})^{1-x_j}. \tag{2.5}$$

The product of Equation 2.5 for all observed response vectors $\mathbf{x}$ constitutes the likelihood function for the model parameters π_{jl} and ω_l. The above model has JL free item parameters (π_{jl}) and $L-1$ marginal state probability parameters (ω_l). Equations 2.4 and 2.5 represent the fundamental probabilistic structure common to all models reviewed in this section.

Diagnosing a student's cognitive state is equivalent to estimating the student's latent state ϕ from his/her observed response vector $\mathbf{x}$. This is made through the posterior state probabilities. After observing responses $\mathbf{x}$, the posterior probability of state ϕ_l is

$$\omega_l'(\mathbf{x}) = P(\phi = \phi_l|\mathbf{x}) \tag{2.6}$$

$$= \frac{\omega_l^{(0)} \prod_{j=1}^{J} {\pi_{jl}}^{x_j} (1 - \pi_{jl})^{1-x_j}}{\sum_{l'=1}^{L} \omega_{l'}^{(0)} \prod_{j=1}^{J} {\pi_{jl}}^{x_j} (1 - \pi_{jl})^{1-x_j}}, \qquad l = 1, \ldots, L, \tag{2.7}$$

where $\omega_l^{(0)}$ is a prior probability for state ϕ_l. Cognitive diagnosis is usually made by picking the state with the largest posterior probability: $\hat{\phi} = \arg\max_l \omega_l'(\mathbf{x})$.

The unconstrained LCM is often used for exploratory purposes rather than directly used

as a model for CDA. For example, there have been many attempts to identify strategies to solve Siegler's balance scale tasks using unconstrained LCMs (Boom, Hoijtink, & Kunnen, 2001; Boom & ter Laak, 2007; Jansen & van der Maas, 1997, 2002). In the CDA context, LCM is applied in a more confirmatory manner so that it reflects the underlying cognitive model. More specifically, some constraints are imposed on item response probabilities π_{jl} so that they represent the correspondence between responses and cognitive states specified by the cognitive model.

The State Mastery Model

The state mastery model (Falmagne, 1989; Macready & Dayton, 1977, 1980) is a LCM for dichotomous responses, but it differs from the basic LCM in that each state ϕ_l designates an expected response (correct or incorrect) for each item, and item response probabilities are structured accordingly.

More specifically, each state ϕ_l specifies an expected response pattern $\mathbf{v}_l = (v_{1l}, \ldots, v_{Jl})$, $v_{jl} \in \{0, 1\}$ for the J items. In other words, $\mathbf{v}_l$ represents which items are answered correctly in state ϕ_l. In Table 2.2, for example, Rule 1 predicts correct responses to items 6 through 10 and incorrect responses to the other items. Thus, the expected response pattern for Rule 1 is $\mathbf{v}_1 = (0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$. Expected response patterns for the other cognitive rules, $\mathbf{v}_2, \ldots, \mathbf{v}_4$, can be specified in the same manner. In particular, Rule 4 is the "correct" rule which leads to a correct response to all items, so its expected response pattern is a vector of 1s: $\mathbf{v}_4 = (1, \ldots, 1)$.

The item response probability in the state mastery model is given by

$$\pi_{jl} = {\alpha_{jl}}^{(1-v_{jl})}(1 - \beta_{jl})^{v_{jl}}, \tag{2.8}$$

where α_{jl} is the false positive rate, which is the probability of a correct response to item j when an incorrect response is expected in state ϕ_l, and β_{jl} is the false negative rate,

which is the probability of an incorrect response to item j when a correct response is expected in state ϕ_l. In the above example, the response probabilities of the 20 items are $\{\pi_{j1}\} = (\alpha_{1,1}, \ldots, \alpha_{5,1}, 1-\beta_{6,1}, \ldots, 1-\beta_{10,1}, \alpha_{11,1}, \ldots, \alpha_{20,1})$ for Rule 1, $\{\pi_{j4}\} = (1-\beta_{1,4}, 1-\beta_{2,4}, \ldots, 1-\beta_{20,4})$ for Rule 4, and so forth. If the items are good indicators of cognitive rules, then both α and β parameters are expected to be small (i.e., close to 0), and the resulting item response probability is close to 0 where an incorrect response is expected and close to 1 where a correct response is expected.

Equation 2.8 is a mere reparameterization of the unconstrained LCM, because the number of free parameters for each item is the same as that of the unconstrained LCM. Further constraints, such as state independence ($\alpha_{jl} = \alpha_j$ and $\beta_{jl} = \beta_j$ for all l), item independence ($\alpha_{jl} = \alpha_l$ and $\beta_{jl} = \beta_l$ for all j), and equal false rates ($\alpha_{jl} = \beta_{jl}$ for all j and l), may be imposed for the construction of a more parsimonious model (Macready & Dayton, 1980). For the state independence model, for example, the number of free item parameters is $2J$, which is a substantial reduction from the unconstrained model.

Binary Skills Models

Binary skills models build upon cognitive attribute modeling, in which each latent state is represented by a combination of multiple latent indicator variables $\boldsymbol{\phi}_l = (\phi_{l1}, \ldots, \phi_{lM})$, $\phi_{lm} \in \{0, 1\}$. $\phi_{lm} = 1$ indicates possession or mastery of the mth attribute in state $\boldsymbol{\phi}_l$, so the vector $\boldsymbol{\phi}_l$ indicates the set of attributes that an examinee has mastered (thus, there are $L = 2^M$ possible states if there are M attributes). Each item is characterized by the corresponding attribute set $\mathbf{q}_j = (q_{j1}, \ldots, q_{jM})$, $q_{jm} \in \{0, 1\}$, where $q_{jm} = 1$ indicates that the mth attribute is *required* to solve item j. Patterns of required attributes over items are summarized by a Q matrix of size $J \times M$ such that $\mathbf{Q} = [\mathbf{q}_1 \cdots \mathbf{q}_J]^T$ (Tatsuoka, 1991). Table 2.5 shows a Q matrix for mixed fraction subtraction problems. For example, the $\mathbf{q}$ vector for item 4 is $\mathbf{q}_4 = (1, 1, 1, 1, 0)$, $\mathbf{q}_6 = (1, 0, 0, 0, 0)$ for item 6, and so forth.

Binary skills models have several variations, depending on how attributes jointly specify item response probabilities. The deterministic input noisy "and" gate (DINA) model (Haertel, 1984, 1989; Maris, 1999) assumes that all attributes required by an item must be mastered in order for an examinee to answer correctly on that item. In other words, missing any of the required attributes is equivalent to missing all of the required attributes, leading to an incorrect response. The item response probability in the DINA model is

$$\pi_{jl} = (1 - s_j)^{\xi_j(\boldsymbol{\phi}_l)} g_j{}^{1-\xi_j(\boldsymbol{\phi}_l)}, \tag{2.9}$$

where $\xi_j(\boldsymbol{\phi}_l) = \prod_{m=1}^{M} \phi_{lm}^{q_{jm}}$ is the latent score that indicates whether all attributes required for item j are mastered or not in state $\boldsymbol{\phi}_l$, $s_j = P(X_j = 0|\xi_j(\boldsymbol{\phi}_l) = 1)$ is the false negative rate for item j, and $g_j = P(X_j = 1|\xi_j(\boldsymbol{\phi}_l) = 0)$ is the false positive rate for item j.

For example, the state of an examinee who mastered attributes 1 and 2 in Table 2.5 is denoted by $\boldsymbol{\phi}' = (1, 1, 0, 0, 0)$. Latent scores of this examinee for the eight items shown in Table 2.5 are $(\xi_4(\boldsymbol{\phi}'), \xi_6(\boldsymbol{\phi}') \ldots, \xi_{12}(\boldsymbol{\phi}')) = (0, 1, 0, 1, 0, 0, 0, 1)$ (i.e., a correct response is expected for items 6, 8, and 12). The corresponding item response probabilities are then $(g_4, 1 - s_6, g_7, 1 - s_8, g_9, g_{10}, g_{11}, 1 - s_{12})$.

The DINA model is equivalent to the state mastery model in that whether all required attributes are mastered or not completely determines the expected response for each item through the latent score. More specifically, the vector of latent scores $\{\xi_j(\boldsymbol{\phi}_l)\}$, $j = 1, \ldots, J$, has the same function as the expected response vector $\mathbf{v}_l$ in the state mastery model. The item response probability then equals either the false positive rate (g_j, which is equivalent to α_j in the state mastery model with the state independence constraints) or the true positive rate ($1 - s_j$, which is equivalent to $1 - \beta_j$), depending on whether a correct or incorrect response is expected. The number of free item parameters in this case is $2J$, which equals the number of free item parameters in the state mastery model with state independence constraints.

Other models in this category include the noisy input deterministic "and" gate (NIDA) model (Junker & Sijtsma, 2001) and the reparameterized unified model (RUM; Hartz, 2002; Roussos et al., 2007). The NIDA model relaxes the assumption of the DINA model that a correct response requires all necessary attributes to be mastered by taking into account the difficulty of applying each attribute. RUM is a reparameterized version of the NIDA model, but also has a Rasch item response function (i.e., the one-parameter logistic model) attached to each item response probability in order to account for a general or residual ability that is not specified in the Q matrix.

There are other methods that deal with binary attributes, such as the rule space analysis (Tatsuoka, 1983, 1985, 1990) and the attribute hierarchy method (Leighton, Gierl, & Hunka, 2004; Gierl, Leighton, & Hunka, 2007). These are classification methods rather than psychometric models, so their details are not described here.

2.3.2 Models for Multiple Choice Responses

In order to take full advantage of diagnostic items, the models described in the previous section need to be extended to accommodate multiple choice responses. Response options in a multiple choice item do not usually have a specific order, so the natural choice is a model for nominal categorical responses.

In the IRT context, a number of models have been proposed for multiple choice items, such as the nominal response model (Bock, 1972), multiple choice models (Samejima, 1979; Thissen & Steinberg, 1984; Thissen, Steinberg, & Fitzpatrick, 1989), and the distractor rejection models (Revuelta, 2004). These models were intended to improve ability estimation (i.e., incorrect responses also bear information regarding examinees' ability; Thissen & Steinberg, 1984) and/or to examine characteristics of items more closely by modeling the behavior of incorrect responses (e.g., equivalent and informative distractors; Samejima, 1988). In these models, however, incorrect responses are not given cognitive interpretations

and no qualitative difference is addressed. In the context of CDA, little development of CDPMs for nominal categorical responses has been seen so far, because nominal response items are "rarely used in cognitively diagnostic tests" (Fu, 2005, part I, p. 121).

Extension of the basic LCM (Equation 2.5) to multiple choice responses is straightforward. A response to item j now takes one of the K_j possible values: $X_j \in \{1, \ldots, K_j\}$. The item response probability becomes the probability of response $X_j = k$ in item j given state ϕ_l, and is denoted by $\pi_{jkl} = P(X_j = k|\phi = \phi_l)$, $\sum_{k=1}^{K_j} \pi_{jkl} = 1$. Under the assumptions of mutually exclusive and exhaustive latent states and local independence, the probability of observing response vector $\mathbf{x}$ is

$$P(\mathbf{x}) = \sum_{l=1}^{L} \omega_l \prod_{j=1}^{J} \prod_{k=1}^{K_j} \pi_{jkl}^{I(x_j=k)}, \tag{2.10}$$

where $I(\cdot)$ is the indicator function, whose value is 1 if its argument is true, and 0, otherwise. The above model has $\sum_{j=1}^{J}(K_j - 1)L$ free item parameters (π_{jkl}) and $L - 1$ marginal state probability parameters (ω_l). Again, desired is a model that explicitly represents the correspondence between cognitive states and their expected responses.

LCM for Diagnostic Items: Extension of the State Mastery Model

Kato (2008b) proposed an application of polytomous linear logistic LCM (Formann, 1992) to diagnostic items (LCM-DI). In one of his models, the item response probability takes the form

$$\pi_{jkl} = \frac{\exp(\eta_{jkl})}{\sum_{k'=1}^{K_j} \exp(\eta_{jk'l})}, \tag{2.11}$$

where η_{jkl} is the (multinomial) logit of π_{jkl} and can take any real value. Furthermore, η_{jkl} is constrained as follows:

$$\eta_{jkl} = \begin{cases} \alpha_{jl}, & \text{if state } \phi_l \text{ generates response } k \text{ for item } j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.12}$$

Which response each cognitive state generates is made explicit by the response-by-rule matrix (Equation 2.1). The above constraint is equivalent to assigning α_{jl} to the logit of response k for which $a_{kl}^{(j)} = 1$ and 0 to the logit of response k' for which $a_{k'l}^{(j)} = 0$.

For example, consider the response-by-rule matrix of item 16 in Table 2.2, which is given in Equation 2.2. Let $\boldsymbol{\eta}_j$ denote a $K_j \times L$ matrix of logits whose (k, l) element is η_{jkl}. Then, for this item, we have

$$\boldsymbol{\eta}_{16} = \begin{array}{c} \\ \text{Left} \\ \text{Balance} \\ \text{Right} \end{array} \begin{array}{c} \begin{array}{cccc} \text{Rule 1} & \text{Rule 2} & \text{Rule 3} & \text{Rule 4} \end{array} \\ \begin{bmatrix} \alpha_{16,1} & \alpha_{16,2} & \alpha_{16,3} & 0 \\ 0 & 0 & \alpha_{16,3} & \alpha_{16,4} \\ 0 & 0 & \alpha_{16,3} & 0 \end{bmatrix} \end{array}. \tag{2.13}$$

All responses are assumed to occur with the same probability for Rule 3, so $\alpha_{16,3}$ is not identifiable. In this case, $\alpha_{16,3}$ is simply set to 0.

The above logits are transformed to item response probabilities by using Equation 2.11. For example, item response probabilities under Rule 1 are $\pi_{16,11} = e^{\alpha_{16,1}}/(e^{\alpha_{16,1}}+e^0+e^0) = e^{\alpha_{16,1}}/(e^{\alpha_{16,1}}+2)$ for response "left," and $\pi_{16,21} = \pi_{16,31} = e^0/(e^{\alpha_{16,1}}+e^0+e^0) = 1/(e^{\alpha_{16,1}}+2)$ for responses "right" and "balance." We expect that $\alpha_{jl} > 0$ because the expected response should have a higher probability than the other responses. Also, this model assumes that responses that are *not* expected are chosen with equal probability, so logits are set to 0 for those responses. As a result, the number of free item parameters in this model is $JL - \sum_{j=1}^{J}(L - L_j) = \sum_{j=1}^{J} L_j$, where L_j $(\leq L)$ is the number of states to which a free

logit parameter (α_{jl}) is assigned in item j.

This model can be applied to dichotomous data as well (consider that the $\boldsymbol{\eta}_j$ matrix has only two rows for correct and incorrect responses, and in each column α_{jl} is assigned to the expected response, which is either correct or incorrect). If this is the case, the above model is equivalent to the state mastery model. Thus, LCM-DI is regarded as an extension of the state mastery model.

The Multiple Choice DINA Model

An extension of the DINA model to multiple choice items (MC-DINA) was considered by de la Torre (2009). In MC-DINA, different responses are expected by different configurations of attribute vector $\boldsymbol{\phi}_l$.

In de la Torre's (2009) formulation, the $\mathbf{q}_j$ vector for each item, which constituted a row in the original Q matrix, now becomes a $K_j \times M$ matrix, $\mathbf{Q}_j$, which is similar to the response-by-rule matrix. Let $\mathbf{q}_{jk} = (q_{jk1}, \ldots, q_{jkM})$, $q_{jkm} \in \{0, 1\}$, denote the kth row of $\mathbf{Q}_j$. Then, $q_{jkm} = 1$ if attribute m is required for response k, and $q_{jkm} = 0$, otherwise. The K_j response options in item j divide all 2^M possible attribute patterns (i.e., $\boldsymbol{\phi}_l$, $l = 1, \ldots, 2^M$) into $G_j + 1$ groups, where G_j ($\leq K_j$) is the number of response options with a distinctive non-null $\mathbf{q}_{jk}$ vector.

de la Torre (2009) considered a latent group classification for each item, which is defined as $g_j(\boldsymbol{\phi}_l) = \arg\max_{k'}\{\boldsymbol{\phi}_l'\mathbf{q}_{jk'} | \boldsymbol{\phi}_l'\mathbf{q}_{jk'} = \mathbf{q}_{jk'}'\mathbf{q}_{jk'}\}$ for item j and state $\boldsymbol{\phi}_l$. $g_j(\boldsymbol{\phi}_l)$ determines the expected response for item j given $\boldsymbol{\phi}_l$ as the latent score $\xi_j(\boldsymbol{\phi}_l)$ does in the DINA model. The expected response for a given state $\boldsymbol{\phi}_l$ is response k' for which (a) non-zero attributes in $\boldsymbol{\phi}_l$ include all attributes required by response k', and (b) the number of matches of non-zero attributes between $\boldsymbol{\phi}_l$ and $\mathbf{q}_{jk'}$ is largest among those responses satisfying (a). $g_j(\boldsymbol{\phi}_l)$ can be 0 if there is no match between $\boldsymbol{\phi}_l$ and $\mathbf{q}_{jk'}$. This implies that such a state does not predict any specific response for that item. Finally, the item response probability

in the MC-DINA model is defined by

$$\pi_{jkl} = P(X_j = k|\boldsymbol{\phi} = \boldsymbol{\phi}_l) = P(X_j = k|g_j(\boldsymbol{\phi}_l) = k') = \pi_{jkk'}, \tag{2.14}$$

where k' takes one of the G_j+1 possible values in a subset of $\{0, \ldots, K_j\}$ and $\sum_{k=1}^{K_j} \pi_{jkk'} = 1$.

The above formulation of the item response probability implies that two different states $\boldsymbol{\phi}_l$ and $\boldsymbol{\phi}_{l'}$ lead to the same item response probability if their expected responses are the same (i.e., $g_j(\boldsymbol{\phi}_l) = g_j(\boldsymbol{\phi}_{l'}) = k'$). de la Torre (2009) presented an example of modeling a mixed fraction subtraction item: $2\frac{4}{12} - \frac{7}{12} =$? The item has four response options and its $\mathbf{Q}_j$ matrix is

$$\mathbf{Q}_j = \begin{array}{c} \\ 2\frac{3}{12} \\ 2\frac{1}{4} \\ 1\frac{9}{12} \\ 1\frac{3}{4} \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}, \tag{2.15}$$

where the columns represent the five attributes shown in Table 2.5. The rows of Equation 2.15 are all non-zero and different from each other (i.e., $G_j = 4$), so the $2^5 = 32$ states are divided into five groups by the four responses of this item. For example, the expected response of state $\boldsymbol{\phi} = (1, 1, 0, 0, 0)$ is 2 (i.e., $2\frac{1}{4}$); this state has all attributes required for responses 1 and 2, but the number of matches is larger with response 2. Similarly, the expected responses of states such as $(1, 1, 1, 0, 0)$ and $(1, 1, 1, 0, 1)$ are also 2. Thus, these states have the same item response probabilities: $(\pi_{j12}, \pi_{j22}, \pi_{j32}, \pi_{j42})$. In contrast, the expected responses of states such as $(0, 0, 1, 0, 0)$ and $(0, 0, 0, 0, 1)$ are 0, because these states include no attribute pattern in $\mathbf{Q}_j$. The corresponding item response probabilities of these

states are $(\pi_{j10}, \pi_{j20}, \pi_{j30}, \pi_{j40})$.

We expect that the probability of an expected response is higher than those of non-expected responses within each item, i.e., $\pi_{jkk} > \pi_{jkk'}$ for $k \neq k'$. The above model has $\sum_{j=1}^{J}(K_j - 1)(G_j + 1)$ free item parameters, but further constraints may be imposed to reduce the number of parameters. For example, the probability of responses that are not expected may be set equal, that is, $\pi_{jkk'} = (1 - \pi_{jkk})/(K_j - 1)$ for all $k \neq k'$. Item response probabilities for states for which $g_j(\boldsymbol{\phi}_l) = 0$ may also be set equal (de la Torre, 2009). If these constraints are placed, the number of free item parameters reduces to $\sum_{j=1}^{J} G_j$.

The MC-DINA model simplifies to the DINA model if responses are dichotomous. Thus, it is a generalization of the DINA model for multiple choice items (de la Torre, 2009).

2.3.3 Comparison of CDPMs

A common feature of the above CDPMs is that the correspondence between cognitive states and their expected responses is explicitly embedded in them through constraints on item parameters. Meanwhile, main differences in the above CDPMs are (a) whether latent states are represented by a single latent categorical variable (as in the state mastery model) or multiple latent indicator variables (binary skills), and (b) whether item responses are treated as dichotomous or polytomous (multiple choice).

Choice between a single latent state variable and binary skills should be guided by the cognitive model as the construct-centered approach entails. As mentioned above, cognitive attribute modeling is becoming more popular in CDA and so are the binary skills models. The fundamental requirement for cognitive attribute modeling is that the construct be decomposed into basic cognitive attributes in a meaningful manner as in the mixed fraction subtraction example in Table 2.5. If this is possible, binary skills models can provide diagnostic information at a finer grain size than the state mastery model. In contrast, models with a single latent categorical variable is preferable when cognitive states represent proce-

dural or conceptual differences (i.e., cognitive rules), which may not allow the systematic decomposition into attributes. Even though cognitive rules may consist of several skills which can be listed as cognitive attributes, it is more appropriate to treat each cognitive rule as a unit state when, for example, the order or way of applying those skills matters.

In this vein, cognitive attribute modeling may not be realistic especially when used for multiple choice items. This is because specifying expected responses usually requires more procedural definition of how those responses are generated, while response specificity of cognitive attributes is not as clear as that of cognitive rules. For example, the formulation of the $\mathbf{Q}_j$ matrix in Equation 2.15 implicitly assumed that the required attributes are applied in the following order: $2\frac{4}{12} - \frac{7}{12} =$ (a) borrow from whole (attribute 4) to obtain $1\frac{16}{12} - \frac{7}{12}$, (b) basic fraction subtraction (attribute 1) to obtain $1\frac{9}{12}$, and (c) simply/reduce (attribute 2) to reach the correct answer $1\frac{3}{4}$ (de la Torre, 2009). However, this order is not reflected in the formulation at all, while the same set of attributes applied in a different order may lead to a different response. Consider the state $\boldsymbol{\phi}' = (1, 0, 0, 1, 0)$ (i.e., having attributes 1 and 4). From Equation 2.15 the expected response for this state is $1\frac{9}{12}$ (response 3), but this is true only when these attributes are applied in the order $4 \rightarrow 1$. If these attributes are applied in the reverse order ($1 \rightarrow 4$), then a different response would result (e.g., $1\frac{3}{12}$). In the latter case, the student knows the two attributes but does not know when and how to apply them. Cognitive attribute modeling would misdiagnose this case as not having required attributes, while in the single state variable models this difference is simply represented by two different states (i.e., "$1 \rightarrow 4$" and "$4 \rightarrow 1$"). This weak response specificity of cognitive attributes may limit the applicability of the MC-DINA model.

The purpose of assessment and reporting requirements also affect the model choice. All of the above models provide a posterior probability for each state ϕ_l based on observed item responses (Equation 2.7). On the one hand, the binary skills models not only diagnose a skill mastery pattern ($\boldsymbol{\phi}_l$) but also can provide a skill mastery profile as a set of attribute

probabilities, each of which represents the posterior probability of having attribute m. Users of binary skills models are often more concerned with the degree of mastery of individual cognitive skills (as a skill mastery profile) rather than with a specific cognitive state diagnosis. Skill mastery profiles can be used for both summative and formative purposes; they directly indicate which skills have not been mastered as a result of an instruction (summative), or which skills should be taught in the subsequent instruction (formative). On the other hand, models with a single latent categorical variable only provide a cognitive state diagnosis (ϕ_l). Although the grain size of such information is not as fine as skill profiles, these models are of practical use if (a) the cognitive states involve procedural and strategic differences that the binary skills models cannot capture, and (b) they are tied to specific remedial instructions (as in the design of facets). Thus, models with a single latent categorical variable are intended for more diagnostic and instructional use than binary skills models.

Whether item responses are treated as dichotomous or polytomous depends on the observation specification in CDA. Dichotomous models have generality in the sense that they can be applied not only to the multiple choice items but also to other item types such as constructed response items as long as they are scored dichotomously. If diagnostic items are to be used, then the models for multiple choice responses are preferred because these models can derive more diagnostic information from each single item than when items are scored dichotomously. Both the simulation study by de la Torre (2009) and the real data example by Kato (2008a) indicated that diagnostic efficiency was substantially improved by taking into account information from incorrect responses.

In addition to the above considerations, model complexity affects the number of items and examinees required for parameter estimation and thus draws practical attention. In general, more examinees are required as the number of parameters increases, which is a function of the number of items and the number of latent states. The sample size should be

large enough to ensure that a substantial number of response patterns that closely match expected response patterns are observed so that item parameters related to each state can be estimated with a certain level of accuracy.

Both the state mastery model with state independent constraints and the DINA model have $2J$ free item parameters, which only depend on the number of items. However, the DINA model generally requires more items so that all 2^M states can be well differentiated (usually 2^M is larger than the number of cognitive states L in the state mastery model). This in turn implies that the DINA model requires a larger sample for item parameter estimation.

Regarding the models for multiple choice responses, the number of free item parameters in LCM-DI is $\sum_{j=1}^{J} L_j$, which is affected by the number of cognitive states and the number of items. The number of free item parameters in the constrained MC-DINA model counterpart (i.e., equal probabilities of non-expected responses and random responses for the group for which no particular response is expected) is $\sum_{j=1}^{J} G_j$, which is a function of the number of "distinctive" response options in each item and the number of items. Thus, the MC-DINA model appears to be relatively insensitive to the number of latent states in terms of the number of item parameters. As is the case with dichotomous models, however, the MC-DINA model would require more items for a larger number of states to be well differentiated.

With the same set of items, models for multiple choice responses in general require a larger sample size than their dichotomous counterparts because they have more item parameters. This is a downside of multiple choice models, but the number of items can be reduced with well designed diagnostic items and this brings some parsimony.

2.3.4 Estimation and Model Checking

Estimation of model parameters and model selection are part of the observation component in the course of developing a CDA. Maximum likelihood estimates of parameters can be obtained by the EM algorithm (Dempster, Laird, & Rubin, 1977) in the above LCM-based CDPMs. For more complex models such as RUM, the Markov chain Monte Carlo (MCMC; Gelman & Rubin, 1992) method is available. MCMC repeatedly draws samples from the posterior distribution of parameters, from which one can obtain Bayes estimates. Both approaches are so general and established to be directly applied to most estimation problems in CDPMs. In MCMC, however, a prior distribution must be specified for each parameter to be estimated, and different priors can lead to very different estimates. Sensitivity analysis should be conducted in order to examine the influence of priors.

Assessment of model fit, in contrast, has much room for further investigation. It is a necessary step to ensure a CDPM to be a valid representation of a cognitive model and a valid tool for making accurate and reliable cognitive diagnosis. While a variety of existing statistics for different types of model fit (e.g., absolute and relative global fit, item fit, and person fit) are also applicable to CDPMs (Rupp, 2007), the type of model fit specific to CDPMs is the sensitivity (or robustness) to misspecification of cognitive states or attributes. There are ongoing studies to examine how reliably item response probabilities are estimated or how accuracy of resulting cognitive diagnosis is affected when, for example, one or more important states or attributes are missing in the model (Rupp, 2007).

If model parameters are estimated by MCMC, it is possible to conduct the posterior predictive model checks (PPMCs; Meng, 1994; Gelfand, Meng, & Stern, 1996). In PPMC, response data are repeatedly generated under each of the different sets of parameter values obtained from an MCMC run. This generates a predictive distribution of data, from which one can construct a reference distribution of any statistic used to assess model fit, and a sample value of the same statistic is compared to the reference distribution to produce the

posterior predictive p-value. Strengths of PPMC include that (a) it is a general method that can be applied to almost any model as long as the posterior distribution of model parameters are available (which is almost always the case because MCMC is applicable to a wide variety of models); (b) it takes into account uncertainty about parameter values; and (c) it works with any statistic that can be calculated from a sample, so it can be applied to different types of model fit with appropriate statistics (Rupp, 2007).

2.4 Summary and Research Questions

The construct-centered approach to CDA entails that a cognitive model guides the design of assessment tasks and interpretation of observations on those tasks. Analysis of cognitive research revealed that learning evokes qualitative changes to knowledge structure and cognitive processes, and thus CDA should be able to capture these changes as different cognitive states in order to provide instructionally-relevant information. For this purpose, it is crucial to establish an interpretive framework that explicates how different cognitive states manifest themselves as different observable responses on assessment tasks. Models on cognitive development, facets and facet clusters, construct maps, and cognitive attribute modeling can serve as such a framework and also provide a basis for developing items. One of the central issues in this vein is that errors that student make have direct implications on students' use of defective cognitive rules, and this point should be taken into account in the design of CDA. In this regard, use of diagnostic items would provide a way to address this issue.

One of the implications of using diagnostic items in CDA is that, given that most items are scored dichotomously in most of the current practices of CDA, diagnostic items are expected to allow more efficient cognitive diagnosis by increasing the interpretability of incorrect responses in individual items. Although there are several ways to define efficiency, one such indicator is the number of items required to reach a cognitive diagnosis; a more

efficient CDA can reach a diagnosis with fewer items administered while maintaining the accuracy of the diagnosis. Efficiency in CDA as defined in this manner is of practical concern. As mentioned above, shortened test length leads to shorter testing time, which in turn implies that teachers can use more time for instruction and students have more time to do their own work. Also, more efficient CDA reduces the administration cost and minimizes unnecessary exposure of test items.

Although intuitively it seems no doubt that diagnostic items make CDA more efficient and there is some supporting evidence (e.g., Kato, 2008a, 2008b), the extent to which efficiency is improved (i.e., efficiency improvement) is not fully known. Thus, it is worth considering whether efficiency improvement brought by diagnostic items is substantial; if not, we do not need to spend time and cost to construct interpretable response options.

This concern can be further elaborated by considering the fact that capability of individual diagnostic items likely varies depending on how responses are associated with cognitive rules in each item. Semi-density, which means that each response corresponds to one and only one cognitive rule in an item, is the ideal property for diagnostic items, but it is hard to achieve in practice. Response interpretability and response discrimination are among the preconditions for semi-density, and increasing these properties would lead to higher efficiency. Also, given that semi-density is rarely achievable, Wylie and Wiliam (2007) recommended designing items so that "in no case do incorrect and correct cognitive rules map on to the same response" (p. 12) as a more realistic goal. However, no research has been conducted to show to what extent considerations of these factors actually improve the performance of cognitive diagnosis.

An appropriate CDPM must be used with diagnostic items in order to take advantage of diagnostic information from incorrect responses. The state mastery model and LCM-DI assume a single latent categorical variable whose values represent different cognitive states, while binary skills models such as DINA and MC-DINA assume a set of latent indicator

variables that represent mastery patterns of basic cognitive attributes. Choice between the two types of models largely depends on the underlying cognitive model as well as the purpose of assessment and reporting requirements. However, the state mastery model and LCM-DI are more compatible with diagnostic items than binary skills models. They can easily accommodate procedural cognitive states that explicitly define *how* cognitive states generate specific responses; whereas cognitive attribute modeling, on which binary skills models build, has weaker response specificity. Although the state mastery model and LCM-DI have their own limitations, currently they seem to be the most appropriate models that can be used for diagnostic items.

Given these considerations, this thesis will address the following research questions by using LCM-DI: (a) to what extent does the use of diagnostic items improve the efficiency of cognitive diagnosis? and more specifically (b) what aspects of diagnostic items, such as response interpretability and response discrimination, are more responsible for the efficiency of cognitive diagnosis than others and how they affect the efficiency? Answering the first question provides empirical evidence that diagnostic items are useful for more efficient CDA. Answering the second question would make a good guideline for what aspects should have priority to develop better, more efficient diagnostic items.

Chapter 3

Method

Two studies were conducted to address the research questions stated in section 2.4. In the first study (Study 1; chapter 4), extensive simulations were conducted to address both the first and second research questions. Hypothetical diagnostic items were generated by systematically changing several characteristics, and their performance was compared with respect to efficiency improvement between when the full information from diagnostic items was used and when they were scored dichotomously. Evidence for efficiency improvement, if found, would be further supported if diagnostic items are proved to work in more practical settings as well. Accordingly, the second study (Study 2; chapter 5) addressed the first research question using real data from Siegler's Balance Scale tasks.

Studies 1 and 2 had three common features. First, both studies used LCM-DI (Kato, 2008b; also refer to section 2.3.2) as a base model for item responses. Like other LCMs, LCM-DI builds on several restrictive assumptions that may limit its practical use for cognitive diagnosis. Nonetheless, the proposed studies used it because of its clarity and simplicity; LCM-DI is compatible with the response-by-rule matrix, which represents the structure of a diagnostic item, that is, how responses relate to cognitive rules in each item.

Second, efficiency of cognitive diagnosis was measured by the number of items required

to reach a diagnosis with a certain level of accuracy. Also, efficiency improvement brought by diagnostic items was measured by the ratio of the numbers of administered items from diagnostic items and their dichotomized counterparts. Since the number of items required depended on a particular set of items used (e.g., a chosen set of items might be more capable of diagnosing a particular cognitive rule), it was necessary to ensure that items should be chosen in a "fair" manner for each examinee.

This consideration lead to the third feature: Items were administered adaptively in the current studies. Items were sequentially administered for each hypothetical or real examinee so that at each time an item which was optimal in terms of some criterion was selected. This process was repeated until the final diagnosis was made, and the number of items administered by that time was recorded and used as a measure of diagnostic efficiency. LCM-DI works with the existing adaptive testing procedures developed for CDPMs (Xu, Chang, & Douglas, 2003).

In the following sections, methods common to the two studies are presented. The first section describes LCM-DI once again, and the second section describes the item selection procedures for adaptive testing, followed by the last section which briefly describes computer programs used in the studies. Detailed methodology specific to each study is described in their respective chapters.

3.1 A Latent Class Model for Diagnostic Items

LCM-DI was used as a base model for generation of hypothetical diagnostic items in chapter 4 and analysis of Siegler's Balance Scale items in chapter 5. The notations for the model components are almost unchanged from those used in section 2.3.2. Suppose that there are L cognitive rules of interest and J multiple choice items. Cognitive rules are denoted by $\phi \in \{1, \ldots, L\}$. The L cognitive rules are assumed to be mutually exclusive and exhaustive, and correspondingly ϕ constitutes a latent multinomial variable in the model. Items are

indexed by $j = 1, \ldots, J$, and each item has K_j response options. Let $X_j \in \{1, \ldots, K_j\}$ be a response to item j. Item responses are assumed to be locally independent given rule ϕ.

Define the probability that an examinee who uses cognitive rule $\phi = l$ gives response k to item j as $\pi_{jkl} = P(X_j = k|\phi = l)$, where $\sum_{k=1}^{K_j} \pi_{jkl} = 1$ for all j and l. Also define the probability that an examinee uses rule $\phi = l$ as $\omega_l = P(\phi = l)$, $\sum_{l=1}^{L} \omega_l = 1$. The assumptions of mutually exclusive and exhaustive cognitive rules and local independence lead to the probability of observing a response vector $\mathbf{x}$ to be Equation 2.10. Taking the product of Equation 2.10 for all observed response patterns constitutes the likelihood function for estimating the model parameters $(\boldsymbol{\pi}, \boldsymbol{\omega})$.

In LCM-DI, the set of response probabilities $\boldsymbol{\pi}_j$ for each item is represented by the corresponding set of logit parameters $\boldsymbol{\eta}_j$ (Equations 2.11). Constraints are imposed on the logits based on the corresponding response-by-rule matrix (Equation 2.12), resulting in the set of constrained model parameters $\boldsymbol{\alpha}_j$. The constraint in Equation 2.12 is equivalent to assigning a free parameter α_{jl} to the logit of response k for which $a_{kl}^{(j)} = 1$ and 0 to the logit of response k' for which $a_{k'l}^{(j)} = 0$. In sum, LCM-DI assumes that the response k which is expected by the given rule l has a probability defined by the logit η_{jkl}, while the responses $k' \neq k$ which are not expected by rule l are chosen with equal probability. LCM-DI is directly applicable to dichotomous items, in which case $K_j = 2$ for all items, $X_j = 1$ denotes a correct response, and $X_j = 2$ denotes an incorrect response.

Estimating item response probabilities $\boldsymbol{\pi}$ is equivalent to estimating logits $\boldsymbol{\alpha}$. In either case, the item response parameters $\boldsymbol{\alpha}$ or $\boldsymbol{\pi}$ along with the marginal rule probability $\boldsymbol{\omega}$ are estimated by the maximum likelihood method through the EM algorithm (e.g., Bartholomew & Knott, 1999; Formann, 1992). The probability parameterization $(\boldsymbol{\pi})$ was used to generate hypothetical diagnostic items in chapter 4 and for adaptive testing simulations in both chapters 4 and 5, while the logit parameterization $(\boldsymbol{\alpha})$ was used in the parameter estimation in chapter 5.

Diagnosis of cognitive rule usage was made through the posterior rule distribution given a multiple choice response vector $\mathbf{x}$. The posterior probability of rule $\phi = l$ given a response pattern $\mathbf{x}$ is

$$\omega'_l(\mathbf{x}) = P(\phi = \phi_l|\mathbf{x}) \tag{3.1}$$

$$= \frac{\omega_l^{(0)} \prod_{j=1}^{J} \prod_{k=1}^{K_j} \pi_{jkl}^{I(x_{ij}=k)}}{\sum_{l'=1}^{L} \omega_{l'}^{(0)} \prod_{j=1}^{J} \prod_{k=1}^{K_j} \pi_{jkl'}^{I(x_{ij}=k)}} \tag{3.2}$$

for $l = 1, \ldots, L$, where $\omega_l^{(0)}$ is a prior probability for rule $\phi = l$. Cognitive diagnosis was made by selecting a cognitive rule with the largest posterior probability: $\hat{\phi} = \arg\max_l \omega'_l(\mathbf{x})$. Throughout the two studies, the uniform prior distribution (i.e., $\omega_l^{(0)} = 1/L$, $l = 1, \ldots, L$) was used.

3.2 Adaptive Testing with LCM-DI

In general, adaptive testing works on one or more item pools with known item parameters and a set of rules that specifies how to start testing, how to select items, and when to stop testing. This section describes item selection criteria that were used in the current studies. They were different from those used in IRT-based adaptive testing. Other parts of adaptive testing differed by study and are described in their respective chapters.

There are several different criteria for item selection which work with LCMs, but this study employed two different information criteria (the global discrimination index and the pre-posterior Shannon entropy) and random selection. Different item selection criteria were considered because they directly affected diagnostic efficiency as measured by the number of items administered.

Adaptive testing requires updating the current estimate of each examinee's rule usage $\hat{\phi}$ at each time when an item is administered and a response is observed. For this purpose,

the notation in Equation 3.2 is modified in the following manner. Let B denote the set of all items in an item pool, and $B^{(s)} = \{j^{(1)}, \ldots, j^{(s)}\}$ be a set of s items that have already been administered by the sth step. After observing responses to the set of s items in $B^{(s)}$ for examinee i, the posterior probability of rule l given response pattern $\mathbf{x}_i^{(s)}$ is given by

$$\omega_{il}^{(s)} = P(\phi_i = l|\mathbf{X} = \mathbf{x}_i^{(s)}) \tag{3.3}$$

$$= \frac{\omega_l^{(0)} \prod_{j \in B^{(s)}} \prod_{k=1}^{K_j} \pi_{jkl}^{I(x_{ij}=k)}}{\sum_{l'=1}^{L} \omega_{l'}^{(0)} \prod_{j \in B^{(s)}} \prod_{k=1}^{K_j} \pi_{jkl'}^{I(x_{ij}=k)}} \tag{3.4}$$

for $l = 1, \ldots, L$, where $\omega_l^{(0)}$, $l = 1, \ldots, L$ are prior probabilities, which are assumed to be uniform as stated in the previous section. The current rule estimate for examinee i is the rule with the largest posterior probability: $\hat{\phi}_i^{(s)} = \arg\max_l \omega_{il}^{(s)}$.

The first item selection criterion was the *global discrimination index* (GDI; Xu et al., 2003), which is based on the Kullback-Leibler (KL) information. Henson and Douglas (2005) proposed the use of the KL information for the evaluation of diagnostic capability of items. The KL information of item j from rule l to another rule l' is defined by

$$\mathrm{KL}_j(l||l') = \sum_{k=1}^{K_j} \pi_{jkl} \log \frac{\pi_{jkl}}{\pi_{jkl'}}. \tag{3.5}$$

The KL information is a measure of discrepancy between two probability distributions, which in this case are $P(X_j = k|\phi = l)$ and $P(X_j = k|\phi = l')$ (i.e., distributions of item responses given states l and l', respectively). KL becomes large if the two item response probability distributions diverge, while it is 0 when they are identical. When there are L rules, $\mathrm{KL}_j(l||l')$ constitutes a $L \times L$ asymmetric matrix with 0s on its diagonal. The GDI

for rule l on item j is then defined as the sum of KLs in the lth row of the matrix:

$$\mathrm{GDI}_j(l) = \sum_{l=1}^{L} \mathrm{KL}_j(l||l'). \tag{3.6}$$

$\mathrm{GDI}_j(l)$ represents the degree to which rule l is distinguished from the other rules on average when a response to item j is observed. At the sth step, the GDI item selection proceeds by selecting an item $j^{(s+1)}$ with the largest GDI for the current diagnosis $\hat{\phi}_i^{(s)}$:

$$j^{(s+1)} = \underset{j \in B \setminus B^{(s)}}{\arg\max}\, \mathrm{GDI}_j(\hat{\phi}_i^{(s)}) \tag{3.7}$$

$$= \underset{j \in B \setminus B^{(s)}}{\arg\max} \sum_{l=1}^{L} \mathrm{KL}_j(\hat{\phi}_i^{(s)}||l). \tag{3.8}$$

The second item selection criterion was based on the Shannon entropy applied to the posterior rule distributions. In general, the Shannon entropy for a probability distribution with L discrete states $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_L)$ is defined by

$$S(\boldsymbol{\omega}) = -\sum_{l=1}^{L} \omega_l \log \omega_l. \tag{3.9}$$

The Shannon entropy is a measure of uncertainty in a probability distribution; it reaches its maximum when all values are equally probable (i.e., the maximum uncertainty), and it decreases as the distribution tends to concentrate on a particular value (i.e., less uncertainty).

At the sth step, the Shannon entropy item selection proceeds by selecting an item $j^{(s+1)}$

that minimizes the pre-posterior Shannon entropy (ShE; Xu et al., 2003):

$$j^{(s+1)} = \underset{j \in B \setminus B^{(s)}}{\arg\min}\, S_j(\boldsymbol{\omega}_i^{(s+1)}), \tag{3.10}$$

$$= \underset{j \in B \setminus B^{(s)}}{\arg\min} \sum_{k=1}^{K_j} S(\boldsymbol{\omega}_i^{(s+1)} | X_{ij} = k) P(X_{ij} = k | \mathbf{x}_i^{(s)}), \tag{3.11}$$

where $\boldsymbol{\omega}_i^{(s+1)} | X_{ij} = k$ is the posterior rule distribution given responses $\mathbf{x}_i^{(s)}$ and $X_{ij} = k$, $j \notin B^{(s)}$, and $P(X_{ij} = k | \mathbf{x}_i^{(s)}) = \sum_{l=1}^{L} \pi_{jkl}\, \omega_{il}^{(s)}$ is the predictive distribution of response X_{ij}, $j \notin B^{(s)}$, given the current observed response pattern $\mathbf{x}_i^{(s)}$. The posterior rule distributions were calculated by Equation 3.2. The ShE item selection seeks for an item that is *expected* to minimize uncertainty in the posterior rule distribution.

The GDI item selection is a "static" procedure in the sense that each item has a fixed GDI value for each rule, which is constant for all examinees and throughout adaptive testing iterations. In contrast, the ShE for each candidate item depends on the current posterior probabilities, which differ by examinee, and thus it is more "dynamic" in nature. Although the latter requires more computation, Xu et al. (2003) reported that for items constructed under a cognitive attribute model, the ShE procedure outperformed GDI with respect to the correct classification rate for fixed length tests.

Finally, the random item selection, by which an item was randomly selected irrespective of the current rule estimate, was also considered. This provided a reference performance to which adaptive testing by GDI or ShE was compared, while it still ensured impartial selection of items over all examinees.

3.3 Computer Programs

The statistical package R (R Development Core Team, 2009a) was used for parameter estimation in LCM-DI, adaptive testing simulations, and all other relevant analysis. A

computational subroutine that implemented the EM algorithm was written in C and called from R. These programs were tested for their validity, and details of the testing are described in Appendix A. Actual programs used are attached in Appendix B.

Chapter 4

Study 1: Adaptive Testing Simulation for Hypothetical Diagnostic Items

4.1 Introduction

The purpose of Study 1 was to examine the effect of several aspects that characterize diagnostic items on efficiency of cognitive diagnosis and its improvement. More specifically, this study considered the following characteristics: (a) the number of cognitive rules (L), (b) response interpretability (RI), (c) the number of cognitive rules that generate a particular response (NRP), and (d) item discrimination (ID).

The number of cognitive rules, L, is usually determined by the specification of assessment, but also can be changed in the process of refining cognitive and psychometrics models. More items will be required to distinguish among a larger number of cognitive rules. If efficiency improvement is more apparent with a larger number of cognitive rules, however, it will be supporting evidence for using diagnostic items and thus this factor was

included in the current study.

RI and NRP are more controllable features of diagnostic items when designing them and thus of more practical concern than the number of cognitive rules. These features roughly correspond to response interpretability and response discrimination as Bart and Williams-Morris (1990) proposed (see chapter 2), and they are made explicit through response-by-rule matrices. RI in this study was defined as the *number* of response options in an item that are associated with at least one cognitive rule, although in its original definition in Bart and Williams-Morris (1990) it was defined as the *proportion* of such response options. Since increasing RI means that more responses are interpreted by cognitive rules, it was expected to have a positive effect on efficiency.

Response discrimination is another fundamental feature of diagnostic items, but it was not easy to manipulate for the purpose of the current study. Instead, this study focused on NRP as a similar but more manageable feature. For a given value of RI, increasing NRP implies that more cognitive rules are associated with each response on average. This roughly corresponds to decreasing response discrimination, but its effect can be positive or negative. From the literature (e.g., Bart & Williams-Morris, 1990; Wylie & Wiliam, 2007), lower NPR (or higher response discrimination) is desirable in a single item, because an observed response is related to a cognitive rule less ambiguously. When multiple items are present and the number of cognitive rules is larger than the number of response options, however, it is not immediately clear which is better (a) to have a set of items in which each item targets specific rules with low NRP and items as a whole cover all cognitive rules, or (b) to have a set of items each of which covers as many cognitive rules as possible (e.g., two items with high NRP, when appropriately combined, may provide an efficient diagnosis for a particular cognitive rule).

Item discrimination, ID, was defined as the average item response probability for rule-predicted responses in the LCM-DI, and this relates to psychometric quality of diagnostic

items. More specifically, ID in this study referred to the magnitude of item response probabilities for rule-predicted responses in the LCM-DI, that is, π_{jkl} for response k which is generated by rule l. If ID is higher (and each rule predicts a different response), then it is more likely that each cognitive rule manifests itself as the corresponding expected response, leading to better efficiency. At the operational level, ID reflects the clarity and appropriateness of the task definition (i.e., whether the item appropriately elicits student responses that are relevant to the target cognitive rules) and the extent to which cognitive rules and the corresponding responses are clearly defined. It also reflects difficulty of applying a rule. For example, ID can be low for rules that require relatively complex cognitive operations; more complex rules likely increase the possibility of unintended errors in the course of applying them.

In order to examine how these item characteristics affect efficiency, this study employed Monte Carlo adaptive testing simulations (a general procedure of Monte Carlo adaptive testing simulations was described by Weiss, n.d., at the CAT Central web site). Hypothetical item pools were created by systematically manipulating the above features. Adaptive testing simulations were conducted using subsets of items in those item pools, and efficiency was compared across simulations.

4.2 Description of Procedure

4.2.1 Item Pools

Three different item pools were constructed corresponding to the number of cognitive rules $L = 5$, 7, and 9. These item pools are denoted by IP5, IP7, and IP9, respectively. In each case, Rule 1 always represented the "random guessing" rule, with which examinees randomly choose responses in all items. Also, Rule 2 always represented the "correct" rule, which generates a correct response for all items. For simplicity, other rules (i.e.,

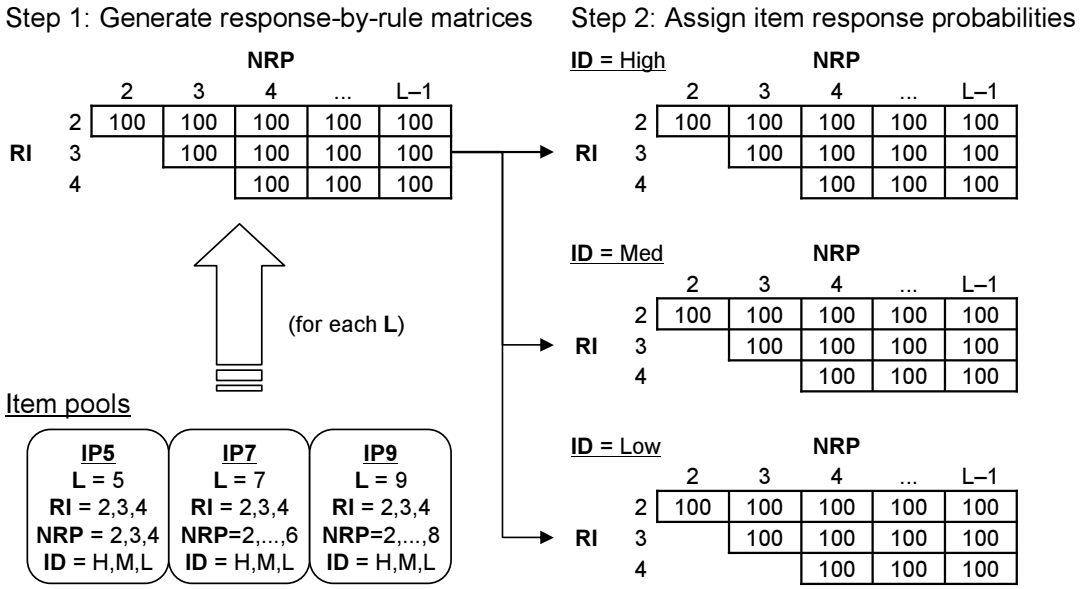

Figure 4.1: Flow of the item pool generation procedure. The number in each cell is the number of items generated for that particular combination of diagnostic item characteristics.

$\phi = 3, \ldots, L$) were assumed to either (a) generate one particular response or (b) relate to no particular response (implying random guessing) for each item. In all three item pools, all items had four response options, that is, $K = K_j = 4$ and $X_j \in \{1, 2, 3, 4\}$ for all j. Let response $X_j = 1$ always be the "correct" response for each item without loss of generality.

Within each item pool, RI, NRP, and ID were manipulated systematically to generate items. Since RI and NRP are characteristics related to a response-by-rule matrix and ID is a characteristic which belongs to item response probabilities in LCM-DI, item generation required two steps. In the first step, a set of response-by-rule matrices were generated by systematically changing RI and NRP for each L. This was followed by the second step in which item response probabilities (π_{jkl} or equivalently α_{jkl} under the LCM-DI) were generated by different specifications of ID and attached to the response-by-rule matrices.

Figure 4.1 depicts the overall flow of the above item generation steps. In the first step,

it was necessary to specify components of a response-by-rule matrix ($a_{kl}^{(j)}$) for each item. From the general item and cognitive rule specifications above, all response-by-rule matrices had the following form in common:

$$\mathbf{A}_j = \begin{array}{c} \\ \text{Response 1} \\ \text{Response 2} \\ \text{Response 3} \\ \text{Response 4} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{Rule 1} & \text{Rule 2} & \text{Rule 3} & \cdots & \text{Rule } L \end{array} \\ \begin{bmatrix} 0 & 1 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \end{bmatrix} \end{array}, \tag{4.1}$$

where the "$*$" entries varied item by item. Each column had either only one "1" (the rule generates one particular response) or no "1" (random guessing or no response predicted; columns for rules that represented random guessing could be either all ones or all zeros, but the latter was adopted for convenience in this study).

RI, which is the number of response options that are associated with at least one cognitive rule, was set to 2, 3, or 4. RI corresponds to the number of rows in a response-by-rule matrix where there is at least one "1." Given L and RI, NRP, the number of cognitive rules that generate a particular response, ranged from RI to $L - 1$. The minimum value was RI, because RI responses had to be associated with all different rules by definition. The maximum value was $L - 1$, because Rule 1 was always the random guessing rule.

For each given combination of L, RI, and NRP, 100 response-by-rule matrices were generated by

1. randomly selecting RI $-$ 1 rules from Rules 3 through L and assigning them to responses 2, 3, or 4 so that they predicted all different responses; and

2. randomly selecting NRP $-$ RI rules from the rules which were *not* selected in step 1 and randomly assigning them to responses that were already accounted for in step 1.

In the second step, item response probabilities were generated and attached to the response-by-rule matrices obtained in the first step. For a given response-by-rule matrix, ID was set to high, medium, or low. In the High ID condition, values of π_{jkl} for which $a_{kl}^{(j)} = 1$ were randomly drawn from the uniform distribution with range $(.90, 1.00)$ (mean .95). Similarly, values of π_{jkl} were randomly drawn from $U(.80, .90)$ (mean .85) and $U(.70, .80)$ (mean .75) in the Medium and Low conditions, respectively.

For each response-by-rule matrix, three items were generated each of which corresponded to one of the three ID conditions (high, medium, and low); under each condition, a probability value was assigned to each π_{jkl} for which $a_{kl}^{(j)} = 1$ and set $\pi_{jk'l} = (1 - \pi_{jkl})/3$ for $k' \neq k$ for the given rule l. The value of π_{jkl} for rules that predicted no particular response was all set to .25 to represent random guessing among four response options. These probability specifications ensured the structure of LCM-DI.

In order to describe the two-step item generation process more closely, Figure 4.2 shows an example of generating three items with $L = 5$, RI = 2, and NRP = 3. This item belongs to IP5. In Step 1, a response-by-rule matrix is generated with constraints RI = 2 and NRP = 3 (the left half of Figure 4.2). The first two columns represent the random guessing and correct rules, respectively, and these two columns are fixed in all items over the three item pools. Because RI = 2, one column (rule) is randomly selected from columns 3 to 5 and a response corresponding to that rule is also randomly chosen from $k = 2$, 3, and 4. In the current example, Rule 3 was selected and assigned to response 3. Since NRP = 3, one more column needs to be randomly selected from columns 4 and 5, to which a response is assigned from two responses that were already assigned to other rules (the possibility is response 1 or 3). In the current case, Rule 5 was selected and assigned to response 1.

Once a response-by-rule matrix had been specified, three different sets of item response probabilities were assigned according to the three ID conditions (Step 2; the right half of Figure 4.2). For each ID condition, item response probabilities were randomly generated

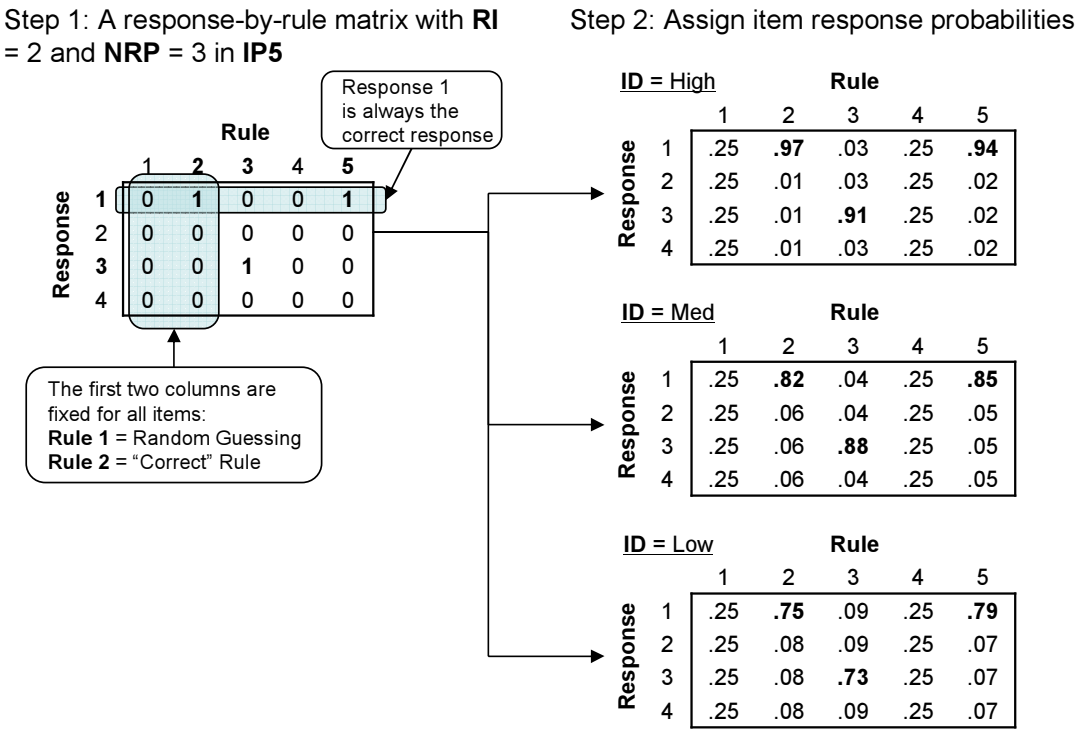

Figure 4.2: Detailed item generation procedure for items with $L = 5$, RI $= 2$, and NRP $= 3$.

from the corresponding uniform distribution for the "1" entries in the response-by-rule matrix. In the present example, three probability values need to be assigned (π_{j12}, π_{j33}, and π_{j15}) under each ID condition. Values .97, .91, and .94 were assigned to these probabilities, respectively, in the high ID condition, and within each of those columns probabilities of the other responses are set equal (.01, .03, and .02, respectively). In the columns for which no particular response is associated (columns 1 and 4), response probabilities are set to .25. Probabilities in the other ID conditions are assigned in the same manner.

In order to examine efficiency improvement between multiple choice and dichotomous cases, item response probabilities for dichotomized responses also needed to be specified. Probabilities of correct responses were simply those for response 1. Probabilities of incorrect responses were also readily available by collapsing the original item response probabilities

over responses 2, 3, and 4. This completed the creation of item pools.

4.2.2 Procedure of Adaptive Testing Simulations

Multiple sets of adaptive testing simulations were conducted with subsets of items that had a particular characteristic in common. More specifically, for each L, a subset of 100 items were selected from the corresponding item pool (IP5, IP7, or IP9) according to the item characteristics under consideration:

- Response interpretability (RI): 2, 3, or 4
- Number of cognitive rules that generate a particular response (NRP): RI to $L-1$ for a given RI
- Item discrimination (ID): High (.95), Medium (.85), or Low (.75)

All possible combinations of these three item characteristics were considered for each L. This lead to 108 different conditions, each of which involved 100 items that shared the same item characteristics.

For a given subset of items, it was ensured that expected response patterns over the selected subset of items were all different over a given set of rules. Then, multiple choice responses of $100L$ hypothetical examinees were simulated. One-hundred examinees were assigned to each of the L cognitive rules, and then their multiple choice responses were generated by using the corresponding item response probabilities. The multiple choice responses were then dichotomized to obtain the corresponding dichotomous data.

For each combination of item characteristics, adaptive testing proceeded for each examinee as follows:

Starting rule: The first item is randomly selected from the item subset (but the same first item is used for each examinee within each item characteristic combination of RI, NRP, and ID).

Item selection: GDI, ShE, or RND

Stopping rule: Testing stops if a posterior probability ($\omega_{il}^{(s)}$) for any rule exceeds .90.

There were two runs of adaptive testing for each combination of item characteristics, one for multiple choice data and the other for the corresponding dichotomized data. As a result, the total number of adaptive testing runs across all combinations of L, RI, NRP, ID, item selection methods, and response type (multiple choice and dichotomous) was 648 (there were 324 "pairs" of multiple choice/dichotomous response type comparisons).

Outcomes recorded for each examinee within each simulation run were (a) the final rule estimate, (b) the number of items administered to reach the final diagnosis, which was considered as a measure of efficiency. For each simulation "pair" (the same adaptive testing condition except for the response type), the ratio of the number of items for multiple choice administration to that for dichotomous administration was computed for each examinee. This ratio (efficiency ratio) was used as a measure of efficiency improvement from dichotomous to multiple choice responses.

These numbers and ratios were compared across item characteristics (L, RI, NRP, and ID) and item selection methods (GDI, ShE, and RND). Results from each simulation were represented by the median number of items administered (efficiency) and the median efficiency ratio (efficiency improvement). Also, correct classification rates were computed from examinees' true rules and final rule estimates within each simulation. They were used to compare the performance of multiple choice and dichotomized items and also to ensure that adaptive testing did not lower accuracy of cognitive diagnosis. Agreement between the true rules and rules estimated from all 100 items within each item subset was also computed to evaluate the diagnostic capability of those items as a test.

It should be noted that comparisons between the two response types became invalid when RI = NRP, where all "incorrect" rules that were associated with a particular response could not predict a "correct" response in each item (because all rules had to predict all

different responses when RI = NRP and the "correct" rule always predicted the correct response) and those rules could not be distinguished well from each other when responses were dichotomized. Thus, cases in which RI = NRP were not considered for *efficiency improvement* (i.e., comparison of efficiency between the response types); this resulted in 243 adaptive testing pairs for the investigation of efficiency improvement. Meanwhile, all 324 multiple choice cases were used to examine the effects of item characteristics and item selection methods on *efficiency*.

4.3 Results

4.3.1 Test Characteristics

There were 108 subsets of items, each of which corresponded to a combination of item characteristics (L, RI, NRP, and ID). Within each item subset, there were 100 items which had common item characteristics.

"Test" difficulty (i.e., the mean probability of correct responses for all items within an item subset) over all 108 subsets of items was .31 with standard deviation .05. Either L or ID hardly affected the overall test difficulty, but RI negatively correlated with it; the mean difficulty was .36, .29, and .25 for RI = 2, 3, and 4, respectively. Overall, items used in the current study were more difficult than items in typical achievement tests in actual use.

It was confirmed that in all 108 item subsets the expected response patterns were all distinctive for both multiple choice and dichotomized responses. Agreement between cognitive diagnoses based on all 100 items and the true rules within each of the item subsets (except for those with RI = NRP) was almost perfect in all (both multiple choice and dichotomous) cases; majority of them indicated perfect agreements and the overall mean proportion of agreement was .997 with standard deviation .009. Thus, each item subset, as a whole, had enough power to distinguish a given set of cognitive rules.

Correct classification rates over all 648 adaptive testing simulations had mean .94 with standard deviation .01. Although dichotomous cases yielded slightly smaller correct classification rates than multiple choice cases, the differences were very small; Thus, loss of diagnostic accuracy due to adaptive testing was considered minimum.

There were "examinees" who did not meet the stopping criterion with all 100 items in some cases (termed Not Reached; NR). The proportion of NR was 0 in almost all multiple choice cases with maximum .004. It was 0 or very close to 0 in most of the dichotomous cases as well (RI = NRP cases were excluded from this calculation). Among $L = 9$ and Low ID conditions, however, NR exceeded .10 when RI = 2 and NRP = 3 (the mean was .14 over three item selection methods) and when RI = 3 and NRP = 4 (mean .11). This also indicated efficiency improvement due to the use of multiple choice responses.

4.3.2 Effects of Response Type and Item Characteristics on Efficiency Improvement

Efficiency improvement from dichotomous to multiple choice responses was one of the primary concerns of this study, and the efficiency ratios were analyzed first. Cases in which RI = NRP were excluded from this analysis, because the dichotomous results were considered invalid for the reason mentioned above. Thus, 243 efficiency improvement ratios were compared across item characteristics and item selection methods.

Plot (a) in Figure 4.3 shows the distribution of median efficiency ratios for all 243 cases. The mean of these ratios was 0.58. Thus, the number of items administered is reduced on average by 42% by using multiple choice responses. The standard deviation was 0.15, and the entire distribution ranged from 0.22 to 0.86. Although there is large variability, the use of multiple choice responses in general lead to substantial improvement in efficiency within the variation of item characteristics and item selection methods considered in the current study.

Efficiency improvement was expected to vary depending on item characteristics and item selection methods. Plots (b) through (f) in Figure 4.3 show the distributions of efficiency ratios by item characteristics and item selection methods. These plots indicate that efficiency improvement more or less depended on the item characteristics and item selection methods, while there was still large variability within each fixed value of these variables. Among these, RI showed a positive effect on efficiency improvement; more improvement is expected as more responses are explained by rules. Item selection methods had the opposite effect; efficiency improvement decreases as supposedly more effective item selection methods are used (ShE was considered more effective than GDI in terms of reducing the number of items administered, while both of these methods were better than RND in the same regard). Effects of the other variables were less clear than RI and item selection methods; L and NRP might increase efficiency improvement slightly, while ID seemed to work in the opposite direction.

In order to examine the effects of item characteristics and item selection methods on efficiency improvement more closely and systematically, the efficiency ratios were subjected to multiple regression analysis. Initially, L, RI, NRP, ID, item selection methods, and all of their two-way interactions were entered in the regression equation. Item selection methods were entered as a factor with RND being the base category, and the other variables were entered as numeric linear predictors. Then, all interactions that were not significant with the ordinary ANOVA F-tests were removed from the equation. This left out $L \times$ RI, $L \times$ item selection methods, RI $\times$ NRP, and RI $\times$ ID interactions.

Table 4.1 shows the ANOVA table from the final model. The adjusted R^2 was .89, so item characteristics and item selection methods accounted for the efficiency ratio very well. The sum of squares (SS) for RI was larger than for the other main and interaction effects, which agreed with the observation of plots in Figure 4.3.

Table 4.2 shows the estimated regression coefficients in the final model equation. Be-

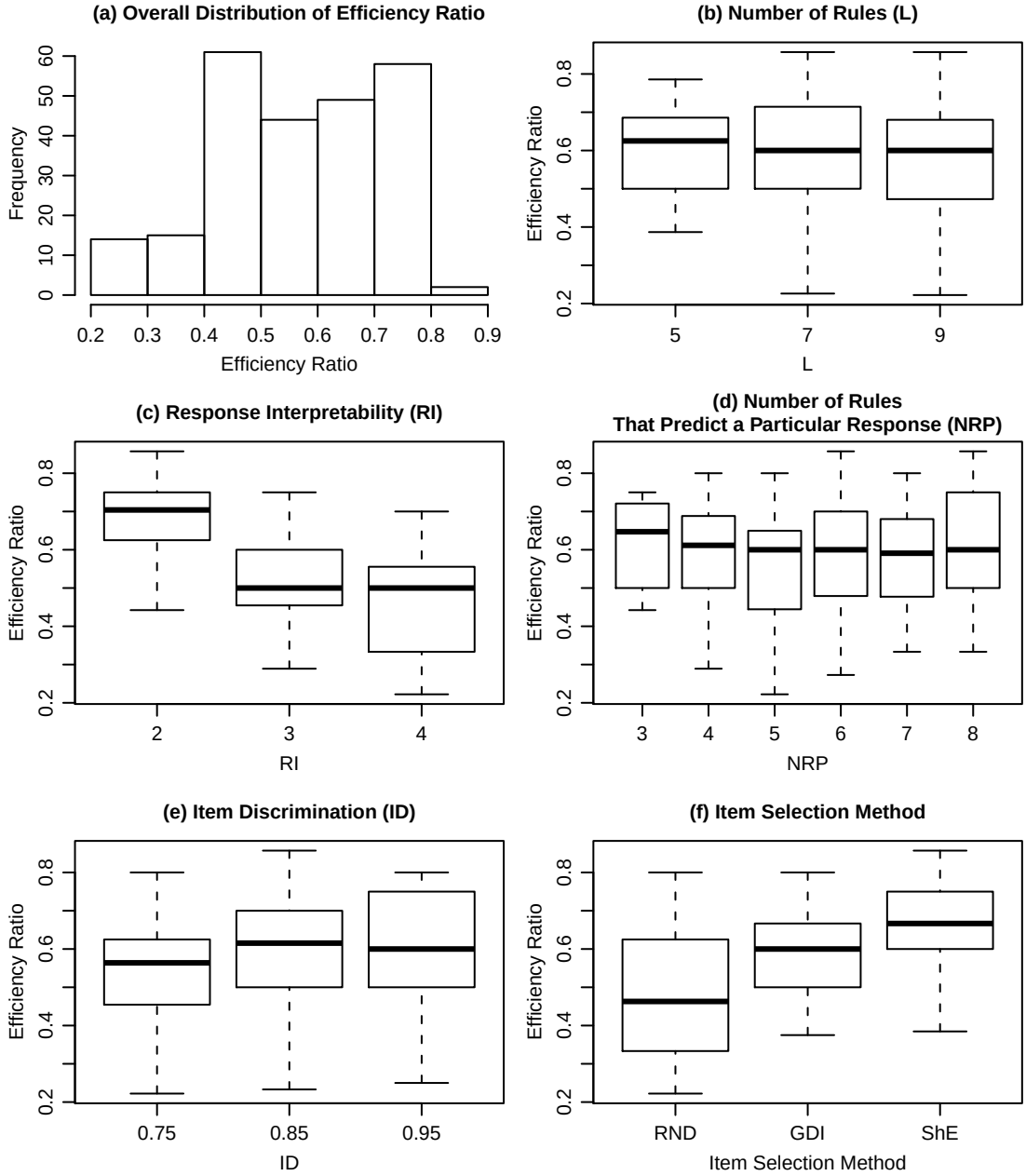

Figure 4.3: Overall distribution of efficiency ratios and decomposition by item characteristics and item selection methods. In each boxplot, the horizontal bar in the middle of a box indicates the median, a box indicates the interquartile range, and whiskers cover the central 98% range.

Table 4.1: ANOVA Table for Efficiency Ratio

Source[a]	Df	SS	F	p-Value
Intercept	1	0.05	20.94	0.0000
L	1	0.02	9.74	0.0020
RI	1	1.78	735.98	0.0000
NRP	1	0.07	28.16	0.0000
ID	1	0.01	2.61	0.1076
SEL	2	0.09	18.27	0.0000
$L \times$ NRP	1	0.01	4.53	0.0345
$L \times$ ID	1	0.03	13.54	0.0003
RI $\times$ SEL	2	0.30	61.47	0.0000
NRP $\times$ ID	1	0.02	8.65	0.0036
NRP $\times$ SEL	2	0.21	44.12	0.0000
ID $\times$ SEL	2	0.10	20.05	0.0000
Error	227	0.55		

[a] L = number of rules; RI = response interpretability; NRP = number of rules that predict a particular response; ID = item discrimination; GDI = global discrimination index item selection; ShE = Shannon entropy item selection; SEL = item selection methods (GDI and ShE).

cause the efficiency ratio decreases if the use of multiple choice responses improves efficiency, negative coefficients indicate positive effects on efficiency improvement.

The number of cognitive rules (L) by itself had a positive effect on efficiency improvement; the main effect coefficient was -0.0916. However, L interacted with NRP (-0.0044) and ID (0.1205); NRP strengthened the effect of L while ID weakened it. As a result, the coefficient for L varied as a function of NRP and ID:

$$\text{Coef}(L) = -0.0916 - 0.0044(\text{NRP}) + 0.1205(\text{ID}). \tag{4.2}$$

The value of the above equation was computed for all combinations of NRP and ID in the current data to examine the behavior of the coefficient. The resulting distribution is shown in Table 4.3. It indicates that more than 75% of them were negative. Thus, L increased efficiency improvement in most cases (i.e., use of multiple choice responses lead to more

Table 4.2: Regression Coefficients for Efficiency Ratio

Variable[a]	Estimate	Std. Error	t	p-Value
Intercept	0.9810	0.2144	4.58	0.0000
L	−0.0916	0.0294	−3.12	0.0020
RI	−0.2022	0.0075	−27.13	0.0000
NRP	0.1668	0.0314	5.31	0.0000
ID	−0.3784	0.2343	−1.62	0.1076
GDI	−0.3715	0.0875	−4.25	0.0000
ShE	0.1403	0.0875	1.60	0.1102
$L \times$ NRP	−0.0044	0.0021	−2.13	0.0345
$L \times$ ID	0.1205	0.0328	3.68	0.0003
RI × GDI	0.1127	0.0105	10.74	0.0000
RI × ShE	0.0814	0.0105	7.76	0.0000
NRP × ID	−0.0885	0.0301	−2.94	0.0036
NRP × GDI	−0.0509	0.0055	−9.20	0.0000
NRP × ShE	−0.0345	0.0055	−6.24	0.0000
ID × GDI	0.5224	0.0947	5.52	0.0000
ID × ShE	0.0064	0.0947	0.07	0.9465

[a] L = number of rules; RI = response interpretability; NRP = number of rules that predict a particular response; ID = item discrimination; GDI = global discrimination index item selection; ShE = Shannon entropy item selection.

efficiency improvement as L became large), although the amount of improvement was only about 1% on average.

Response interpretability (RI) had significant interactions with item selection methods. The main effect coefficient of −0.2022 represents a positive effect on efficiency improvement when items were selected randomly (RND). The coefficient increased to −0.0895 (= −0.2022 + 0.1127) and −0.1208 (= −0.2022 + 0.0814) when items were selected adaptively with GDI and ShE, respectively (note that GDI and ShE in Table 4.2 were treated as dummy variables; GDI = 1 when the item selection method was GDI, ShE = 1 when the item selection method was ShE, and both are 0 for RND). Thus, adaptive item selection worked in the direction of decreasing the positive effect of RI on efficiency improvement. Still, RI had a positive impact on efficiency improvement in spite of these adverse inter-

Table 4.3: Distributions of Regression Coefficients for Efficiency Ratio

Variable[a]	Interacts With[a]	Min.	25%	Median	Mean	75%	Max.
L	NRP, ID	−0.04	−0.02	−0.01	−0.01	−0.00	0.01
RI[b]	SEL	−0.20	–	−0.12	−0.14	–	−0.09
NRP	L, ID, SEL	−0.01	0.01	0.03	0.03	0.05	0.08
ID[c]	L, NRP, SEL	−0.01	0.00	0.02	0.03	0.05	0.10
GDI[d]	RI, NRP, ID	−0.16	0.00	0.10	0.09	0.17	0.32
ShE[d]	RI, NRP, ID	0.03	0.11	0.18	0.17	0.23	0.30

[a] L = number of rules; RI = response interpretability; NRP = number of rules that predict a particular response; ID = item discrimination; GDI = global discrimination index item selection; ShE = Shannon entropy item selection; SEL = item selection methods (GDI and ShE).
[b] Summary for RI was only based on three different values, which are shown as Min., Median, and Max., respectively.
[c] The coefficient values are adjusted for the increment of .10.
[d] Coefficients for GDI and ShE represent their relative effects to random item selection (RND).

actions. More efficiency improvement is expected when multiple choice responses are used with items with larger RI; the amount of improvement by increasing RI by one ranges from 9% (with GDI) to 20% (with RND).

The number of rules that predict a particular response (NRP) had a negative main effect on efficiency improvement (0.1668), while this might be compensated to some extent by interactions with L, ID, and item selection methods, which were all positive toward efficiency improvement. The overall effect of NRP was computed by entering all possible combinations of NRP, L, ID, and item selection methods in the following equation:

$$\text{Coef(NRP)} = 0.1668 - 0.0044(L) - 0.0885(\text{ID}) - 0.0509I(\text{GDI}) - 0.0345I(\text{ShE}), \quad (4.3)$$

where $I(\cdot)$ is the indicator variable. The result is shown in Table 4.3. In most cases, larger NRP lead to negative efficiency improvement. The average amount of decrease was about 3%, but it could be as high as 8%. The above equation shows that higher decrease in

efficiency improvement is more likely when L and ID are relatively small and items are *not* administered adaptively.

Item discrimination (ID) had a positive effect on efficiency improvement (-0.3784) but also interacted with L, NRP, and item selection methods, which overwhelmed the positive main effect. The coefficient for ID varied as a function of these variables, and it is given by

$$\text{Coef(ID)} = (-0.3784 + 0.1205(L) - 0.0885(\text{NRP}) + 0.5224I(\text{GDI}) + 0.0064I(\text{ShE})) \times .10, \tag{4.4}$$

where .10 is multiplied to adjust the coefficient so that it represents the amount of improvement when ID is increased by .10. Table 4.3 shows that the overall effect of ID was negative to efficiency improvement, that is, less efficiency resulted as ID became higher (on average, there was 3% decrease in efficiecy improvement as ID increased by .10). This indicates that if items have high ID, dichotomous responses still work so well to limit efficiency improvement by using multiple choice responses. It should also be noted that the use of GDI item selection with high ID items largely decreases efficiency improvement by ID (interaction 0.5224). This is probably because GDI, by its definition, more directly relates to item response probabilities than ShE, which is basically computed from posterior rule probabilities. When ID is high, GDI still can pick up optimal items even if responses are dichotomized.

Item selection methods (GDI and ShE) interacted with all other variables except for L as described above. They compensated the negative effect of NRP on efficiency improvement, while they decreased the positive effects of RI and ID. Thus, they tended to absorb the effects of item characteristics whether those effects were positive or negative. Their overall

effects are given by

$$\text{Coef(GDI)} = -0.3715 + 0.1127(\text{RI}) - 0.0509(\text{NRP}) + 0.5224(\text{ID}), \tag{4.5}$$

$$\text{Coef(ShE)} = 0.1403 + 0.0814(\text{RI}) - 0.0345(\text{NRP}) + 0.0064(\text{ID}). \tag{4.6}$$

The average overall effects were both negative to efficiency improvement: 0.09 for GDI and 0.17 for ShE (see Table 4.3). This is probably because diagnosis with dichotomous responses gains more by optimized item selection than with multiple choice responses. ShE on average had larger negative effects than GDI. This indicates that efficiency improvement becomes even smaller as item selection is more optimized, on the ground that ShE provided more efficient diagnosis than GDI (Xu et al., 2003).

To summarize, efficiency of cognitive diagnosis was improved by using multiple choice responses. On average, efficiency improved by 42% within the configurations of diagnostic items considered in this study, meaning that one can reach a diagnosis with 42% fewer items. However, there was large variability in efficiency improvement due to item characteristics and item selection methods. Although these variables interacted with each other to make simple interpretations difficult, use of multiple responses likely leads to larger efficiency improvement as (a) there are more cognitive rules (larger L) and/or (b) more responses are interpreted by cognitive rules in each item (larger RI). Meanwhile, efficiency improvement tends to be limited as (a) more rules are associated with responses in each item (larger NRP), (b) probabilities of rule-predicted responses become higher (larger ID), and/or (c) item selection in adaptive testing is more optimized (GDI or ShE item selection).

4.3.3 Effect of Item Characteristics on Efficiency Improvement in Diagnostic Items

The previous section showed that the use of multiple choice responses resulted in substantial efficiency improvement. Given that result, it was worth considering what item characteristics make diagnostic items better (i.e., improved efficiency). For this purpose, efficiency (i.e., the number of items administered) was compared across item characteristics and item selection methods only among adaptive testing simulations for multiple choice responses.

The data for this analysis consisted of 324 multiple choice cases out of the total 648 adaptive testing simulation runs. They included all combinations of L, RI, NRP, ID, and item selection methods (GDI, ShE, and RND). Plot (a) in Figure 4.4 shows the distribution of the observed median number of items, as a measure of efficiency, from all 324 cases. The distribution was highly skewed to the right; the median was 5 with interquartile range $(3, 6)$ and range $(2, 36)$. Plots (b) through (f) in Figure 4.4 show decompositions by item characteristics and item selection methods. It is clear that efficiency was highly dependent on these variables. L seemed to decrease efficiency, while all the other variables increased it.

In order to examine the effects of these variables more closely and systematically, the data were analyzed by multiple regression as in the previous section. In the current analysis, however, the dependent variable was the median number of items in each case; the median was used to minimize the bias due to the NR cases and the skewed distribution of the number of items within each simulation result. The median number of items was then log-transformed (with base e), because the original median distribution was highly skewed as shown in plot (a) in Figure 4.4. Initially, main effects of all of all item characteristics and item selection methods and all of their two-way interactions were entered in the regression equation as independent variables; item selection methods were entered as a factor with RND being the base category, and the other variables were entered as numeric linear

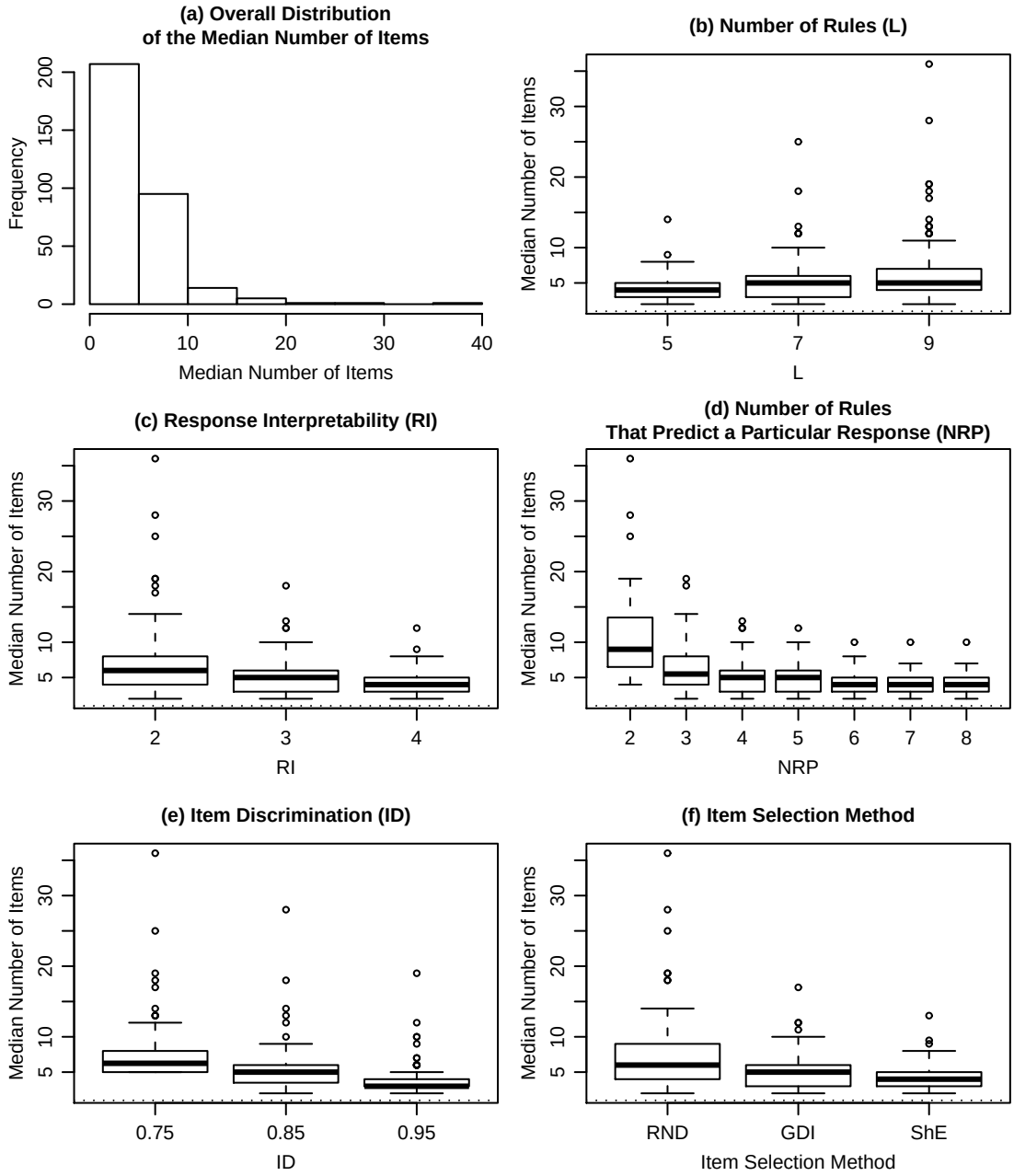

Figure 4.4: Overall distribution of the median number of items administered and decomposition by item characteristics and item selection methods. In each boxplot, the horizontal bar in the middle of a box indicates the median, a box indicates the interquartile range, and whiskers cover the central 98% range.

Table 4.4: ANOVA Table for Efficiency

Source[a]	Df	SS	F	p-Value
Intercept	1	3.72	194.09	0.0000
L	1	0.03	1.50	0.2218
RI	1	1.31	68.50	0.0000
NRP	1	1.85	96.66	0.0000
ID	1	1.79	93.12	0.0000
SEL	2	0.15	4.03	0.0187
$L \times$ NRP	1	0.10	5.42	0.0205
$L \times$ ID	1	0.12	6.35	0.0123
$L \times$ SEL	2	1.49	38.74	0.0000
RI $\times$ NRP	1	0.18	9.23	0.0026
RI $\times$ SEL	2	0.21	5.47	0.0046
NRP $\times$ SEL	2	1.57	41.04	0.0000
Error	308	5.91		

[a] L = number of rules; RI = response interpretability; NRP = number of rules that predict a particular response; ID = item discrimination; GDI = global discrimination index item selection; ShE = Shannon entropy item selection; SEL = item selection methods (GDI and ShE).

predictors. Then, all interactions that were not significant at the .05 level in the ordinary ANOVA F-tests were removed from the equation. This resulted in the final regression equation which contained all main effects and interactions except for $L \times$ RI, RI $\times$ ID, NRP $\times$ ID, and ID $\times$ item selection methods interactions. Table 4.4 shows the ANOVA table for the final model. The adjusted R^2 was .926, indicating that more than 90% of the variance of efficiency was explained by the item characteristics and item selection methods. The main effects had much higher sum of squares (SS) than the interactions, and among them ID had the largest SS, followed by NRP, SEL, and RI.

Table 4.5 shows the estimated regression coefficients in the final model equation. Negative coefficients indicate that efficiency is improved by increasing the corresponding variables (i.e., positive effects on efficiency). Because the dependent variable is on the natural logarithmic scale, the exponent of each coefficient represents the multiplicative factor to

Table 4.5: Regression Coefficients for Efficiency

Variable	Estimate	Std. Error	t	p-Value
Intercept	6.2545	0.4489	13.93	0.0000
L	0.0699	0.0571	1.22	0.2218
RI	−0.3040	0.0367	−8.28	0.0000
NRP	−0.3586	0.0365	−9.83	0.0000
ID	−4.7651	0.4938	−9.65	0.0000
GDI	−0.2843	0.1147	−2.48	0.0138
ShE	−0.2798	0.1147	−2.44	0.0153
$L \times$ NRP	0.0099	0.0042	2.33	0.0205
$L \times$ ID	0.1593	0.0632	2.52	0.0123
$L \times$ GDI	−0.0823	0.0143	−5.76	0.0000
$L \times$ ShE	−0.1234	0.0143	−8.64	0.0000
RI $\times$ NRP	0.0194	0.0064	3.04	0.0026
RI $\times$ GDI	0.0524	0.0246	2.13	0.0339
RI $\times$ ShE	0.0800	0.0246	3.26	0.0013
NRP $\times$ GDI	0.0911	0.0131	6.97	0.0000
NRP $\times$ ShE	0.1109	0.0131	8.49	0.0000

[a] L = number of rules; RI = response interpretability; NRP = number of rules that predict a particular response; ID = item discrimination; GDI = global discrimination index item selection; ShE = Shannon entropy item selection.

the number of items administered. For example, the coefficient for the main effect of L is 0.0699 and its exponent is $\exp(0.0699) = 1.0724$. This means that the number of items was increased by 7.24% if one more rule was added (ignoring the effects of the other variables). The various interactions made difficult simple interpretation of these coefficients, and thus the same approach as in the previous section was used to estimate the average behavior of the variables.

The number of rules (L) had a positive coefficient (0.0699), though it was not significant. L also interacted with NRP, ID, and item selection methods. The equation for the coefficient for L including all interactions relevant to it is given by

$$\text{Coef}(L) = 0.0699 + 0.0099(\text{NRP}) + 0.1593(\text{ID}) - 0.0823I(\text{GDI}) - 0.1234I(\text{ShE}). \quad (4.7)$$

Table 4.6: Distributions of Regression Coefficients for Efficiency

Variable[a]	Interacts With[a]	Min	Median	Mean	Max	exp(Mean)
L	NRP, ID, SEL	0.09	0.18	0.19	0.30	1.20
RI	NRP, SEL	−0.27	−0.17	−0.16	−0.07	0.85
NRP	L, RI, SEL	−0.27	−0.14	−0.16	−0.08	0.85
ID[b]	L	−0.40	−0.37	−0.37	−0.33	0.69
GDI[c]	L, RI, NRP	−0.74	−0.31	−0.33	−0.09	0.72
ShE[c]	L, RI, NRP	−1.01	−0.44	−0.47	−0.13	0.62

[a] L = number of rules; RI = response interpretability; NRP = number of rules that predict a particular response; ID = item discrimination; GDI = global discrimination index item selection; ShE = Shannon entropy item selection; SEL = item selection methods (GDI and ShE).
[b] Summary for ID was only based on three different values, which are shown as Min, Median, and Max, respectively. The coefficient values are adjusted for the increment of .10.
[c] Coefficients for GDI and ShE represent their relative effects to random item selection (RND).

Applying the above equation to all combination of NRP, ID, and item selection methods results in the distribution of the L coefficient as shown in Table 4.6. All coefficient estimates were positive, indicating that L had a negative effect on efficiency. The mean coefficient was 0.19, and this was transformed to obtain the factor of 1.20 as shown in the last column of Table 4.6. Thus, on average, adding one more rule required 20% more items. With higher NRP and ID this value was increased, while use of adaptive item selection decreased the negative effect of L.

Response interpretability (RI) had a negative main effect coefficient (−0.3040). It interacted with NRP and item selection methods, all of which slightly increased the coefficient toward zero:

$$\text{Coef(RI)} = -0.3040 + 0.0194(\text{NRP}) + 0.0524I(\text{GDI}) + 0.0800(\text{ShE}). \tag{4.8}$$

Table 4.6 shows that the resulting coefficient estimates were negative in all cases, and

thus increasing the number of rule-predicted responses was always effective to increase efficiency. The mean coefficient and its exponential transformation were −0.16 and 0.85, respectively. Thus, increasing RI by one on average lead to 15% reduction in the number of items administered.

The number of rules that predict a particular response (NRP) had a negative main effect coefficient (−0.3586). However, it also had significant interactions with L, RI, and item selection methods, all of which had positive coefficients. Thus, the positive effect of NRP on efficiency tended to be suppressed as L and/or RI increased or item selection was optimized. The coefficient for NRP that takes into account all interactions is

$$\text{Coef(NRP)} = -0.3586 + 0.0099(L) + 0.0194(\text{RI}) + 0.0911I(\text{GDI}) + 0.1109I(\text{ShE}). \quad (4.9)$$

Table 4.6 shows that all estimated coefficients were negative, and thus increasing the number of rules that predict a particular response was always effective to increase efficiency. The mean coefficient and its exponential transformation were −0.16 and 0.85, respectively. Thus, increasing NRP by one on average leads to 15% reduction in the number of items administered.

Item discrimination (ID) had a negative main effect coefficient (−4.7651) and a positive interaction with L (0.1593). Thus, it had a positive effect on increasing efficiency but the effect was suppressed as L became larger. The equation for the coefficient of ID for various values of L is

$$\text{Coef(ID)} = (-4.7651 + 0.1593(L)) \times 0.10, \quad (4.10)$$

where 0.10 is multiplied to adjust the coefficient so that it represents the amount of improvement when ID is increased by .10. Table 4.6 shows that all resulting estimated coefficients were negative, and thus increasing probabilities of rule-predicted responses was always ef-

fective to increase efficiency. The mean coefficient and its exponential transformation were -0.37 and 0.69, respectively. Thus, increasing ID by 0.10 on average lead to approximately 31% reduction in the number of items administered.

Item selection methods, indicated by GDI and ShE as relative effects to RND, had negative coefficients (-0.2843 and -0.2798, respectively). This is a natural result because adaptive testing is intended to reduce the number of items administered. They interacted with L, RI, and NRP. The interactions with L were negative, implying that adaptive item selection was more effective as L became larger. The interactions with RI and NRP were positive, and this means that the effectiveness of adaptive testing was reduced as either more responses were interpreted by rules or more rules were associated with each response, or both. ShE had larger interactions than GDI and thus was more sensitive to these other factors. Distributions of their coefficients were computed by the following equations:

$$\text{Coef(GDI)} = -0.2843 - 0.0823(L) + 0.0524(\text{RI}) + 0.0911(\text{NRP}), \tag{4.11}$$

$$\text{Coef(ShE)} = -0.2798 - 0.1234(L) + 0.0800(\text{RI}) + 0.1109(\text{NRP}). \tag{4.12}$$

Table 4.6 indicates that adaptive item selection was highly effective to reduce the number of items administered. On average, GDI reduced the number of items by 28%, and ShE by 38%.

To summarize, item characteristics and item selection methods all affected efficiency to a large extent. Increasing the number of cognitive rules required more items, and even more items were needed when more rules were associated with responses and/or probabilities of rule-predicted responses were higher. All variables other than the number of cognitive rules increased efficiency in spite of their interactions with other variables. Increasing by one the number of rule-predicted responses or the number of rules that predict a particular response lead to approximately 15% reduction in the number of items

administered on average. The number of rules that predict a particular response was supposed to be inversely related to response discrimination, which is one of the desired properties for diagnostic items. However, the results indicated that more rules should be associated with responses, suggesting that it would be better to have responses predicted by more rules rather than to try to make each response predicted by a single rule. Item response probabilities were also responsible for improved efficiency; increasing them by 0.10 reduced the number of items by approximately 31%.

Chapter 5

Study 2: Adaptive Testing Simulation for Siegler's Balance Scale Task Data

5.1 Introduction

Although Study 1 was useful in examining how characteristics of diagnostic items systematically affect diagnostic efficiency, it was also worth investigating the extent of efficiency improvement with real items and response data in order to show the practicality of diagnostic items. Moreover, the results of Study 1 could be used to consider the possibility of further efficiency improvement for a given set of items by referring to diagnostic characteristics (i.e., RI, NRP, and ID) of those items. For these purposes, the study in this chapter used a real dataset from the Balance Scale Task (Siegler, 1981), which is intended to assess student understanding of the physical concept of torque (see section 2.1.1).

The procedure of Study 2 was basically the same as Study 1; efficiency improvement was examined by comparing the numbers of items administered between the response types

(multiple choice and dichotomous) obtained by adaptive testing simulations. Prior to the adaptive testing simulation, however, item response probabilities had to be estimated for an appropriate set of cognitive rules. Thus, the first half of Study 2 was devoted to finding a best fitting model. In the second half, efficiency improvement by adaptive testing simulations was examined.

5.2 Description of Procedure

The data used in this study consisted of multiple choice responses of 719 students of age 4 to 17 (mean age 11.2; 312 males and 407 females) to 20 balance scale items. Each balance scale item has a certain number of weights at a certain location on each side of a balance scale, and students predict which side will go down. Thus, each item is a multiple choice item with three response options: Left, Right, and Balance. The 20 items used in this study were classified into four types based on Siegler's (1981) classification: Distance (D; items 1 through 5), Conflict Weight (CW; items 6 through 10), Conflict Distance (CD; items 11 through 15), and Conflict Balance (CB; items 16 through 20). Distance items have equal weights on both sides, but their distances are different. All conflict type items involve unequal weights at unequal distances. For CW items, the side with the greater weight goes down, while for CD items the side with the larger distance goes down. For CB items, the scale always balances.

Table 5.1 shows observed response proportions for the 20 items along with their types and configurations. The "correct" response is Left for items 1 through 15 and Balance for items 16 through 20. The first 10 items provided relatively high correct response rates (ranging from .62 to .95). However, proportions of incorrect responses exceeded those of the correct responses for most of the last 10 items; dominant responses were Right for items 11 through 15 and Left for items 16 through 20. The mean proportion correct score for the 20 items over the 719 students was .53 with standard deviation .17. However, the

Table 5.1: Observed Response Proportions for the Balance Scale Data

		Configuration[b]		Response Proportion		
Item	Type[a]	Left	Right	Left	Balance	Right
1	D	(2, 2)	(1, 2)	.72	.25	.03
2	D	(3, 2)	(2, 2)	.75	.21	.04
3	D	(4, 3)	(2, 3)	.79	.18	.03
4	D	(4, 2)	(3, 2)	.74	.21	.04
5	D	(4, 1)	(2, 1)	.76	.21	.03
6	CW	(2, 3)	(4, 1)	.81	.13	.06
7	CW	(3, 2)	(4, 1)	.65	.29	.06
8	CW	(3, 3)	(4, 2)	.62	.30	.08
9	CW	(2, 4)	(4, 1)	.95	.03	.02
10	CW	(1, 4)	(3, 1)	.88	.08	.05
11	CD	(3, 1)	(1, 2)	.19	.17	.64
12	CD	(4, 2)	(1, 4)	.34	.17	.48
13	CD	(4, 1)	(1, 3)	.25	.15	.60
14	CD	(3, 2)	(1, 3)	.41	.20	.39
15	CD	(4, 3)	(2, 4)	.42	.16	.42
16	CB	(1, 3)	(3, 1)	.83	.13	.04
17	CB	(2, 3)	(3, 2)	.53	.35	.12
18	CB	(3, 4)	(4, 3)	.56	.34	.10
19	CB	(1, 4)	(2, 2)	.77	.18	.04
20	CB	(2, 2)	(4, 1)	.46	.23	.31

[a] D = Distance; CW = Conflict Weight; CD = Conflict Distance; CB = Conflict Balance.
[b] Numbers in parentheses indicate the distance from the fulcrum and the number of weights, respectively.
Note. The correct response is Left for items 1 through 15 and Balance for items 16 through 20.

histogram of the proportion correct score indicated bimodality; it had a sharp peak at .25 and another one around .55, which was larger and wider than the first (Figure 5.1).

As presented in section 2.1.1, Siegler (1981) identified his four rules based on detailed observations and experiments with children. However, what cognitive rules exist for the balance scale task is still controversial. Researchers argued that Siegler's four rules are not the only ones that students actually use, and they provided evidence for other possible rules based on observations and statistical analyses. Among those rules are the addition

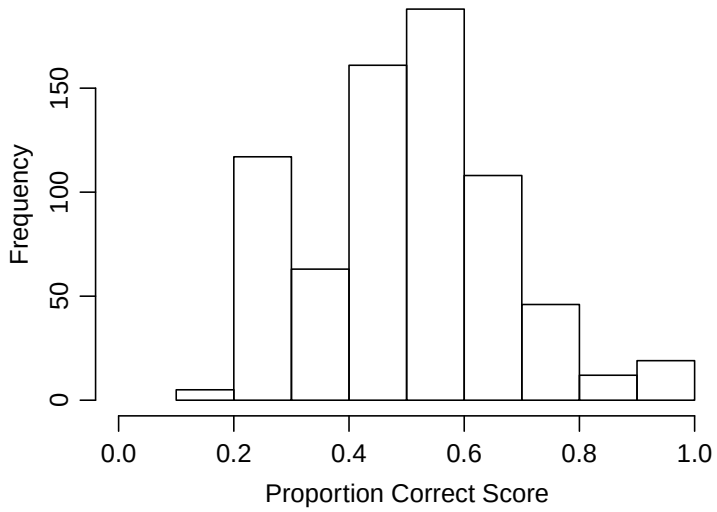

Figure 5.1: Histogram of proportion correct score.

rule and the qualitative proportion rule (Add and QP, respectively; Normandeau, Larivée, Roulin, & Longeot, 1989), the smallest distance down rule (SDD; Siegler & Chen, 1998; Jansen & van der Maas, 2002), the distance dominant rule (DD; Jansen & van der Maas, 2002), and the buggy rule (BG, also called the compensation rule; Boom & ter Laak, 2007; van Maanen, Been, & Sijtsma, 1988). Also, it might be reasonable to assume that some children simply make random guessing for any item (RG), and thus it was considered as one rule. Descriptions of these additional rules along with Siegler's original four rules are shown in Table 5.2, and these rules were considered in the current study. There are other rules of which consistent use was less clear or observed only for a particular child. Such rules, for example, include a set of rules termed Rule 1' (Siegler & Chen, 1998) and the dichotomous encoding rule (also termed the perceptual muddle-through rule or Rule 3A; Klahr & Siegler, 1978). These kind of rules are more idiosyncratic and were not considered in the current study.

Table 5.3 shows expected response patterns corresponding to the 10 rules in Table 5.2. All rules were distinguished from each other by their expected response patterns, and this

Table 5.2: Extended Cognitive Rules for Siegler's Balance Scale Task

Rule[a]	Description
RG	Complete random guessing. Children using this strategy make random guessing on all items.
1	Look at the weights only. Predict that the side with the greater weight will go down. If both sides have the same weight, the beam will balance.
2	Same as Rule 1 when the distances are the same. If the weights are equal but the distances are not, the side with the greater distance will go down. If the weights and distances are both equal, then predict that the beam will balance.
3	Same as Rule 2, but when one side has more (less) weight with a shorter (longer) distance than the other, children "muddle through," possibly leading to random guessing.
4	Compare the torques from both sides by computing the *product* of weight and distance on each side of the scale, and predict that the side with larger torque will go down. This is the correct rule, that is, it always produces a correct response.
Add	Same as Rule 2, but compare the *sums* of weight and distance when one side has greater weight at a smaller distance than the other. The side with the larger sum will go down in that case.
QP	Same as Rule 2, but predict that the scale will balance when one side has more weight at a smaller distance than the other.
SDD	Predict that the side with the *smaller* distance will go down.
DD	Predict that the side with the *larger* distance will go down. This rule is regarded as an alternative to Rule 2 in which distance is taken as a more dominant dimension than weight.
Buggy	If the side with the greater weight has a smaller distance, then the weights on that side are moved farther from the fulcrum until the distances on both sides become equal while one weight is removed every time when the weights are moved to the next farther peg. The answer will be the resulting side with the greater weight.

[a] RG = Complete Random Guessing; Add = Addition; QP = Qualitative Proportion; SDD = Smallest Distance Down; DD = Distance Dominant. Rules 1 through 4 were originally identified by Siegler (1981).

was also true when all responses were dichotomized.

Table 5.3: Configurations and Expected Responses for the Balance Scale Items

	Configuration[a]		Cognitive Rule[b]									
Item	Left	Right	RG	1	2	3	4	Add	QP	SDD	DD	BG
1	(2, 2)	(1, 2)	–	B	L	L	L	L	L	R	L	L
2	(3, 2)	(2, 2)	–	B	L	L	L	L	L	R	L	L
3	(4, 3)	(2, 3)	–	B	L	L	L	L	L	R	L	L
4	(4, 2)	(3, 2)	–	B	L	L	L	L	L	R	L	L
5	(4, 1)	(2, 1)	–	B	L	L	L	L	L	R	L	L
6	(2, 3)	(4, 1)	–	L	L	–	L	L	B	L	R	L
7	(3, 2)	(4, 1)	–	L	L	–	L	B	B	L	R	B
8	(3, 3)	(4, 2)	–	L	L	–	L	B	B	L	R	B
9	(2, 4)	(4, 1)	–	L	L	–	L	L	L	L	R	L
10	(1, 4)	(3, 1)	–	L	L	–	L	L	L	L	R	L
11	(3, 1)	(1, 2)	–	R	R	–	L	L	L	R	L	L
12	(4, 2)	(1, 4)	–	R	R	–	L	R	L	R	L	L
13	(4, 1)	(1, 3)	–	R	R	–	L	R	L	R	L	L
14	(3, 2)	(1, 3)	–	R	R	–	L	L	L	R	L	L
15	(4, 3)	(2, 4)	–	R	R	–	L	L	L	R	L	L
16	(1, 3)	(3, 1)	–	L	L	–	B	L	B	L	R	B
17	(2, 3)	(3, 2)	–	L	L	–	B	B	B	L	R	B
18	(3, 4)	(4, 3)	–	L	L	–	B	B	B	L	R	B
19	(1, 4)	(2, 2)	–	L	L	–	B	L	L	L	R	L
20	(2, 2)	(4, 1)	–	L	L	–	B	R	R	L	R	R

[a] Numbers in parentheses indicate the distance from the fulcrum and the number of weights, respectively.
[b] RG = Complete Random Guessing; Add = Addition; QP = Qualitative Proportion; SDD = Smallest Distance Down; DD = Distance Dominant; BG = Buggy.
Note. L = Left; R = Right; B = Balance. Blank cells indicate that no particular response is associated (random guessing).

5.2.1 Model Selection

Given the 10 rules in Table 5.2, the first step was to find a LCM-DI that best fit the current data, or equivalently to select a set of rules that best accounted for the current data, and then obtain item response probability estimates that were used for the subsequent adaptive testing simulations. Since the number of all possible combinations of these 10 rules were prohibitively large, it was necessary to limit the combinations of rules in advance.

Although statistical analyses have consistently shown strong evidence for Rules 1 and 2, the literature also indicated that most disagreements were found about the validity of Rule 3, which predicts random guessing on conflict items. Among the current set of rules, Add, QP, and Buggy rules were considered as possible alternatives to Rule 3, and thus different combinations of these four rules were examined. SDD and DD were considered as transitional rules from Rule 1 to Rule 2, and thus it was worth considering different combinations of these two rules independently of the above four rules relevant to Rule 3. Accordingly, the model selection proceeded along the following search path: (a) Rules 1, 2, 4, and RG were included in all models, and (b) models with all different combinations of the other "selective" rules (Rule 3, Add, QP, SDD, DD, and Buggy) were tried out. This resulted in 64 models with differing numbers of classes (ranging from a 4-class model without any selective rules to a 10-class model with all 10 rules included). For each given combination of rules, response-by-rule matrices for the 20 items were constructed and the corresponding item response probabilities were estimated under the LCM-DI. Rules 3 and RG predicted random guessing on all or a part of items, and item response probabilities for those items were fixed to .33. The Bayesian information criterion (BIC; Bozdogan, 1987) was used to compare the models; models with smaller BIC were preferred.

The main purpose of the current study was to examine efficiency improvement from using dichotomous responses to using multiple choice responses, and item response probabilities for dichotomized responses were also necessary for this purpose. They were obtained by fitting LCM-DIs to dichotomized data. The original multiple choice data were dichotomized, and LCM-DIs with differing number of classes were fitted. It should be noted that when applied to dichotomous responses, the LCM-DI simplified to the unconstrained LCM except for Rules 3 and RG for which fixed-value constraints were imposed on all or a part of the item response probabilities. Thus, the dichotomous counterparts consisted of (a) six unconstrained LCMs with the number of classes ranging from 5 to 10, in all of which

two classes were always constrained to represent Rules 3 and RG, and (b) six unconstrained LCMs with the number of classes ranging from 4 to 9, in all of which one class was always constrained to represent RG. The first six models were matched with 32 multiple choice LCM-DIs which included Rule 3, and the last six models with 32 multiple choice models which did not include Rule 3 (i.e., for a given number of latent classes, a certain number of multiple choice models were compared to one dichotomous model). These dichotomous models were also compared by BIC.

Having estimated multiple choice and dichotomous versions of LCM-DIs, the next step was to select a best matching pair of multiple choice and dichotomous models that ensured their equivalence as much as possible in order to make a valid examination of efficiency improvement. The best matching pair was sought for by computing mean absolute differences of estimated item response probabilities and marginal rule probabilities between two models. For the item response probabilities, estimates in the multiple choice model were first collapsed to obtain "dichotomized" response probabilities. Then, those dichotomized probabilities were compared to estimates under the corresponding dichotomous model and the mean absolute differences were computed between them. Latent classes were matched between each pair of models so that the overall mean absolute difference was minimized for each pair of models. The mean absolute difference for the marginal rule probabilities was directly computed using the multiple choice model estimates and the dichotomous model estimates with rearranged classes. A model pair with smaller mean absolute differences was preferred. s Goodness of fit for each of the selected pair of models was assessed by the parametric bootstrap method with the likelihood ratio G^2 statistic (e.g., Bartholomew & Knott, 1999, p. 91). The selected models were tested at significance level .05; a p-value larger than .05 was considered providing adequate model fit. Standard errors of logit parameter estimates ($\hat{\alpha}$) were also obtained by the parametric bootstrap.

The final model selection was made by considering the above factors (i.e., smallness of

BIC and mean absolute differences, and goodness of fit) as well as how close the estimated item probabilities were to the patterns predicted by the set of cognitive rules.

5.2.2 Adaptive Testing Simulation

Once the final model pair has been selected, efficiency improvement between the multiple choice and dichotomous versions of final LCM-DIs can be examined by an adaptive testing procedure similar to Study 1. Since adaptive testing was conducted with the given set of 20 balance scale items, the current design is classified into the *post hoc* adaptive testing simulation (a general procedure of post hoc adaptive testing simulations is described by Weiss, n.d., at the CAT Central web site). The item pool consisted of the 20 items as shown in Table 5.3. The same item response data that were used for item parameter estimation were subjected to adaptive testing simulations. The original multiple choice data and their dichotomized counterparts were used with item response probabilities from the corresponding LCM-DIs. Since there was no control over the response-by-rule matrices and item response probabilities unlike Study 1, item selection methods (GDI, ShE, and RND) were the only factor that was manipulated in the current study. Thus, there were six (i.e., two response types times three item selection methods) adaptive testing runs in total. In each run, adaptive testing proceeded as follows:

Starting rule: The first item is randomly selected from the item pool (but the same first item is used for each examinee across different adaptive testing runs).

Item selection: GDI, ShE, or RND

Stopping rule: Testing stops if a posterior probability ($\omega_{il}^{(s)}$) for any rule exceeds .85.

Results from each simulation run were summarized by the median number of items administered (representing efficiency), and the efficiency ratio was also computed between multiple choice and dichotomized versions within the same item selection method. The

average correct classification rate was also computed for each simulation run with the cognitive rule estimated with all 20 items being the "true" rule.

Finally, diagnostic characteristics of the 20 items (i.e., RI, NRP, and ID) were computed from the response-by-rule matrices and item response probability estimates in the final multiple choice model. With this information, recommendations for further efficiency improvement were considered for the current set of items by referring to the results of Study 1.

5.3 Results

5.3.1 Model Selection

Among the 64 multiple choice models, those which did *not* include Rule 3 tended to produce larger BIC and mean absolute differences, and thus the model selection focused on the 32 multiple choice models which included Rule 3. Table 5.4 shows the model comparison results for these 32 models. Rules 1 through 4 and RG were included in all models, and only the selective rules included in each model are shown in Table 5.4. The last two columns show the mean absolute differences of estimated probabilities between the multiple choice and dichotomous models for item response probabilities and marginal rule probabilities, respectively.

Among the multiple choice models, Model 28 with nine rules (Rules 1 to 4, RG, Add, QP, SDD, and Buggy) yielded the smallest BIC (17504.80). This model also had the smallest discrepancy with the corresponding 9-class dichotomous model in terms of item response probabilities (.06) among the 9-class multiple choice models. The p-value for the parametric bootstrap G^2 statistic was .11 for Model 28 and .28 for the dichotomous model, both of which indicated acceptable model fit.

Figure 5.2 depicts estimated item response probabilities by class in Model 28 and the

Table 5.4: Model Comparison Results.

Model	L	Rules Included[a]					BIC-MC	BIC-D	π-DIF	ω-DIF
1	5	(No selective rule included)					18373.95	12427.90	0.01	0.01
2	6	Add					17821.44	12305.11	0.04	0.06
3	6	QP					17998.64	12305.11	0.09	0.08
4	6	SDD					18116.80	12305.11	0.05	0.07
5	6	DD					18200.55	12305.11	0.06	0.05
6	6	BG					17966.60	12305.11	0.08	0.08
7	7	Add	QP				17657.96	12304.63	0.29	0.08
8	7	Add	SDD				17833.81	12304.63	0.19	0.07
9	7	Add	DD				17758.72	12304.63	0.18	0.09
10	7	Add	BG				17788.57	12304.63	0.04	0.04
11	7	QP	SDD				18017.69	12304.63	0.21	0.08
12	7	QP	DD				17942.54	12304.63	0.20	0.08
13	7	QP	BG				18044.10	12304.63	0.04	0.03
14	7	SDD	DD				17946.00	12304.63	0.22	0.13
15	7	SDD	BG				17764.56	12304.63	0.23	0.13
16	7	DD	BG				17913.79	12304.63	0.20	0.11
17	8	Add	QP	SDD			17624.08	12228.48	0.32	0.09
18	8	Add	QP	DD			17705.50	12228.48	0.30	0.07
19	8	Add	QP	BG			17704.58	12228.48	0.05	0.03
20	8	Add	SDD	DD			17771.09	12228.48	0.31	0.06
21	8	Add	SDD	BG			17820.08	12228.48	0.27	0.06
22	8	Add	DD	BG			17771.49	12228.48	0.22	0.06
23	8	QP	SDD	DD			17711.59	12228.48	0.31	0.13
24	8	QP	SDD	BG			17675.15	12228.48	0.30	0.09
25	8	QP	DD	BG			17779.07	12228.48	0.06	0.03
26	8	SDD	DD	BG			17713.99	12228.48	0.32	0.10
27	9	Add	QP	SDD	DD		17664.92	12279.64	0.07	0.04
28*	9	Add	QP	SDD	BG		17504.80	12279.64	0.06	0.04
29	9	Add	QP	DD	BG		17747.68	12279.64	0.17	0.08
30	9	Add	SDD	DD	BG		17780.72	12279.64	0.16	0.07
31	9	QP	SDD	DD	BG		17872.85	12279.64	0.16	0.03
32	10	Add	QP	SDD	DD	BG	17515.84	12350.98	0.29	0.07

* Selected pair of models. [a] Add = Addition; QP = Qualitative Proportion; SDD = Smallest Distance Down; DD = Distance Dominant; BG = Buggy.

Note. L = number of classes; BIC-MC = BIC under the multiple choice model; BIC-D = BIC under the dichotomous model; π-DIF = mean absolute item response probability difference; ω-DIF = mean absolute marginal class probability difference. Rules 1 through 4 and RG were included in all models. There is only one dichotomous model for the same L.

corresponding dichotomous model. Actual item response probability estimates are given in Tables C.1 through C.9 in appendix C. Figure 5.2 also shows estimates of marginal rule probabilities for both multiple choice and dichotomous models. The nine classes are labeled by the intended rules. In each plot in Figure 5.2, item response probability estimates under the multiple choice model are indicated by □ (Left), ◯ (Balance), and △ (Right), and estimates of correct response probabilities under the dichotomous model are indicated by ×. For each class (rule), the multiple choice and dichotomous models provide a good match if × shows the following pattern: × is close to the correct response probabilities for items for which the rule predicts correct responses (i.e., □ for items 1 through 15 and ◯ for items 16 through 20) and close to the incorrect response probabilities for items for which the rule predicts incorrect responses (i.e., either ◯ or △ for items 1 through 15 and either □ or △ for items 16 through 20).

The median of standard errors of the logit item parameter (α) estimates was 0.78 in the multiple choice model, and 4.08 in the dichotomous model. Standard errors for the dichotomous model were much larger than those in the multiple choice model, probably due to the loss of information brought by the dichotomization of responses. In both models, larger standard errors tended to occur as estimates deviated farther from 0.

Although the item response probabilities were structured based on the response-by-rule matrices for the given set of rules, it was still possible that patterns of estimated item response probabilities for the corresponding class did not match the expected patterns. This occurs when the corresponding α estimates are negative, leading to expected response probabilities smaller than non-expected response probabilities. On the one hand, there were clear indications of Rules 1 through 4 and QP where we observe high expected response probabilities and close matches between multiple choice and dichotomous probabilities.

On the other hand, several negative α estimates were found in the other classes. Most problematic were the classes intended for the SDD and Buggy rules. For the SDD class,

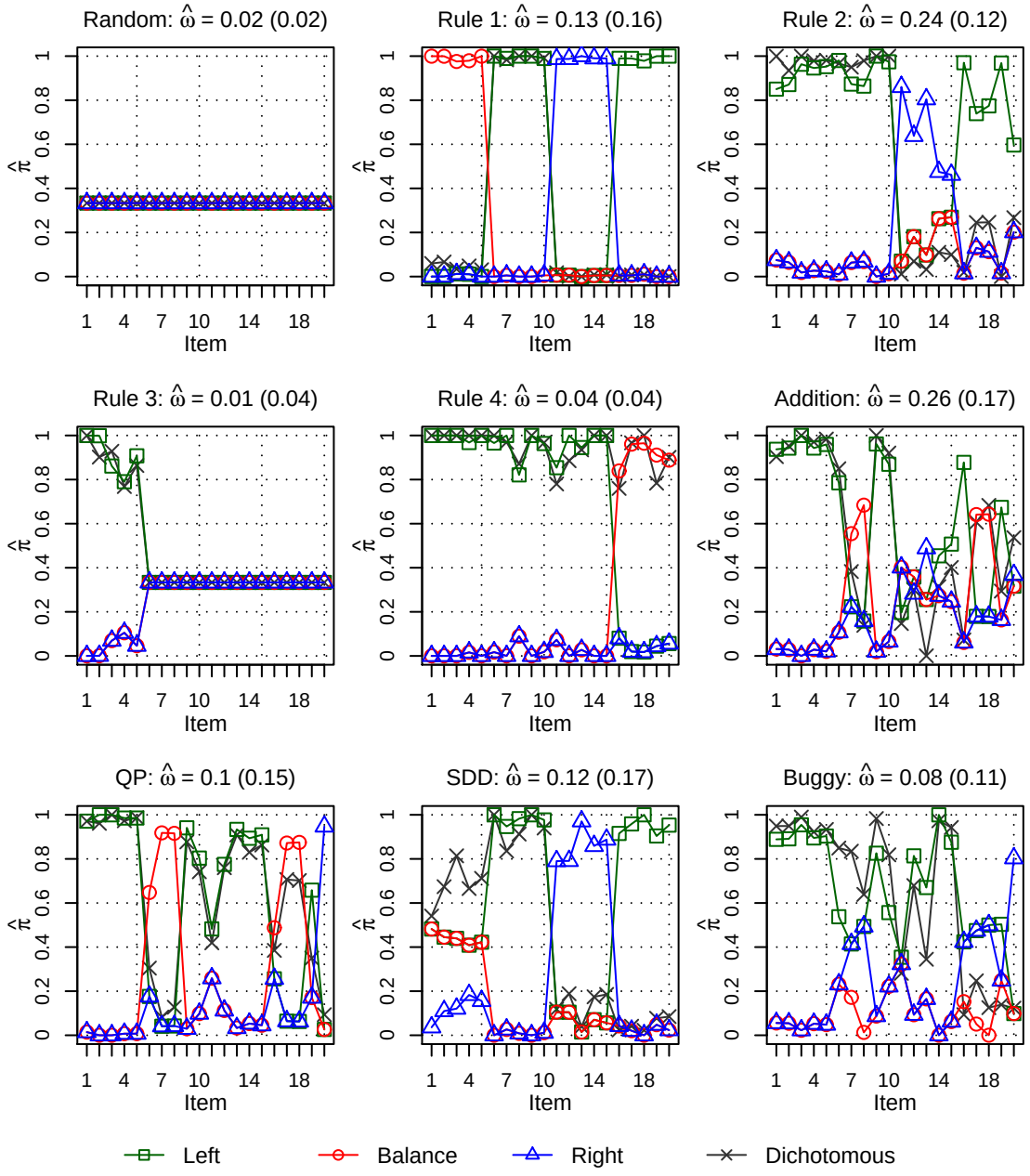

Figure 5.2: Plots of estimated item response probabilities ($\hat{\pi}$). For each plot, $\hat{\omega}$ indicates the estimated class size (class sizes under the dichotomous model are shown in parentheses). The $\times$ symbol indicates the estimated probability of a correct response under the dichotomous model, and the other symbols represent estimated response probabilities under the multiple choice model. The correct response is "Left" for items 1 through 15, and "Balance" for items 16 through 20. Response probabilities of all items for Random and those of items 6 through 20 for Rule 3 were fixed to .33.

the multiple choice probability estimates for the first five items were strongly negative and also had relatively large discrepancies with the corresponding dichotomous probability estimates. As a result, the pattern of item response probabilities in this class resembled that of Rule 2 (the only difference between the two rules was the expected responses to the first five items), and these two rules might not be separated well. In order to see whether these two rules could be collapsed into one rule, Model 19, in which the SDD rule was removed from Model 28, was examined. However, item response probability estimates did not match the expected patterns very well in Model 19, while agreement between estimated multiple choice and dichotomous probabilities was better than for Model 28. Also, BIC for Model 19 was not very small compared to other models (17704.58), and the p-value for the parametric bootstrap G^2 was barely acceptable ($p = .05$). This indicated no strong supporting evidence that these two rules could be integrated into one rule. Thus, this class was retained but might be relabeled as Rule 2', which might be interpreted as applying Rule 2 less consistently than in the "true" Rule 2.

For the class intended for the Buggy rule, negative α estimates were observed for items 7, 8, 16, 17, and 18, in which the expected responses were all Balance. A possible interpretation was that students in this group might use the Buggy rule but they tried to avoid Balance responses. Another possible reason was the similarity between the QP and Buggy rules. Expected response patterns for these rules were almost the same except for item 6 for the current set of items (see Table 5.3), and it was possible that these rules were not separated well in the intended manner (e.g., QP was dominant to Buggy and probability estimates for the Buggy rule might be distorted so that these two classes became as different as possible). However, removing the Buggy class from Model 28 did not resolve the problem (Model 17); BIC substantially increased for Model 17 and the mean absolute difference for item response probabilities was too large to accept. Also, relatively large discrepancies between multiple choice and dichotomous probability estimates were found

for items 6, 7, and 13. It should also be noted that in terms of the class size (ω), relatively large discrepancies were found for the classes intended for Rule 2 (.24 vs. .12) and Add (.26 vs. .17), even though discrepancies of item response probabilities in these classes were small.

Large deviations of estimated item response probabilities from the expected pattern (i.e., negative α values) were threats to the validity of hypothesized cognitive rules. The above results indicated that further validation of cognitive rules were necessary, although there were limitations with the current items and data (e.g., limited number of items and subjects for the number of rules). Discrepancies found between multiple choice and dichotomous model estimates were probably more important for the purpose of the current study, because they likely invalidated the assessment of efficiency improvement from dichotomous to multiple choice responses. In spite of these issues, Model 28 and the corresponding dichotomous model were accepted for the final models in the present study, and they were used for the subsequent adaptive testing simulations because of their relatively better fits and closer match than the other pairs of models.

Figure 5.3 indicates some validity of the nine rules represented by Model 28. The figure shows distributions of students' age by their "true" rule usage, which was estimated from their responses to all 20 items. Locations of the age distributions well approximated the hypothesized developmental order of the cognitive rules. In terms of Siegler's four rules, Rules 1 to 4, Rule 1 located at the lowest place on the age scale, Rule 4 at the highest, and Rules 2 and 3 in the middle, although Rules 2 and 3 had a large overlap. SDD, by its definition, is considered as a transitional rule from Rule 1 to Rule 2, and its distribution was located between those of Rules 1 and 2 as expected. Add, QP, and Buggy are possible variants of Rule 3, and central locations of these rules were close to that of Rule 3 (Add and QP were located a little higher than Rule 3, which made sense, while Buggy had a lower center and wider tails, which might indicate its unwanted similarity to Rule 2 as

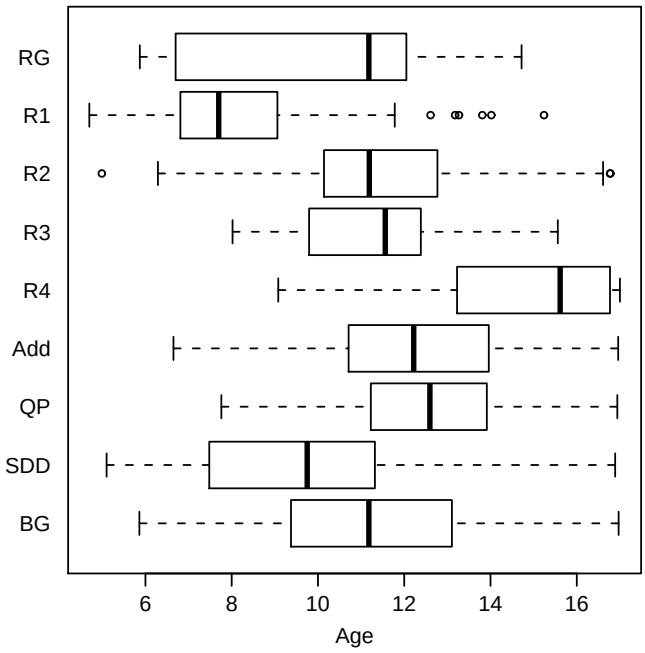

Figure 5.3: Boxplot of age by rule usage estimated from responses to all 20 balance scale items. The vertical bar in the middle of a box indicates the median, a box indicates the interquartile range, and whiskers cover the central 98% range. RG = Complete Random Guessing; R1 = Rule 1; R2 = Rule 2; R3 = Rule 3; R4 = Rule 4; Add = Addition; QP = Qualitative Proportion; SDD = Smallest Distance Down; BG = Buggy.

mentioned above).

Test GDIs (i.e., the sums of $\mathrm{GDI}_j(l)$ over the 20 items for each rule) for the final models are shown in Table 5.5. As expected, GDIs in the multiple choice model were larger than those in the dichotomous model. Thus, use of multiple choice responses brought more information for discriminating the rules than dichotomous responses. Within each model, larger GDIs indicate that those rules were, on average, better distinguished from the other rules. For the multiple choice model, GDI for Rule 2 (66.61) was smaller than the others. For the dichotomous model, Rules 2, Add, SDD, and BG had relatively small GDIs (44.69, 49.14, 50.78, and 45.81, respectively). It was anticipated that these rules were relatively hard to diagnose, but the test as a whole maintained the ability to distinguish all nine rules

Table 5.5: Test GDI for the Balance Scale Items

	Cognitive Rule[b]								
Model[a]	RG	1	2	3	4	Add	QP	SDD	BG
MC	116.26	141.36	66.61	94.51	101.64	76.56	105.64	91.32	82.04
D	101.59	106.47	44.69	73.69	84.82	49.14	73.69	50.78	45.81

[a] MC = multiple choice model (Model 28); D = dichotomous model.
[b] RG = Complete Random Guessing; Add = Addition; QP = Qualitative Proportion; SDD = Smallest Distance Down; BG = Buggy.

Table 5.6: Summary of the Number of Items Administered by Response Type and Item Selection Method

Condition[a]	Min	25%	Median	Mean	75%	Max	NR[b]	CR[c]
RND-M	3	8	11	12.08	16	20	.11	.91
RND-D	3	10	14	13.56	18	20	.15	.96
GDI-M	3	5	10	10.48	15	20	.13	.90
GDI-D	3	7	11	11.54	16	20	.16	.96
ShE-M	3	4	6	7.47	9	20	.06	.88
ShE-D	3	5	7	9.02	12	20	.10	.94

[a] GDI = global discrimination index item selection; ShE = Shannon entropy item selection; RND = random item selection; M = multiple choice model; D = dichotomous model.
[b] NR = proportion of students who did not meet the adaptive testing stopping criterion.
[c] CR = correct classification rate.

fairly well.

5.3.2 Adaptive Testing Simulations

Table 5.6 shows the summary of the number of items administered under each adaptive testing condition. The table also shows the proportions of students who did not meet the stopping criterion (NR) and the correct classification rates (CR). The correct classification rates were computed by assuming that cognitive diagnosis by all 20 items was "true" (these true rules were estimated separately for multiple choice and dichotomous scoring by using respective models). The results are also visualized by the boxplot in Figure 5.4.

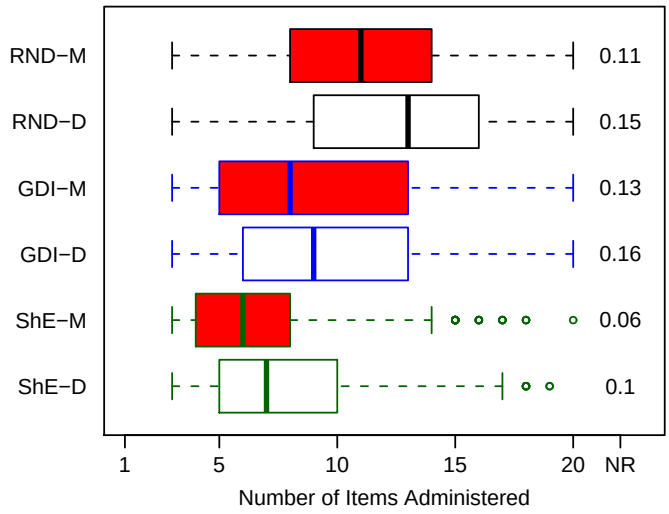

Figure 5.4: Boxplot of the number of items administered by adaptive testing condition. The vertical bar in the middle of a box indicates the median, a box indicates the interquartile range, and whiskers cover the central 98% range. GDI = global discrimination index item selection; ShE = Shannon entropy item selection; RND = random item selection; M = multiple choice model; D = dichotomous model; NR = proportion of students who did not meet the adaptive testing stopping criterion.

Table 5.6 and Figure 5.4 clearly show that the quantiles and means for multiple choice responses were smaller than for dichotomous responses in all three item selection methods, indicating that the former was more efficient. The median number of items administered was reduced by one (GDI and ShE) to three (RND) items by considering multiple choice responses. Efficiency improvement was also indicated by the smaller NRs; more students reached the final diagnosis when multiple choice responses were used. However, item selection methods had much larger effect on efficiency than the response type. Reduction in the median number of items from RND was 9% (from 11 to 10 items) for GDI and 45% (from 11 to 6) for ShE for multiple choice responses, but it was even larger for dichotomous responses: 21% (from 14 to 11) for GDI and 50% (from 14 to 7) for ShE. As shown by Xu et al. (2003) and in chapter 4, ShE substantially outperformed GDI. Correct classification rates for all conditions exceeded .90 except for ShE-M, and thus misdiagnosis due to adap-

tive testing was considered minimal. However, dichotomous scoring yielded higher correct classification rates than multiple choice scoring in all item selection conditions.

Table 5.7: Summary of Efficiency Improvement by Item Selection Method

Condition[a]	Min	25%	Median	Mean	75%	Max
RND	0.19	0.60	0.85	0.95	1.13	5.00
GDI	0.20	0.60	0.80	0.89	1.00	5.00
ShE	0.20	0.60	0.80	0.95	1.00	6.00

[a] GDI = global discrimination index item selection; ShE = Shannon entropy item selection; RND = random item selection.

Table 5.7 shows the summary of efficiency ratio by item selection method in detail. The median efficiency ratio was 0.85, 0.80, and 0.80 for RND, GDI, and ShE, respectively. Thus, within the same item selection method, use of multiple choice responses decreased the number of items by approximately 20%, which is a substantial reduction. Although the item selection methods directly affected the efficiency itself (i.e., the number of items), they made little difference in efficiency *improvement*. The 75 percentile is 1.13, 1.00, and 1.00 for RND, GDI, and ShE, respectively. Thus, use of multiple choice responses was effective in reducing the number of items for approximately 75% of students in all adaptive item selection methods. However, this also implies that efficiency became worse for the other 25% of students. One possible explanation for this is the discrepancies in item response probabilities between the multiple choice and dichotomous models; some items with low discrimination (i.e., low GDI) in the multiple choice model could have better discrimination in the dichotomous model due to those discrepancies. However, efficiency larger than 1.0 was also observed in the results of Study 1, where there was no discrepancies between multiple choice and dichotomous models. In what situations use of multiple choice responses worsens efficiency is left to future research.

From these observations, it is concluded that the full use of diagnostic items substantially improved the efficiency of cognitive diagnosis for the balance scale items. However,

conventional dichotomous scoring worked fairly well in terms of efficiency if the test was administered adaptively. These results were consistent with Study 1.

One thing to note is that comparison for efficiency improvement in the current study had limited validity due to the discrepancies between the multiple choice and dichotomous models on which the adaptive testing simulations were based. Due to these discrepancies, what each class represented were probably different and the two models were likely measuring somewhat different sets of rules. In fact, agreement between the "true" rules from these models based on the 20 items was .75, although the correct classification rates *within* each model exceeded .90 in almost all cases.

5.3.3 Item Characteristics

Table 5.8 shows diagnostic characteristics of the 20 balance scale items based on the nine rules retained in the final multiple choice model. Because these items, which consisted of the four types (D, CW, CD, and CB), were originally intended to distinguish Siegler's four rules, they might not be optimal for the nine rules that were considered in the current study. Compared to the number of rules ($L = 9$), NRP was at a satisfactory level; almost all rules predicted a particular response in each item. However, RI and ID could be made better for improved efficiency.

In terms of RI, only six items had the full response interpretability (i.e., RI = 3) and RI ≤ 2 for the other 14 items. Especially, items 9 and 10, for which RI = 1, were almost of no use for the given set of rules, even though they could still contribute to distinguishing "random guessing" rules (Rules 3 and RG) from the others. These items can be replaced by items with larger RI. Since the number of configurations for balance scale items is limited ($4^4 = 256$ combinations with the four-peg balance scale and with the maximum number of weights being 4), it is possible to list all configurations, for each of which RI (and other characteristics) is computed. In this manner, one can choose a best set of items for a given

Table 5.8: Diagnostic Characteristics of the Balance Scale Items.

Item	RI	NPR	ID	Item	RI	NPR	ID
1	3	8	.84	11	2	7	.65
2	3	8	.85	12	2	7	.76
3	3	8	.86	13	2	7	.83
4	3	8	.84	14	2	7	.81
5	3	8	.86	15	2	7	.80
6	2	7	.85	16	2	7	.75
7	2	7	.78	17	2	7	.74
8	2	7	.75	18	2	7	.75
9	1	7	.96	19	2	7	.80
10	1	7	.88	20	3	7	.79
				Mean	2.20	7.25	.81

Note. RI = response interpretability; NPR = number of rules that predict a particular response; ID = item discrimination. The total number of rules is $L = 9$, and the number of response options is $K = 3$.

set of rules. In more general situations where there is more flexibility for response options, one can simply replace "idle" responses by rule-predicted ones.

The mean ID over the 20 items was .81, which corresponded to Low to Medium condition in Study 1. However, there was variability across rules as well as items. As shown in Figure 5.2, estimated probabilities for the expected responses were very close to 1 for some rules such as Rules 1, 3, and 4, while lower estimates were found for other rules such as Add, SDD, and Buggy. It would be helpful in improving IDs to consider why these latter rules did not clearly show up. These rules might not be fully detected with the current data because the items used did not represent these rules well (e.g., the expected response patterns of QP and Buggy were almost the same), or the sample size was too small for conforming response patterns to consistently appear in the data. This is related to the above issue of selecting a best set of items for a given rule; estimation of item response probabilities could be improved with more appropriate set of items for a given set of rules and a large-enough sample of examinees. It is also possible to speculate that smaller ID might be inherent in

these rules, that is, some of these rules might be difficult to "successfully" apply (e.g., the Buggy rule involves more operations of moving weights than simply counting or operating numbers, which might make the rule more prone to unintended errors). Improving ID would be difficult if this is the case, but continuing effort to validate these rules would be crucial to evaluate and estimate item response probabilities properly.

Chapter 6

Conclusions

6.1 Summary and Conclusions

This thesis evaluated potential usefulness of diagnostic items for cognitive diagnosis in terms of the following research questions: (a) to what extent does the use of diagnostic items improve the efficiency of cognitive diagnosis? and (b) what aspects of diagnostic items are more responsible for the efficiency of cognitive diagnosis than others? Two studies were conducted to address these questions. In Study 1, efficiency improvement brought by diagnostic items and how the characteristics of diagnostic items affect efficiency were examined by using hypothetical items and adaptive testing simulations. In Study 2, adaptive testing simulations were applied to Sigler's balance scale task data to examine efficiency improvement in real items. In this thesis, efficiency of cognitive diagnosis was defined as the number of items administered to make a cognitive diagnosis, and efficiency improvement as the ratio of two such efficiencies, one from adaptive testing in which dichotomous responses were used and the other in which multiple choice responses were used.

Both Studies 1 and 2 showed that efficiency improvement brought by diagnostic items was substantial. Study 1 showed that the number of items was reduced by 14 to 78% with

average 42%. It also revealed that the amount of improvement was largely affected by item characteristics and item selection methods in adaptive testing. Although these variables interacted with each other to make simple interpretation difficult, the results of Study 1 suggested that we are more likely to benefit from using multiple choice responses in diagnostic items especially when (a) more cognitive rules are involved in a test and/or (b) more responses are associated with cognitive rules in each item, while efficiency improvement can be lowered when (a) more cognitive rules are associated with responses in each item, (b) probabilities of rule-predicted responses are higher, and/or (c) items are selected adaptively. Study 2 showed that efficiency improvement was substantial also in real items; the amount of improvement ranged from 15 to 20% with the balance scale data. These results strongly encourage the use of diagnostic items for more efficient cognitive diagnosis.

Study 1 also examined how characteristics of diagnostic items affect diagnostic efficiency. The results indicated that more items are required as more cognitive rules are involved (20% more items for one addtional rule), but this can be made up or even turned over by (a) making more responses interpretable by cognitive rules in each item, (b) associating more cognitive rules with responses in each item, and/or (c) having higher probabilities for rule-predicted responses (i.e., high item discrimination). The positive effects of these characteristics were quite comparable. Making one more response interpreted by rules and associating one more cognitive rule with responses both lead to 15% less number of items on average. Thus, in practice, it is always recommended to make as many responses interpreted by cognitive rules as possible and making as many cognitive rules associated with responses as possible. At the same time, it is also important to ensure that response-by-rule matrices are made as different as possible among items (this point was not very explicit in Study 1, in which generation of response-by-rule matrices involved random assignment of rules to responses). For example, associating two cognitive rules with the same response in all items will not help differentiate these rules; they can be associated with the same response

in some items but should be tied to different responses in other items.

Probabilities of rule-predicted responses reduced the number of items by 31% on average if they were raised by .10. However, it may not always be possible to achieve high probabilities of rule-predicted responses, because they have potentially controllable and uncontrollable (i.e., more inherent to cognitive rules) aspects. The bottomline is to set up a situation in which item response probabilities are evaluated in a proper manner. This can be made by establishing validity of cognitive rules and ensuring the appropriateness of items for eliciting the use of intended cognitive rules; less valid cognitive rules may not appear clearly as observed responses, while inappropriate items likely fail to manifest the underlying rule usage. In practical settings in which item response probabilities must be estimated with a sample of items and examinees, we also need to make sure (a) that an appropriate set of items are being used so that cognitive rules are well differentiated by their expected response patterns and (b) that there are a large number of examinees from an appropriate population so that at least the minimum number of response patterns conforming to each cognitive rule (given that the rule is valid) are observed in order for item response probabilities to be estimated with a certain level of accuracy.

6.2 Limitations of the Studies

Although the two studies in this thesis showed the usefulness of diagnostic items, the ways in which these studies were conducted may limit the validity of the results. These limitations are epitomized into the following issues: (a) validity of simulation settings in Study 1, (b) validity of efficiency comparison in Study 2, and (c) validity of LCM-DI as a psychometric model for diagnostic items.

6.2.1 Validity of Simulation Settings in Study 1

In Study 1, several item characteristics were systematically varied to examine their effects on efficiency of cognitive diagnosis. However, validity of the results depends on the extent to which the settings of item characteristics reflected reality. It is difficult to make a proper judgment about this issue without a general reference, but comparing the item characteristic settings in Study 1 with the setting in Study 2, which dealt with real items and responses, can provide some clues.

Item characteristics related to the response-by-rule matrix are the number of cognitive rules (L), response interpretability (RI), and the number of rules that predict a particular response (NRP). Although direct comparisons of these characteristics between Studies 1 and 2 are not easy because of the different number of response options, the average of these item characteristics in Study 2 (i.e., $L = 9$, RI = 2.20, and NRP = 7.25; see Table 5.8) was well within the range covered in Study 1. However, RI and NRP varied across individual items to some extent in Study 2, while in Study 1 items within each subset shared the same values on these variables. Thus, within each item subset in Study 1, items might be too uniform compared to items included in real tests. On the contrary, the random assignment of rules to responses in each item in Study 1 might introduce too large variation among response-by-rule matrices, which could inflate diagnostic power of items and result in spurious efficiency improvement. In the balance scale items, in contrast, the number of ways in which rules were assigned to responses was fairly limited because the response options were fixed for all items, and as a result response-by-rule matrices were more "uniform" across items (e.g., response-by-rule matrices were all the same for the first five items despite the different configurations; see Table 5.3). This kind of inflexibility in the way in which response-by-rule matrices are constructed may occur in other practical situations as well.

Another item characteristic is item discrimination (ID), which is the average probability

of rule-predicted responses. Again, the average ID for the balance scale items in Study 2 (.81) was within the range considered in Study 1 (see Table 5.8), but ID across items had larger variability (ranging from .65 to .96) than those considered in each item subset in Study 1 (within each ID condition, the range was .10). Thus, ID can be more heterogeneous in real tests than considered in Study 1. Related to ID, test difficulty for item subsets in Study 1 showed discrepancies with that of Study 2. The test difficulty in Study 2 was .53, but the average difficulty in Study 2 was much smaller (.31). Thus, the tests considered in Study 1 were fairly difficult. Given that the ID values used in Study 1 did not deviate too much from those observed in Study 2, these discrepancies in test difficulty was most likely due to the uniform marginal class probabilities (i.e., equal ω for all rules) assumed in Study 1; marginal rule probability estimates in Study 2 ranged from .01 to .26, indicating that there was substantial variation across rules.

In sum, Study 1 had some validity in the sense that average item characteristics fell within the range that might be observed in actual settings. However, real items can have much larger variation within a test (while they may have more uniform response-by-rule matrices in some aspects as seen in the balance scale items), and marginal rule probabilities can also vary to a greater extent in practice. Although Study 2 had its own limitations, the possible consequence of these discrepancies is that efficiency improvement in real settings can be smaller than observed in Study 1. There was no such case in Study 1 that item characteristics exactly matched the average values for the balance scale items (Table 5.8), but consider the case in which $L = 9$, RI = 3, NRP = 7, and ID = .85 in Study 1. In this condition, the median efficiency ratio was 0.44 for RND, 0.50 for GDI, and .63 for ShE, all of which are much smaller than for the balance scale items for which the corresponding efficiency ratios were 0.85, 0.80, and 0.80.

6.2.2 Validity of Efficiency Comparison in Study 2

Study 2 compared efficiency between multiple choice and dichotomous responses in terms of the balance scale items. However, the comparisons were made without fully establishing the equivalence between the multiple choice and dichotomous LCMs on which the comparisons were made, and thus the results should be taken with caution.

In Study 2, two versions of LCMs were fitted to the corresponding multiple choice and dichotomous data, and item response probability estimates from the two models were used in adaptive testing simulations. The study was intended to reproduce efficiency difference that two test users would observe in reality, if one of them used multiple choice responses and the other used dichotomous responses for the same items without referring to each other's result. For this purpose, the study did not use "dichotomized" item response probabilities which could be obtained by simply collapsing multiple choice item response probability estimates as in Study 1.

While most of the estimates of item response probabilities matched fairly well between the multiple choice and dichotomous models, non-matching estimates were found for some items and rules and also for class sizes. As a result, the proportion of agreements among examinees' estimated rules from the two models, based on the all 20 items, was .75. Thus, the two models were probably measuring somewhat different cognitive rules.

In LCM-DI, cognitive rules are defined and distinguished only by the parameterization based on their expected response patterns, whether they are multiple choice or dichotomous. Dichotomization likely changes, or makes less clear, the operational definitions of cognitive rules by collapsing "incorrect" response options into one incorrect response in each item. For example, suppose that two rules predict different incorrect responses A and B in an item. In the multiple choice model, these rules are defined such that they predict these different responses in this item. When the responses are dichotomized, however, both rules only predict the same "incorrect" response and their more detailed "definitions" in the

multiple choice model are lost. Thus, discrepancies that were found for item response probabilities and class sizes are considered as a natural consequence of dichotomization.

Although the operational definitions of cognitive rules by their expected response patterns in a dichotomous model cannot be finer than those in the corresponding multiple choice model, they will become closer as more items are included in a test. In this respect, the 20 items in Study 2 might be too short to ensure that the cognitive rules were defined close enough.

6.2.3 Validity of LCM-DI as a Psychometric Model for Diagnostic Items

Both Studies 1 and 2 used the LCM-DI as the base model for probabilistic structures behind observed item responses. Assumptions of the LCM-DI, and LCM in general, might be too restrictive to reflect the actual student response behaviors and thus can be a threat to the validity of the results.

Fundamental assumptions of LCM include that (a) within each cognitive rule item response probabilities are fixed for all students (i.e., the homogeneity assumption), (b) cognitive rules included in a model are mutually exclusive and exhaustive, and (c) each examinee keeps using the same cognitive rule for all items. These assumptions can be too simplistic to reflect the reality. First, response tendencies within rules may not be homogeneous over examinees. Even if they are using the same rule, some students may produce responses that closely match the expected response patterns, but others may not due to, for example, inefficient access to knowledge and skills (Gitomer & van Slyke, 1988; also see Pirolli & Wilson, 1998).

Second, all possible cognitive rules are not always covered by a single LCM, while LCM requires that latent classes exhaust all rules. In practice, it is almost impossible to include all cognitive rules in a model. A possible outcome is biased estimates of item response probabilities and validity of the cognitive diagnosis is also threatened.

Third, examinees may not keep using the same cognitive rules during testing on the contrary to the assumption of consistent use of single rules. One such example is strategy switching; students actively change their strategies depending on contents and features of items, and they even change their strategies as a result of learning during testing (e.g., Snow & Lohman, 1993; Klahr and Siegler, 1978, also pointed out this issue for the balance scale task). This kind of violation raises a problem in item parameter estimation as well as in diagnosis of cognitive rules. If each observed response pattern is a result of mixing strategies, fewer response patterns match any of the expected response patterns even though all rules are included in the model and the "true" expected response probabilities associated with those rules are high. A possible remedy is to set up cognitive rules such that they include patterns of strategy switching (i.e., which strategy is used for what type of items) in their definitions. However, it would be difficult to predict student strategy switching behaviors.

Finally, constraints on item response probabilities in LCM-DI can be too restrictive, depending on the nature of cognitive rules of interest. For simplicity and compatibility with the response-by-rule matrix, LCM-DI assumes that responses that are not predicted by a rule is chosen randomly. However, it is possible that a rule prefers a particular response (other than the one it predicts) to others.

As a result, generalizability of the results in Studies 1 and 2 can be very limited. Also, violation of these assumptions could explain why some rules did not appear clearly in Study 2.

6.3 Future Directions

Given the above limitations, future research should be directed to (a) continuing validation of cognitive rules and detailing how students use cognitive rules in various domains and (b) developing CDPMs that are compatible with diagnostic items and better approximate

student response behaviors relevant to cognitive rule usage. Better understanding of cognitive processes and student response behaviors contributes to the construction of better cognitive models and to the development of more valid CDPMs. Although there is a huge literature for basic psychological research on student thinking and learning, research on how findings from the basic research can be applied to construction of cognitive models for CDA is relatively scarce and more studies in this direction will be needed (Leighton & Gierl, 2007c). This line of research could also provide a reference that helps setting up more realistic population parameters (e.g., item response probabilities, class sizes, and structure of response-by-rule matrices) in simulation studies like Study 1 in this thesis.

Further development of CDPMs for diagnostic items should incorporate findings from psychological research on student response behaviors and also address the limitations of LCMs stated in the previous section. Regarding the homogeneity assumption of LCM, mixture IRT models approach the problem by combining a LCM and an IRT model. These models are also called "IRT-within-LCM" models; cognitive rules are represented by latent classes as in usual LCM, but within each class item response probabilities follow some IRT model in order to account for heterogeneity within classes. Models in this class include the mixed Rasch model (Rost, 1990, 1991), the mixed linear latent trait model (Mislevy & Verhelst, 1990), the Saltus model (Mislevy & Wilson, 1996; Wilson, 1989), and the HYBRID model (Yamamoto, 1989), all of which work with dichotomous responses. A mixture IRT model for multiple choice responses was also proposed by Bolt, Cohen, and Wollack (2001). The existing mixture IRT models, however, do not explicitly incorporate the structure between cognitive rules and their expected responses; the current models only express that probabilities of correct responses systematically vary by rule.

Insufficient coverage of cognitive rules in a LCM is almost inevitable in practice. Thus, the problem is rather how to handle the impact of missing cognitive rules on item response probability estimation and resulting cognitive diagnosis. One direction to this problem is

to examine the robustness and practical usefulness of a model. A model can be accepted for practical use as long as item response probabilities are estimated without much bias and cognitive diagnosis that the model provides is valid, that is, most students are diagnosed as they should be. In the context of binary skills models, for example, Rupp and Templin (2008) reported that eliminating attributes lead to overestimation of false negative rates of individual items and higher misclassification rates in the DINA model. Further investigation about robustness is necessary, especially for CDPMs that address multiple choice responses. Another direction to this issue is to introduce an additional latent variable that absorbs the impact of unidentified rules. For example, RUM adds to the NIDA model a variable that represents the residual ability that is not accounted for by cognitive attributes. The residual ability in RUM adjusts each item response probability through the Rasch item response function. However, whether this kind of extension is possible for multiple choice responses is not yet known.

CDPMs that address students' inconsistent use of cognitive rules are very scarce in the literature. One such model is the generalized solution-error response-error model (GSEREM; Rijkes & Kelderman, 2007; Westers & Kelderman, 1991). GSEREM assumes a two-step problem-solving process: the strategy selection process and the response process. Each student first selects a strategy (cognitive rule) when they are given an item (strategy selection process), and then chooses a response using the selected strategy (response process). In contrast to mixture IRT models, GSEREM may be termed a "LCM-within-IRT" model, because in GSEREM strategy selection probabilities follow a multinomial logistic function of a continuous student ability parameter (cf., item response function in the nominal response model) and item response probabilities are defined as conditional probabilities of responses given a rule as in usual LCM. Although GSEREM has potential to better account for inconsistent use of cognitive rules, it suffers from unidentifiability of model parameters unless very strong constraints are placed. This certainly limits the

practical use of the model. Also, to what extent it can approximate the actual strategy switching is not fully known.

As seen above, there have been many attempts to relax the limitations of LCM for more realistic modeling of student response behaviors. The same lines of extensions of LCM-DI will be necessary to evaluate the usefulness of diagnostic items in a more appropriate manner and eventually extend the possibility of their use in practice.

References

Bart, W. M., Post, T., Behr, M. J., & Lesh, R. (1994). A diagnostic analysis of a proportional reasoning test item: An introduction to the properties of a semi-dense item. *Focus on Learning Problems in Mathematics*, *16*(3), 1–11.

Bart, W. M., & Williams-Morris, R. (1990). A refined digraph analysis of a proportional reasoning test. *Applied Measurement in Education*, *3*, 143–165.

Bartholomew, D. J., & Knott, M. (1999). *Latent variable models and factor analysis* (2nd ed.). London: Arnold.

Bejar, I. I. (1984). Educational diagnostic assessment. *Journal of Educational Measurement*, *21*, 175–189.

Bock, R. D. (1972). Estimating item parameters and latent proficiency when the responses are scored in two or more nominal categories. *Psychometrika*, *37*, 29–51.

Bolt, D. M., Cohen, A. S., & Wollack, J. A. (2001). A mixture item response model for multiple-choice data. *Journal of Educational and Behavioral Statistics*, *26*, 381–409.

Boom, J., Hoijtink, H., & Kunnen, S. (2001). Rules in the balance: Classes, strategies, or rules for the balance scale task? *Cognitive Development*, *16*, 717–735.

Boom, J., & ter Laak, J. (2007). Classes in the balance: Latent class analysis and the balance scale task. *Developmental Review*, *27*, 127–149.

Bozdogan, H. (1987). Model-selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, *52*, 345–370.

Briggs, D. C., Alonzo, A. C., Schwab, C., & Wilson, M. (2006). Diagnostic assessment with ordered multiple-choice items. *Educational Assessment*, *11*, 33–63.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, *2*, 155–192.

Ciofalo, J. F., & Wylie, E. C. (2006). Using diagnostic classroom assessment: One question at a time. *Teachers College Record*, January 10, 2006. Retrieved November 18, 2006, from http://www.tcrecord.org/PrintContent.asp?ContentID=12285

de la Torre, J. (2009). A cognitive diagnosis model for cognitively based multiple-choice options. *Applied Psychological Measurement*, *33*, 163–183.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*, 1–38.

Embretson, S. E. (1999). Cognitive psychology applied to testing. In F. T. Durso, R. S. Nickerson, R. W. Schvaneveldt, S. T. Dumais, D. S. Lindsay, & M. T. H. Chi (Eds.), *Handbook of applied cognition* (pp. 629–660). NY: Wiley.

Embretson, S. E., & Gorin, J. (2001). Improving construct validity with cognitive psychology principles. *Journal of Educational Measurement*, *38*, 343–368.

Falmagne, J. C. (1989). A latent trait theory via a stochastic learning theory for a knowledge space. *Psychometrika*, *54*, 283–303.

Formann, A. K. (1992). Linear logistic latent class analysis for polytomous data. *Journal of the American Statistical Association*, *87*, 476–486.

Fu, J. (2005). *A polytomous extension of the fusion model and its Bayesian parameter estimation.* Unpublished doctoral dissertation, University of Wisconsin, Madison.

Fu, J., & Li, Y. (2007, April). *Cognitively diagnostic psychometric models: An integrative review.* Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago, IL. Retrieved April 11, 2007, from http://www.ets.org/Media/Research/pdf/conf_AERA_NCME_07_Fu.pdf

Gelfand, A. E., Meng, X. L., & Stern, H. S. (1996). Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statistica Sinica*, *6*, 733-807.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–472.

Gierl, M. J., Leighton, J. P., & Hunka, S. M. (2007). Using the attribute hierarchy method to make diagnostic inferences. In J. P. Leighton & M. J. Gierl (Eds.), *Cognitive diagnostic assessment for education* (pp. 242–274). NY: Cambridge University Press.

Gitomer, D. H., & van Slyke, D. A. (1988). Error analysis and tutor design. *Machine-Mediated Learning*, *2*, 333–350.

Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, *61*, 215–231.

Graf, E. A., & Ohls, S. (2006, April). *Developing quantitative item models that support diagnostic assessment.* Paper presented at the annual meeting of the American Educational Research Association (AERA) and the National Council on Measurement in Education (NCME), San Francisco, CA.

Haertel, E. H. (1984). An application of latent class models to assessment data. *Applied*

Psychological Measurement, *8*, 333–346.

Haertel, E. H. (1989). Using restricted latent class models to map the skill structure of achievement items. *Journal of Educational Measurement*, *26*, 301–321.

Haladyna, T. (2004). *Developing and validating multiple-choice test items* (3rd ed.). Mahwah, NJ: Lawrence Erlbaum.

Hartz, S. M. (2002). *A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory and practicality.* Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign.

Henson, R., & Douglas, J. (2005). Test construction for cognitive diagnosis. *Applied Psychological Measurement*, *29*, 262–277.

Huff, K., & Goodman, D. P. (2007). The demand for cognitive diagnostic assessment. In J. P. Leighton & M. J. Gierl (Eds.), *Cognitive diagnostic assessment for education* (pp. 19–60). NY: Cambridge University Press.

Jansen, B. R. J., & van der Maas, H. L. J. (1997). A statistical test of the rule assessment methodology by latent class analysis. *Developmental Review*, *17*, 321–357.

Jansen, B. R. J., & van der Maas, H. L. J. (2002). The development of children's rule use on the balance scale task. *Journal of Experimental Child Psychology*, *81*, 383–416.

Junker, B. W. (1999). *Some statistical models and computational methods that may be useful for cognitively-relevant assessment.* Retrieved December 1, 2005, from http://www.stat.cmu.edu/~brian/nrc/cfa/document/final.pdf

Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, *25*, 258-272.

Kato, K. (2008a, July). *Improving efficiency of cognitive diagnosis by considering incorrect responses in multiple-choice items: A computerized adaptive testing perspective.* Poster presented at the International Meeting of the Psychometric Society, Durham, NJ.

Kato, K. (2008b, March). *Utilizing information from incorrect responses for cognitive diagnosis: Latent class modeling for multiple-choice items.* Paper presented at the annual meeting of the American Educational Research Association, New York, NY.

Klahr, D., & Siegler, R. S. (1978). The representation of children's knowledge. In H. W. Reese & L. P. Lipsitt (Eds.), *Advances in child development and behavior* (Vol. 12, pp. 61–116). NY: Academic Press.

Lazersfeld, P. F., & Henry, N. W. (1968). *Latent structure analysis.* New York: Houghton Mifflin.

Leighton, J. P., & Gierl, M. J. (Eds.). (2007a). *Cognitive diagnostic assessment for education.* NY: Cambridge University Press.

Leighton, J. P., & Gierl, M. J. (2007b). Defining and evaluating models of cognition used in educational measurement to make inferences about examinees' thinking processes. *Educational Measurement: Issues and Practice*, *26*(2), 3–16.

Leighton, J. P., & Gierl, M. J. (2007c). Why cognitive diagnostic assessment? In J. P. Leighton & M. J. Gierl (Eds.), *Cognitive diagnostic assessment for education* (pp. 3–18). NY: Cambridge University Press.

Leighton, J. P., Gierl, M. J., & Hunka, S. M. (2004). The attribute hierarchy model: An approach for integrating cognitive theory with assessment practice. *Journal of Educational Measurement*, *41*, 205–236.

Macready, G. B., & Dayton, C. M. (1977). The use of probabilistic models in the assessment of mastery. *Journal of Educational Statistics*, *2*, 99–120.

Macready, G. B., & Dayton, C. M. (1980). The nature and use of state mastery models. *Applied Psychological Measurement*, *4*, 493–516.

Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika*, *64*, 187–212.

Meng, X. L. (1994). Posterior predictive p-values. *The Annals of Statistics*, *22*, 1142–1160.

Messick, S. (1994). The interplay of evidence and consequences in the validation of performance assessments. *Eduational Researcher*, *23*(2), 13–23.

Minstrell, J. (2001). Facets of students' thinking: Designing to cross the gap from research to standards-based practice. In K. Crowley, C. D. Schunn, & T. Okada (Eds.), *Designing for science: Implications from professional, instructional, and everyday science* (pp. 415–443). Mahwah, NJ: Lawrence Erlbaum.

Minstrell, J., & Kraus, P. (2007, April). *Facets and facet clusters as a theoretical framework for designing assessment items.* Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago, IL.

Mislevy, R. J. (1993). Foundations of a new test theory. In N. Frederiksen, R. J. Mislevy, & I. I. Bejar (Eds.), *Test theory for a new generation of tests* (pp. 19–39). Hillsdale, NJ: Lawrence Erlbaum.

Mislevy, R. J. (1994). Evidence and inference in educational assessment. *Psychometrika*, *59*, 439–483.

Mislevy, R. J. (1995). Probability-based inference in cognitive diagnosis. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 43–71). Hillsdale, NJ: Lawrence Erlbaum.

Mislevy, R. J. (1996). Test theory reconceived. *Journal of Educational Measurement*, *33*, 379–416.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). *A brief introduction to evidence-cenetered design* (Research Report No. RR-03-16). Princeton, NJ: Educational Testing Service.

Mislevy, R. J., & Verhelst, N. (1990). Modeling item responses when different subjects employ different solution strategies. *Psychometrika*, *55*, 195–215.

Mislevy, R. J., & Wilson, M. (1996). Marginal maximum likelihood estimation for a psychometric model of discontinuous development. *Psychometrika*, *61*, 41–71.

Mooney, C. Z. (1997). *Monte Carlo simulation.* Thousand Oaks, CA: Sage.

National Research Council. (2001). *Knowing what students know: The science and design of educational assessment* (J. W. Pellegrino, N. Chudowsky, & R. Glaser, Eds.). Washington, DC: National Academy Press.

Nichols, P. D. (1994). A framework for developing cognitively diagnostic assessments. *Review of Educational Research*, *64*, 575–603.

Nichols, P. D., Chipman, S. F., & Brennan, R. L. (Eds.). (1995). *Cognitively diagnostic assessment.* Hillsdale, NJ: Lawrence Erlbaum.

No Child Left Behind Act of 2001, Pub. L. No. 107–110, 115 Stat. 1425 (2002).

Normandeau, S., Larivée, S., Roulin, J., & Longeot, F. (1989). The balance-scale dilemma: Either the subject or the experimenter muddles through. *Journal of Genetic Psychology*, *150*, 237–250.

Osterlind, S. J. (1998). *Constructing test items: Multiple-choice, constructed-response, and other formats* (2nd ed.). Boston, MA: Kluwer Academic Publishers.

Partnership for 21st Century Skills. (2005, June). *Assessment of 21st century skills: The current landscape.* Retrieved April 2, 2008, from http://www.21stcenturyskills.org /images/stories/otherdocs/Assessment_Landscape.pdf

Pirolli, P., & Wilson, M. (1998). A theory of the measurement of knowledge content, access, and learning. *Psychological Review*, *105*, 58–82.

Powell, J. C. (1968). The interpretation of wrong answers from a multiple choice test. *Educational and Psychological Measurement*, *28*, 403–412.

Powell, J. C. (1977). The developmental sequence of cognition as revealed by wrong answers. *Alberta Journal of Educational Research*, *23*, 43–51.

Powell, J. C., & Shklov, N. (1992). Obtaining information about learners' thinking strategies from wrong answers on multiple-choice tests. *Educational and Psychological Measurement*, *52*, 847–865.

R Development Core Team. (2009a). R: A language and environment for statistical computing [Computer software]. Available from http://www.R-project.org Vienna, Austria: R Foundation for Statistical Computing.

R Development Core Team. (2009b). Writing R extensions [Computer software]. Available from http://www.R-project.org Vienna, Austria: R Foundation for Statistical Computing.

Resnick, L. B., & Resnick, D. P. (1992). Assessing the thinking curriculum: New tools for educational reform. In B. R. Gifford & M. C. O'Connor (Eds.), *Changing assessments: Alternative views of aptitude, achievement, and instruction* (pp. 37–75). Norwell, MA: Kluwer.

Revuelta, J. (2004). Analysis of distractor difficulty in multiple-choice items. *Psychometrika*, *69*, 217–234.

Rijkes, C. P. M., & Kelderman, H. (2007). Latent-response Rasch model for strategy shifts in problem-solving processes. In M. von Davier & C. H. Carstensen (Eds.), *Multivariate and mixture distribution Rasch models* (pp. 311–328). NY: Springer.

Rost, J. (1990). Rasch models in latent classes: An integration of two approaches to item analysis. *Applied Psychological Measurement*, *14*, 271–282.

Rost, J. (1991). A logistic mixture distribution model for polychotomous item responses. *British Journal of Mathematical & Statistical Psychology*, *44*, 75–92.

Roussos, L. A., DiBello, L. V., Stout, W., Hartz, S. M., Henson, R. A., & Templin, J. L. (2007). The fusion model skills diagnosis system. In J. P. Leighton & M. J. Gierl (Eds.), *Cognitive diagnostic assessment for education* (pp. 275–318). NY: Cambridge University Press.

Rupp, A. A. (2007, April). *Unique characteristics of cognitive diagnosis models.* Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago, IL.

Rupp, A. A., & Templin, J. (2008). The effects of Q-matrix misspecification on parameter estimates and classification accuracy in the DINA model. *Educational and Psychological Measurement*, *68*, 78–96.

Sadler, P. M. (1998). Psychometric models of student conceptions in science: Reconciling qualitative studies and distractor-driven assessment instruments. *Journal of Research in Science Teaching*, *35*, 265–296.

Samejima, F. (1979). *A new family of models for the multiple choice item* (Research Report No. 79-4). Knoxville, TN: University of Tennessee, Department of Psychology.

Samejima, F. (1988). *Advancement of latent trait theory* (ONR Final Report). Knoxville,

TN: University of Tennessee, Department of Psychology.

Siegler, R. S. (1981). Developmental sequences within and between concepts. *Monographs of the Society for Research in Child Development*, *46*(2, Serial No. 189).

Siegler, R. S., & Chen, Z. (1998). Dvelopmental differences in rule learning: A microgenetic analysis. *Cognitive Psychology*, *36*, 273–310.

Snow, R. E., & Lohman, D. F. (1993). Cognitive psychology, new test design, and new test theory: An introduction. In N. Frederiksen, R. J. Mislevy, & I. I. Bejar (Eds.), *Test theory for a new generation of tests* (pp. 1–17). Hillsdale, NJ: Lawrence Erlbaum.

Stout, W. (2002). Psychometrics: From practice to theory and back: 15 years of nonparametric multidimensional IRT, DIF/test equity, and skills diagnostic assessment. *Psychometrika*, *67*, 485–518.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, *20*, 345–354.

Tatsuoka, K. K. (1985). A probabilistic model for diagnosing misconceptions by the pattern classification approach. *Journal of Educational Statistics*, *10*, 55–73.

Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnosis. In N. Frederiksen, R. Glaser, A. Lesgold, & M. G. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition* (pp. 453–488). Hillsdale, NJ: Lawrence Erlbaum.

Tatsuoka, K. K. (1991). *Boolean algebra applied to determination of universal set of knowledge states* (Research Report No. RR-91-44-ONR). Princeton, NJ: Educational Testing Service.

Thissen, D., & Steinberg, L. (1984). A response model for multiple choice items. *Psychometrika*, *49*, 501–509.

Thissen, D., Steinberg, L., & Fitzpatrick, A. R. (1989). Multiple-choice models: The distractors are also part of the item. *Journal of Educational Measurement*, *26*, 161–176.

van Maanen, L., Been, P. H., & Sijtsma, K. (1988). The linear logistic test model and heterogeneity of cognitive strategies in balance problems [Dutch]. *Tijdschrift voor Onderwijsresearch*, *13*, 301–310.

Weiss, D. J. (n.d.). *Research strategies in CAT.* Retrieved April 28, 2009, from University of Minnesota, Department of Psychology, CAT Central Web site: http://www.psych.umn.edu/psylabs/catcentral/

Westers, P., & Kelderman, H. (1991). Examining differential item functioning due to item difficulty and alternative attractiveness. *Psychometrika*, *57*, 107–118.

Williamson, D. M., Mislevy, R. J., & Bejar, I. I. (Eds.). (2006). *Automated scoring of complex tasks in computer-based testing.* Mahwah, NJ: Lawrence Erlbaum.

Wilson, M. (1989). Saltus: A psychometric model of discontinuity in cognitive development. *Psychological Bulletin*, *105*, 276–289.

Wylie, E. C., & Ciofalo, J. F. (2006, April). *One diagnostic item: Then what?* Paper presented at the annual meeting of the National Council on Measurement in Education, San Francisco, CA.

Wylie, E. C., & Wiliam, D. (2007, April). *Analyzing diagnostic items: What makes a student response interpretable?* Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago, IL.

Xu, X., Chang, H.-H., & Douglas, J. (2003, April). *A simulation study to compare CAT strategies for cognitive diagnosis.* Paper presented at the annual meeting of National Council on Measurement in Education, Montreal, Canada.

Yamamoto, K. (1989). *HYBRID model of IRT and latent class models* (Research Report No. RR-89-41). Princeton, NJ: Educational Testing Service.

Appendix A

Notes on the Computer Programs

This appendix describes how the computer programs used for adaptive testing simulations (Chapters 4 and 5) and parameter estimation in LCM-DI (Chapter 5) were validated. Data generated from a hypothetical population with known LCM parameters were used to evaluate the performance of these programs.

A.1 Generating Parameter Values

The underlying model was the unconstrained LCM for multiple choice responses with the number of items $J = 20$, the number of rules $L = 6$, the number of response options $K_j = 3$ for all items. Parameter values were randomly generated using the procedure described below, and the same set of parameter values were used to generate data to evaluate both adaptive testing and parameter estimation programs.

Under the unconstrained LCM, the matrix of logits of response probabilities for item j

takes the following form:

$$\boldsymbol{\eta}_j = \begin{array}{c} \\ \text{Response 1} \\ \text{Response 2} \\ \text{Response 3} \end{array} \begin{array}{c} \begin{array}{cccc} \text{Rule 1} & \text{Rule 2} & \cdots & \text{Rule 6} \end{array} \\ \begin{bmatrix} \alpha_{j11} & \alpha_{j12} & \cdots & \alpha_{j16} \\ \alpha_{j21} & \alpha_{j22} & \cdots & \alpha_{j26} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \end{array}, \qquad j = 1, \ldots, 20, \tag{A.1}$$

where the logits for the last response under each rule was set to 0, because there are only 2 degrees of freedom for the three-category multinomial probabilities. This resulted in 12 free logit parameters for each item. The marginal rule probabilities were also initially expressed as logits to be compatible with how they are treated in the parameter estimation program. These logits were denoted by $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_5, 0)$, where the last logit was set to 0 to avoid redundancy. The total number of free parameters was thus $12 \times 20 + 5 = 245$.

The logit parameters were randomly generated from the following distributions:

$$\alpha_{jkl} \overset{i.i.d.}{\sim} N(0, 3), \qquad j = 1, ..., 20,\ k = 1, 2, \tag{A.2}$$

$$\beta_l \overset{i.i.d.}{\sim} N(0, 0.5), \qquad l = 1, ..., 5. \tag{A.3}$$

The logit parameters were then transformed into probabilities $(\boldsymbol{\pi}, \boldsymbol{\omega})$, from which item responses were readily generated. The logits generated by the above configuration produced a wide range of item response and marginal rule probability values and thus were considered appropriate for the testing purpose. The same set of parameter values was used for verification of the adaptive testing simulation programs and the parameter estimation program. Generation of the logits, their transformations into probabilities, and subsequent item response data generation were all conducted in R.

A.2 Validating the Programs for Adaptive Testing Simulations

Programs that implemented GDI and ShE adaptive testing simulations were written in R. In order to evaluate their performance, responses of 12 hypothetical examinees were simulated and subjected to adaptive testing.

Two examinees were assigned to each rule, and their responses to the 20 items were generated by using the corresponding item response probabilities obtained in the previous section. These item responses were subjected to adaptive testing implemented by the R programs under the following setting:

Prior rule distribution: Uniform distribution

Starting rule: Item 1 is administered first for all examinees.

Item selection: GDI or ShE

Stopping rule: Testing stops if a posterior probability ($\omega_{il}^{(s)}$) for any rule exceeds .95.

The same adaptive testing procedures were also performed in Excel. Item response probabilities and examinees' responses were read in to Excel spreadsheets, and from these values all numbers necessary for adaptive testing (e.g., posterior rule probabilities, GDI values, and ShE values) were calculated for each examinee. All calculations were made by Excel built-in functions embedded in cells (i.e., functions for sum, product, and the logarithm as well as basic arithmetic operations).

These two parallel adaptive testing implementations (R and Excel) were compared for each examinee in terms of the following points:

- GDI values (GDI_j) for each item and whether an item with the largest GDI was selected in each iteration (for GDI item selection)

- ShE values ($S_j(\boldsymbol{\omega}_i^{(s+1)})$) and whether an item with the smallest ShE was selected in each iteration (for ShE item selection)
- Posterior rule probabilities ($\boldsymbol{\omega}_i^{(s)}$) and the rule estimate ($\hat{\phi}_i^{(s)}$) in each iteration
- Items selected and their order ($\{j^{(1)}, \ldots, j^{(s)}\}$)

Both R and Excel implementations produced exactly the same results for both GDI and ShE, and thus validity of the R programs was confirmed.

A.3 Validating the Program for LCM Parameter Estimation

The program that implemented the EM algorithm for parameter estimation in LCM-DI was written in C so that it can be called from the R console. The C program was designed to implement parameter estimation not only for LCM-DI but also for general linear logistic LCM (Formann, 1992), and thus it should be able to estimate the parameter values in the unconstrained LCM.

Performance of the C program was tested by a Monte Carlo simulation (e.g., Mooney, 1997). Responses of 1,000 hypothetical examinees were generated. For each examinee i, (a) a "true" rule (ϕ_i) was drawn from the marginal rule probability distribution $\boldsymbol{\omega}$, and (b) responses to the 20 items were generated based on the item response probabilities $\boldsymbol{\pi}$ for the examinee's true rule. The set of probability values obtained in Section A.1 was used to generate multiple choice response data. Then, the data matrix was sent to the C program, which estimated the maximum likelihood estimates (MLEs) of the logit parameters. This process was repeated 100 times to obtain 100 sets of MLEs: $(\hat{\boldsymbol{\alpha}}_t, \hat{\boldsymbol{\beta}}_t)$, $t = 1, \ldots, 100$. These estimates were also transformed to obtain MLEs of the item response and marginal rule probabilities: $(\hat{\boldsymbol{\pi}}_t, \hat{\boldsymbol{\omega}}_t)$, $t = 1, \ldots, 100$. These estimates were compared to the true parameter values.

One of the properties to evaluate is the bias of MLEs produced by the C program. For an arbitrary parameter θ and its estimator $\hat{\theta}$ (in the current case, θ can be any of α, β, π, and ω parameters), the bias is defined as $E(\hat{\theta}) - \theta$. The bias cannot be 0 in the current setting because of the limited size of each Monte Carlo sample (i.e., the sample size of 1,000 examinees for each sample may be too short for the asymptotic unbiasedness of MLEs to apply), the limited number of Monte Carlo iterations, and most importantly the computational precision of the C program itself. Nonetheless, small biases, if found, would provide partial support for the program's validity in the sense that the program produces estimates close to their true values on average.

"Smallness" of the bias needs to be evaluated on an explicit scale, because the amount of bias can depend on various factors as mentioned above. In general, the following equation holds for parameter θ and its estimator $\hat{\theta}$:

$$E(\hat{\theta} - \theta)^2 = E[\hat{\theta} - E(\hat{\theta})]^2 + [E(\hat{\theta}) - \theta]^2, \tag{A.4}$$

where all expectations are taken with respect to the sampling distribution of $\hat{\theta}$. The left-hand side of Equation A.4 is the mean squared error (MSE) of $\hat{\theta}$, and its square root is called the root mean squared error (RMSE), which indicates the average variation of $\hat{\theta}$ around the true θ. The first term of the right-hand side of Equation A.4 is the variance of $\hat{\theta}$, and the second term is the squared bias. If the proportion of the squared bias to the MSE (bias-MSE ratio) is small, it would be a good indication that the bias is small.

Each term in Equation A.4 can be estimated from T Monte Carlo samples as follows:

$$\frac{1}{T}\sum_{t=1}^{T}(\hat{\theta}_t - \theta)^2 = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{\theta}_t - \frac{1}{T}\sum_{t=1}^{T}\hat{\theta}_t\right)^2 + \left(\frac{1}{T}\sum_{t=1}^{T}\hat{\theta}_t - \theta\right)^2, \tag{A.5}$$

where $\hat{\theta}_t$ is the estimate of θ obtained at the tth Monte Carlo iteration. In the current simulation, the estimate of bias, RMSE, and the bias-MSE ratio were computed for each

parameter with $T = 100$.

Table A.1: Summary of Bias, Root Mean Squared Error, and Bias-MSE Ratio

Parameter	25%	50%	Mean	75%
$\boldsymbol{\alpha}$ $(n = 240)$				
Bias	−0.4313	−0.0654	−0.0910	0.3339
RMSE	1.6438	2.2165	2.6291	3.0588
Bias-MSE Ratio	.0101	.0347	.0539	.0759
$\boldsymbol{\beta}$ $(n = 5)$				
Bias	0.0680	0.0932	0.0862	0.1117
RMSE	0.9272	0.9788	1.0069	1.0159
Bias-MSE Ratio	.0048	.0084	.0115	.0088
$\boldsymbol{\pi}$ $(n = 240)$				
Bias	−0.0464	0.0128	0.0008	0.0517
RMSE	0.1358	0.1817	0.2044	0.2387
Bias-MSE Ratio	.0440	.0849	.0865	.1284
$\boldsymbol{\omega}$ $(n = 5)$				
Bias	−0.0019	0.0011	0.0000	.0058
RMSE	0.0448	0.0532	0.0586	0.0672
Bias-MSE Ratio	.0023	.0150	.0236	.0368

The top half of Table A.1 shows the results on the original logit scale, and the bottom half shows the results on the probability scale. For $\boldsymbol{\alpha}$, RMSEs were quite large (mean 2.6291), showing that estimates from individual Monte Carlo samples had large variability around the true values. However, the mean bias and the mean ratio were almost 0 (−0.0910 and .0539, respectively), indicating that the bias component was about 5.39% to the MSE and the estimates were fairly unbiased. For $\boldsymbol{\beta}$, the bias component was even smaller (mean .0115) than for $\boldsymbol{\alpha}$, indicating that $\boldsymbol{\beta}$ parameters were well estimated. When these estimates were evaluated on the probability scale, the bias-MSE ratios tended to become larger (the means for $\boldsymbol{\pi}$ and $\boldsymbol{\omega}$ were .0865 and .0236, respectively). However, the amount of average bias was almost 0 for both classes of parameters (0.0008 and 0.0000). These results indicate that the program works properly as intended, producing estimates with biases within an acceptable range.

Appendix B

Computer Programs

Programs that were used in Studies 1 and 2 are shown in this appendix. Four programs were used in these studies. Two of them, `s1.r` and `s2.r`, are the main programs that conducted analyses in Studies 1 and 2, respectively. The other two programs, `sub.r` and `lllca.c`, contain subroutines. Subroutines in `sub.r` are directly called from the two main programs, while those in `lllca.c` do technical computation (i.e., parameter estimation in LCM-DI) and are only called from `sub.r`. The relationships among these programs are shown in Figure B.1.

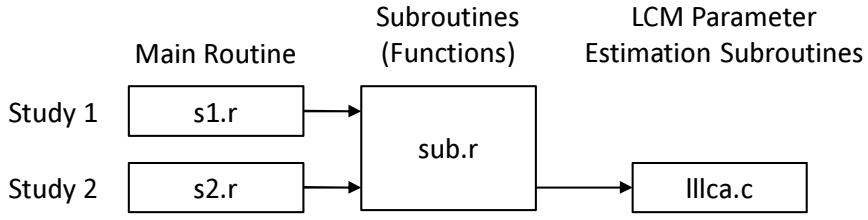

Figure B.1: Relationship Among the Computer Programs

All programs except for `lllca.c` were written in R. `lllca.c` is a C program and must be compiled with an appropriate interpreter before used. The process of compilation depends on the operating system. For more details and general procedures about how to call C

programs from R, see R Development Core Team (2009b). All program files (including the compiled C program) and data files must be placed in the same folder (the programs below assume that all files are located in `h:/analysis/`).

B.1 s1.r

```
# ------------------------------------------------------------------------
# s1.r
# R code for adaptive testing simulation for Study 1

# set paths to the files
pathr = pathd = "h:/analysis"
source(file.path(pathr,"sub.r")) # loading R subroutines

# ------------------------------------------------------------------------
# CREATING ITEM POOLS

# generate response-by-rule matrices
nk = 4   # of response options
nj = 100 # of items per item feature combination
nl = c(5,7,9)       # of rules = L
ri = 2:nk           # response interpretability = RI
id = c(.75,.85,.95) # item discrimination = ID
IPrr = IPpp = IPri = IPnrp = IPid = IPr0 = IPp0 = list(length(nl))

set.seed(3854) # seed for RNG
for(l in 1:length(nl)){

njt  = sum(nl[l]-ri)*length(id)*nj     # of all items in IPx (x = nl[l])
rr   = array(0,dim=c(njt,nk,nl[l]))    # response-by-rule matrices
pp   = array(1/nk,dim=c(njt,nk,nl[l])) # item response probability
iri  = rep(rep(
           unlist(sapply(ri,function(x) rep(x,nl[l]-x))),each=nj),
           length(id)) # RI index
inrp = rep(rep(nl[l]-sequence(nl[l]-ri),each=nj),length(id)) # NRP index
iid  = rep(1:length(id),each=length(sequence(nl[l]-ri))*nj)  # ID index

for(j in 1:sum(iid==1)){
  rr[j,1,2] = 1 # correct rule -> correct response
  ruleidx   = rep(0,nl[l]); ruleidx[c(1,2)] = 1
  respidx   = rep(0,nk); respidx[1] = 1
  rules     = sample(3:nl[l],iri[j]-1); ruleidx[rules] = 1
  resps     = sample(2:nk,iri[j]-1); respidx[resps] = 1
  for(k in 1:(iri[j]-1)) rr[j,resps[k],rules[k]] = 1

  if(inrp[j]-iri[j]>0){
    if(sum(ruleidx==0)==1)
      rules = rep((1:nl[l])[ruleidx==0],inrp[j]-iri[j])
    else if(sum(ruleidx==0)>1)
      rules = sample((1:nl[l])[ruleidx==0],inrp[j]-iri[j])
    resps = sample((1:nk)[respidx==1],inrp[j]-iri[j],replace=TRUE)
```

```
    for(k in 1:(inrp[j]-iri[j])) rr[j,resps[k],rules[k]] = 1
  }
}

if(length(id)>1)
  for(j in 2:length(id))
    rr[(1:sum(iid==1))+(j-1)*sum(iid==1),,] = rr[1:sum(iid==1),,]

# generate item response probabilities
for(j in 1:njt){
  pp[j,,][rr[j,,]==1] = runif(sum(rr[j,,]),id[iid[j]]-.05,id[iid[j]]+.05)
  for(k in 2:nl[l])
    if(sum(rr[j,,k])>0)
      pp[j,rr[j,,k]==0,k] = (1-pp[j,rr[j,,k]==1,k])/(nk-1)
}

IPrr[[l]] = rr; IPpp[[l]] = pp
IPri[[l]] = iri; IPnrp[[l]] = inrp; IPid[[l]] = iid

# collapse (dichotomize) RR matrices and response probabilities
# 1 = incorrect, 2 = correct
r0 = p0 = array(0,dim=c(njt,2,nl[l]))
r0[,2,] = rr[,1,]; r0[,1,] = apply(rr[,-1,],c(1,3),sum)
p0[,2,] = pp[,1,]; p0[,1,] = 1-p0[,2,]
IPr0[[l]] = r0; IPp0[[l]] = p0

} #l

# IPrr response-by-rule matrices (MC)
# IPpp item response probabilities (MC)
# IPr0 response-by-rule matrices (dichotomous)
# IPp0 item response probabilities (dichotomous)
# IPri = RI; IPnrp = NRP; IPid = ID

# ---------------------------------------------------------------------
# ADAPTIVE TESTING SIMULATIONS

s1_main = function(pp,p0,pcmin=.95,ni.l=100){
# this function runs adaptive testing simulations for a given item pool
# pp   item response probabilities (MC)
# p0   item response probabilities (dichotomous)
# ni.l # of examinees per rule

nl  = dim(pp)[3] # of rules
nk  = dim(pp)[2] # of response options
nj  = dim(pp)[1] # of items
nkp = rep(nk,nj); nkd = rep(2,nj)
ni.sim = nl*ni.l                        # total sample size
xsp = xsd = array(0,dim=c(ni.sim,nj)) # data matrices
c.tr = rep(1:nl,each=ni.l)              # true class membership

# generate MC responses
for(i in 1:ni.sim)
  for(j in 1:nj)
    xsp[i,j] = (1:nk)%*%rmultinom(1,1,pp[j,,c.tr[i]])
```

```
# dichotomize the MC responses
xsd[xsp==1] = 1; xsd = xsd + 1 # 1 = incorrect, 2 = correct

# GDI item selection
first.item = sample(1:nj,ni.sim,replace=TRUE) # randomly select 1st item
cdip = cdi(pp,nkp,10); gdip = gdi(cdip)       # compute GDI (MC)
cdid = cdi(p0,nkd,10); gdid = gdi(cdid)       # compute GDI (dichotomous)
CATGDIp = catclsGDIsim(xsp,gdip,pp,pcmin,first=first.item)
CATGDId = catclsGDIsim(xsd,gdid,p0,pcmin,first=first.item)

# ShE item selection
CATShEp = catclsSEsim(xsp,pp,pcmin,first=first.item)
CATShEd = catclsSEsim(xsd,p0,pcmin,first=first.item)

# RND item selection
CATRNDp = catclsRNDsim(xsp,pp,pcmin,first=first.item)
CATRNDd = catclsRNDsim(xsd,p0,pcmin,first=first.item)

# classification by all items
c.allp = postcls(xsp,pp)$cls
c.alld = postcls(xsd,p0)$cls

# summarize results
summary.CAT = function(CATp,CATd){
c(quantile(CATp$njadm),quantile(CATd$njadm),
  quantile(CATp$njadm/CATd$njadm),
  mean(CATp$njadm),mean(CATd$njadm),mean(CATp$njadm/CATd$njadm),
  sd(CATp$njadm),sd(CATd$njadm),sd(CATp$njadm/CATd$njadm),
  sum(CATp$cls==c.tr)/ni.sim,sum(CATd$cls==c.tr)/ni.sim,
  sum(CATp$cls==c.allp)/ni.sim,sum(CATd$cls==c.alld)/ni.sim,
  sum(c.allp==c.tr)/ni.sim,sum(c.alld==c.tr)/ni.sim,
  sum(1-CATp$clsd)/ni.sim,sum(1-CATd$clsd)/ni.sim,
  mean(apply(CATp$iidx>0,2,sum)/sum(CATp$iidx>0)),
  sd(apply(CATp$iidx>0,2,sum)/sum(CATp$iidx>0)),
  mean(apply(CATd$iidx>0,2,sum)/sum(CATd$iidx>0)),
  sd(apply(CATd$iidx>0,2,sum)/sum(CATd$iidx>0)))
}

return(c(
  ni.sim,nj,
  summary.CAT(CATGDIp,CATGDId),
  summary.CAT(CATShEp,CATShEd),
  summary.CAT(CATRNDp,CATRNDd)))
} # s1_main

# setting labels for output
lab.s1  = c(paste("P",seq(0,100,25),sep=""),
            paste("D",seq(0,100,25),sep=""),
            paste("R",seq(0,100,25),sep=""),
            "PMean","DMean","RMean",
            "PSD","DSD","RSD","PCR","DCR","PAR","DAR","APCR","ADCR",
            "PNR","DNR","ExPMean","ExPSD","ExDMean","ExDSD")
lab.out = c("L","RI","NRP","ID","NI","NJ",paste(lab.s1,"_GDI",sep=""),
            paste(lab.s1,"_ShE",sep=""),paste(lab.s1,"_RND",sep=""))
```

```
# count # of all feature combinations
ns = 0
for(l in 1:length(nl))
  for(j in 1:length(ri))
    for(k in ri[j]:(nl[l]-1))
      for(kk in 1:length(id)) ns = ns+1

out = array(NA,dim=c(ns,length(lab.out))) # output variable
colnames(out) = lab.out
idx = list(ns)

# index items in a selected item pool
s = 0
for(l in 1:length(nl)){
  njt = length(IPri[[l]])
    for(j in 1:length(ri)){
      for(k in ri[j]:(nl[l]-1)){
        for(kk in 1:length(id)){
          s = s+1
          idx[[s]] = (1:njt)[IPri[[l]]==ri[j]&IPnrp[[l]]==k&IPid[[l]]==kk]
}}}}

# run adaptive testing for all feature combinations
set.seed(7589713) # seed for RNG
pcmin = .90       # stopping criterion
s = 0
for(l in 1:length(nl)){ # L
  njt = length(IPri[[l]])
    for(j in 1:length(ri)){ # RI
      for(k in ri[j]:(nl[l]-1)){ # NRP
        for(kk in 1:length(id)){ # ID
          s = s+1
          out[s,] = c(nl[l],ri[j],k,id[kk],
                      s1_main(IPpp[[l]][idx[[s]],,],
                              IPp0[[l]][idx[[s]],,],pcmin))
}}}}

# results are output to "s1_out.csv"
write.csv(out,file=file.path(pathr,"s1_out.csv"),row.names=FALSE)
```

B.2 s2.r

```
# ------------------------------------------------------------------------
# s2.r
# R code for the Balance Scale data for Study 2
# ------------------------------------------------------------------------

# set paths to the files
pathr = pathd = "h:/analysis"
source(file.path(pathr,"sub.r")) # R functions

# reading data files
d0 = read.fwf(file.path(pathd,"bb.dat"),
              width=rep(1,20),
```

```
              col.names=paste("x",1:20,sep=""))
# d0 = response data file
# plain text containing polytomous responses
# (1 = left, 2 = balance, 3 = right)
# each line represents one examinee's responses
# there is no separator between data values
# the first line should NOT contain column labels
# Example:
# Line 1: 32131323123123123112
# Line 2: 11323132312323232111
# Line 3: 31323123322113131323
#         ...

rbr0 = read.csv(file.path(pathd,"rr.csv"),header=TRUE)
# rbr0 = response-by-rule matrices
# plain text containing RR matrices information
# data are separated by a comma
# the first line should contain column labels
# Column 1  = item number
# Column 2  = response number
# Column 3 through 3+L = response indicators
# Example:
# Line 1: Item,Response,Rule1,Rule2,Rule3,Rule4
# Line 2: 1,1,0,0,0,0
# Line 3: 1,2,0,0,1,0
# Line 4: 1,3,0,1,0,1
# Line 5: 2,1,0,0,0,1
# Line 6: 2,2,0,1,0,0
# Line 7: 2,3,0,1,0,0
#         ...
#
# the remaining part assume that rbr0 contains the following
# rules in this order:
#   1 = Complete Random Guessing (R0)
#   2 = Rule 1 (R1)
#   3 = Rule 2 (R2)
#   4 = Rule 3 (R3)
#   5 = Torque (R4)
#   6 = Addition (Add)
#   7 = Qualitative Proportion (QP)
#   8 = Smallest Distance Down (SDD)
#   9 = Distance Dominant (DD)
#  10 = Buggy (BG)

x   = as.matrix(d0) # response data
ni  = nrow(x)       # of subjects
nj  = ncol(x)       # of items
nka = 3
nk  = rep(nka,nj)   # of response options (vector)

# ---------------------------------------------------------------------------
# LCM MODEL FITTING: MULTIPLE CHOICE RESPONSES
# ---------------------------------------------------------------------------
# Rule 3 INCLUDED

fp = fr = list()
ridx0 = 1:5  # first 5 rules (FIXED)
ridx1 = 6:10 # selective rules
```

```
ii = 1
fr[[ii]] = ridx0; ii = ii + 1
# fr contains all model fitting results in the list format

for(l in 1:(10-length(ridx0))){
  rridx = combinations(length(ridx1),l,ridx1)
  for(j in 1:nrow(rridx)){
    fr[[ii]] = c(ridx0,rridx[j,])
    ii = ii + 1;
}}

fnl = rep(0,length(fr)); frlab = list()
for(ii in 1:length(fr)){
  fnl[ii] = length(fr[[ii]])
  frlab[[ii]] = names(rbr0)[-(1:2)][fr[[ii]]]
}

chkresp = rep(NA,length(fr))
for(ii in 1:length(fr)){
  rbrx = as.matrix(rbr0)[,c(1,2,fr[[ii]]+2)]
# repsonse pattern check
  eresp = array(NA,dim=c(nj,fnl[ii]))
  for(j in 1:nj) eresp[j,] = (1:3)%*%rbrx[rbrx[,1]==j,3:(fnl[ii]+2)]
  chkresp[ii] = (nrow(unique(t(eresp)))==fnl[ii])
  if(chkresp[ii]){
    zidx = expand.grid(1:3,1:fnl[ii],1:nj)[,3:1]
# create a covariate matrix for conditional probabilities
    h0 = paste("a",rep(1:nj,each=fnl[ii]),"_",
               rep(1:fnl[ii],times=nj),sep="")
    z0 = matrix(0,nrow=nrow(zidx),ncol=fnl[ii]*nj)
    colnames(z0) = h0
    for(j in 1:nj)
      for(l in 1:fnl[ii])
        for(k in 1:3)
          if(rbrx[rbrx[,1]==j & rbrx[,2]==k,l+2]==1)
            z0[zidx[,1]==j&zidx[,2]==l&zidx[,3]==k,(j-1)*fnl[ii]+l] = 1
    z0 = z0[,!apply(z0,2,function(x) all(x==0))]
    fp[[ii]] = lllcaNP(x,nk,fnl[ii],y=NULL,yidx=NULL,
                       z=z0,zidx=zidx,rstart=FALSE)
  } else{ fp[[ii]] = NULL }
} #ii

# ---------------------------------------------------------------------------
# Rule 3 NOT included

fp2 = fr2 = list()
ridx0 = (1:5)[-4] # first 5 rules minus Rule 3 (FIXED)
ridx1 = 6:10      # selective rules

ii = 1
fr2[[ii]] = ridx0; ii = ii + 1
# fr2 contains all model fitting results in the list format

for(l in 1:(9-length(ridx0))){
  rridx = combinations(length(ridx1),l,ridx1)
  for(j in 1:nrow(rridx)){
```

```
    fr2[[ii]] = c(ridx0,rridx[j,])
    ii = ii + 1;
}}

fnl2 = rep(0,length(fr2)); frlab2 = list()
for(ii in 1:length(fr2)){
  fnl2[ii] = length(fr2[[ii]])
  frlab2[[ii]] = names(rbr0)[-(1:2)][fr2[[ii]]]
}

chkresp2 = rep(NA,length(fr2))
for(ii in 1:length(fr2)){
  rbrx = as.matrix(rbr0)[,c(1,2,fr2[[ii]]+2)]
# repsonse pattern check
  eresp = array(NA,dim=c(nj,fnl2[ii]))
  for(j in 1:nj) eresp[j,] = (1:3)%*%rbrx[rbrx[,1]==j,3:(fnl2[ii]+2)]
  chkresp2[ii] = (nrow(unique(t(eresp)))==fnl2[ii])
  if(chkresp2[ii]){
    zidx = expand.grid(1:3,1:fnl2[ii],1:nj)[,3:1]
# create a covariate matrix for conditional probabilities
    h0 = paste("a",rep(1:nj,each=fnl2[ii]),"_",
               rep(1:fnl2[ii],times=nj),sep="")
    z0 = matrix(0,nrow=nrow(zidx),ncol=fnl2[ii]*nj)
    colnames(z0) = h0
    for(j in 1:nj)
      for(l in 1:fnl2[ii])
        for(k in 1:3)
          if(rbrx[rbrx[,1]==j & rbrx[,2]==k,l+2]==1)
            z0[zidx[,1]==j&zidx[,2]==l&zidx[,3]==k,(j-1)*fnl2[ii]+l] = 1
    z0 = z0[,!apply(z0,2,function(x) all(x==0))]
    fp2[[ii]] = lllcaNP(x,nk,fnl2[ii],y=NULL,yidx=NULL,
                        z=z0,zidx=zidx,rstart=FALSE)
  } else{ fp2[[ii]] = NULL }
} #ii

# ------------------------------------------------------------------------
# LCM MODEL FITTING: DICHOTOMIZED RESPONSES
# ------------------------------------------------------------------------
# Rule 3 INCLUDED

# dichotomize the MC data
anskey = rbr0[rbr0[,'R4']==1,'resp']
x1 = x
x1[t(t(x)!=anskey)] = 0; x1[t(t(x)==anskey)] = 1
x1 = x1 + 1 # 1 = incorrect, 2 = correct
nk1 = rep(2,nj)

fd = list(); ffnl = unique(fnl)
# fd contains all model fitting results in the list format

ll = 1
for(nl in ffnl){
  zidx1 = expand.grid(1:2,1:nl,1:nj)[,3:1]
  rbrx1 = matrix(0,nrow=sum(nk1),ncol=nl+2)
  rbrx1[,1] = rep(1:nj,nk1); rbrx1[,2] = sequence(nk1)
  rbrx1[rbrx1[,2]==2,3:(nl+2)] = 1
```

```
  z1 = matrix(0,nrow=nrow(zidx1),ncol=nj*nl)
  s = t = 1
  for(j in 1:nj)
    for(l in 1:nl)
      for(k in 1:2){
        if(k==2){ z1[s,t] = 1; t = t+1 }
        s = s+1
      }

# fixed-value parameter constraints
# Class 1 = R0
# Class 2 = R3
# so for the first two classes...
  alphaf = (apply(z1[zidx1[,2]==1,],2,sum) +
            apply(z1[zidx1[,2]==2&(zidx1[,1]%in%6:20),],2,sum))
  alphav = alphaf * log(1/(3-1)) # fixed parameter values
  fd[[ll]] = lllcaNPC(x1,2,nl,y=NULL,yidx=NULL,z=z1,zidx=zidx1,
                      alphaf=alphaf,alpha=alphav)
  ll = ll+1
} #nl

# ----------------------------------------------------------------------------
# Rule 3 NOT included

fd2 = list(); ffnl2 = unique(fnl2)
# fd2 contains all model fitting results in the list format

ll = 1
for(nl in ffnl2){
  zidx1 = expand.grid(1:2,1:nl,1:nj)[,3:1]
  rbrx1 = matrix(0,nrow=sum(nk1),ncol=nl+2)
  rbrx1[,1] = rep(1:nj,nk1); rbrx1[,2] = sequence(nk1)
  rbrx1[rbrx1[,2]==2,3:(nl+2)] = 1

  z1 = matrix(0,nrow=nrow(zidx1),ncol=nj*nl)
  s = t = 1
  for(j in 1:nj)
    for(l in 1:nl)
      for(k in 1:2){
        if(k==2){ z1[s,t] = 1; t = t+1 }
        s = s+1
      }

# fixed-value parameter constraints
# Class 1 = R0
  alphaf = apply(z1[zidx1[,2]==1,],2,sum) # fixed parameter indicator
  alphav = alphaf * log(1/(3-1)) # fixed parameter values
  fd2[[ll]] = lllcaNPC(x1,2,nl,y=NULL,yidx=NULL,z=z1,zidx=zidx1,
                       alphaf=alphaf,alpha=alphav)
  ll = ll+1
}

# parametric bootstrap
fdpb2 = list(); ll = 1
for(nl in ffnl2){
  set.seed(365+nl)
  zidx1 = expand.grid(1:2,1:nl,1:nj)[,3:1]
  rbrx1 = matrix(0,nrow=sum(nk1),ncol=nl+2)
```

```
  rbrx1[,1] = rep(1:nj,nk1); rbrx1[,2] = sequence(nk1)
  rbrx1[rbrx1[,2]==2,3:(nl+2)] = 1

  z1 = matrix(0,nrow=nrow(zidx1),ncol=nj*nl)
  s = t = 1
  for(j in 1:nj)
    for(l in 1:nl)
      for(k in 1:2){
        if(k==2){ z1[s,t] = 1; t = t+1 }
        s = s+1
      }

  alphaf = apply(z1[zidx1[,2]==1,],2,sum)
  alphav = alphaf * log(1/(3-1))
  fdpb2[[ll]] = pblllcaNPC(fd2[[ll]],x1,nk1,nl,
                           fd2[[ll]]$alpha,alphaf,fd2[[l]]$beta,nboot=100)
  ll = ll+1
} #nl

# proportion-correct score
xp = apply(x1,1,function(x) sum(x==2)/nj)
mean(xp); sd(xp)

# ------------------------------------------------------------------------
# COMPARE MC - DICHOTOMOUS RESULTS
# ------------------------------------------------------------------------
# Rule 3 INCLUDED

fp0 = fp; fd0 = list()
fm = fg = array(NA,dim=c(length(fr),2))
colnames(fm) = c('ppi','omega'); colnames(fg) = c('BIC_P','BIC_D')
frlab2 = array(character(),dim=c(length(fr),10))

# matching classes between MC and dichotomous models
for(ii in 1:length(fr)){
  if(chkresp[ii]){
    nl = fnl[ii]
    fd0[[ii]] = fd[[which(ffnl==nl)]]
    ppid = array(0,dim=c(nj,2,nl))
    for(j in 1:nj){
      ppid[j,2,] = fp0[[ii]]$ppi[j,anskey[j],]
      ppid[j,1,] = 1-ppid[j,2,]
    }

    idx = 1:nl
    for(l in (1:nl)[-c(1,4)]){
      q = rep(nj*2,nl)
      for(ll in (1:nl)[-c(1,2)])
        q[ll] = sum(abs(ppid[,,l]-fd0[[ii]]$ppi[,,ll]))
      idx[l] = which.min(q)
    }
    idx[c(1,4)] = c(1,2)

    if(!all(sort(idx)==(1:nl))){ # in case of ties
      idxx = permutations(nl,nl); idx = idxx[1,]
      q = sum(abs(ppid-fd0[[ii]]$ppi))
      for(k in 2:nrow(idxx)){
```

```
        idx0 = as.vector(idxx[k,])
        q0 = sum(abs(ppid[,,idx0]-fd0[[ii]]$ppi[,,]))
        if(q0<q){ q = q0; idx = idx0 }
    }}

    tmp = fd0[[ii]]$ppi; tmpp = fd0[[ii]]$omega
    for(l in 1:nl){
      fd0[[ii]]$ppi[,,l] = tmp[,,idx[l]]
      fd0[[ii]]$omega[l] = tmpp[idx[l]]
    }

    fm[ii,] = c(mean(abs(ppid-fd0[[ii]]$ppi)),
                mean(abs(fp[[ii]]$omega-fd0[[ii]]$omega)))
  }
} #ii

for(ii in 1:length(fr)){
  if(chkresp[ii]){
    nl = fnl[ii]
    fg[ii,] = c(fp[[ii]]$bic,fd[[which(ffnl==nl)]]$bic)
  }
  frlab2[ii,1:length(frlab[[ii]])] = frlab[[ii]]
}

# output the model comparison result to a file
write.csv(data.frame(fnl,frlab2,fg,fm),
          file.path(pathr,"MDcomparison.csv"))

# ------------------------------------------------------------------------
# Rule 3 NOT included

fp02 = fp2; fd02 = list()
fm2 = fg2 = array(NA,dim=c(length(fr2),2))
colnames(fm2) = c('ppi','omega'); colnames(fg2) = c('BIC-P','BIC-D')
frlab22 = array(character(),dim=c(length(fr2),10))

# matching classes between MC and dichotomous models
for(ii in 1:length(fr2)){
  if(chkresp2[ii]){
    nl = fnl2[ii]
    fd02[[ii]] = fd2[[which(ffnl2==nl)]]
    ppid = array(0,dim=c(nj,2,nl))
    for(j in 1:nj){ # dichotomize MC item response probabilities
      ppid[j,2,] = fp02[[ii]]$ppi[j,anskey[j],]
      ppid[j,1,] = 1-ppid[j,2,]
    }

    idx = 1:nl
    for(l in 2:nl){
      q = rep(nj*2,nl)
      for(ll in (1:nl)[-c(1,2)])
        q[ll] = sum(abs(ppid[,,l]-fd0[[ii]]$ppi[,,ll]))
      idx[l] = which.min(q)
    }
    idx[1] = 1
```

```
    if(!all(sort(idx)==(1:nl))){ # in case of ties
      idxx = permutations(nl,nl); idx = idxx[1,]
      q = sum(abs(ppid-fd02[[ii]]$ppi))
      for(k in 2:nrow(idxx)){
        idx0 = as.vector(idxx[k,])
        q0 = sum(abs(ppid[,,idx0]-fd02[[ii]]$ppi[,,]))
        if(q0<q){ q = q0; idx = idx0 }
    }}

    tmp = fd02[[ii]]$ppi; tmpp = fd02[[ii]]$omega
    for(l in 1:nl){
      fd02[[ii]]$ppi[,,l] = tmp[,,idx[l]]
      fd02[[ii]]$omega[l] = tmpp[idx[l]]
    }

    fm2[ii,] = c(mean(abs(ppid-fd02[[ii]]$ppi)),
                 mean(abs(fp2[[ii]]$omega-fd02[[ii]]$omega)))
  }
} #ii

for(ii in 1:length(fr2)){
  if(chkresp2[ii]){
    nl = fnl2[ii]
    fg2[ii,] = c(fp2[[ii]]$bic,fd2[[which(ffnl2==nl)]]$bic)
  }
  frlab22[ii,1:length(frlab2[[ii]])] = frlab2[[ii]]
}

# output the model comparison result to a file
write.csv(data.frame(fnl2,frlab22,fg2,fm2),
          file.path(pathr,"MDcomparison2.csv"))

# ------------------------------------------------------------------------
# ANALYSIS WITH BEST MATCHING MODELS

fppb = list()
ii = 28 # best matching model
set.seed(365+fnl[ii]) # seed for RNG

# assessing model fit by parametric bootstrap
fppb[[ii]] = pblllcaNP(fp[[ii]],x,nk,fnl[ii],nboot=100)
fdpb[[which(ffnl==fnl[ii])]]
ppid = array(0,dim=c(nj,2,fnl[ii]))
for(j in 1:nj){ # dichotomize MC item response probabilities
  ppid[j,2,] = fp[[ii]]$ppi[j,anskey[j],]
  ppid[j,1,] = 1-ppid[j,2,]
}

# summary of parameter estimates
summary(fppb[[ii]]$alpha); sd(fppb[[ii]]$alpha)
summary(fppb[[ii]]$se.alpha); sd(fppb[[ii]]$se.alpha)
summary(fppb[[ii]]$beta); sd(fppb[[ii]]$beta)
summary(fppb[[ii]]$se.beta); sd(fppb[[ii]]$se.beta)

# compute GDI
# MC
```

```
cdimat   = cdi(fp[[ii]]$ppi,nk,10)
cdimatt  = apply(cdimat,c(1,2),sum) # test CDI
hcdimat  = gdi(cdimat)              # GDI
hcdimatt = apply(hcdimat,1,sum)     # test HCDI

# dichotomous
cdimat1   = cdi(fd0[[ii]]$ppi,nk1,10)
cdimat1t  = apply(cdimat1,c(1,2),sum) # test CDI
hcdimat1  = gdi(cdimat1)              # GDI
hcdimat1t = apply(hcdimat1,1,sum)     # test HCDI

# ------------------------------------------------------------------------
# ADAPTIVE TESTING SIMULATION WITH THE FINAL MODELS

pcmin = .85        # stopping criterion
set.seed(20090605) # seed for RNG
first.item = sample(1:nj,ni,replace=TRUE) # randomly select the 1st item

# classification by all items
postfp = postcls(x,fp[[ii]]$ppi)
postfd = postcls(x1,fd0[[ii]]$ppi)

# GDI item selection
catGDIfp = catclsGDIsim(x,hcdimat,
                        fp[[ii]]$ppi,pcmin=pcmin,first=first.item)
catGDIfd = catclsGDIsim(x1,hcdimat1,
                        fd0[[ii]]$ppi,pcmin=pcmin,first=first.item)

# ShE item selection
catShEfp = catclsSEsim(x,fp[[ii]]$ppi,pcmin=pcmin,first=first.item)
catShEfd = catclsSEsim(x1,fd0[[ii]]$ppi,pcmin=pcmin,first=first.item)

# RND item selection
catRNDfp = catclsRNDsim(x,fp[[ii]]$ppi,pcmin=pcmin,first=first.item)
catRNDfd = catclsRNDsim(x1,fd0[[ii]]$ppi,pcmin=pcmin,first=first.item)

# efficiency improvement
r0=summary(catRNDfp$njadm); r1=summary(catRNDfd$njadm)
g0=summary(catGDIfp$njadm); g1=summary(catGDIfd$njadm)
s0=summary(catShEfp$njadm); s1=summary(catShEfd$njadm)

1-g0/r0; 1-s0/r0          # adaptive
1-r0/r1; 1-g0/g1; 1-s0/s1 # MC

r0s=summary(catRNDfps$njadm); r1s=summary(catRNDfds$njadm)
g0s=summary(catGDIfps$njadm); g1s=summary(catGDIfds$njadm)
s0s=summary(catShEfps$njadm); s1s=summary(catShEfds$njadm)

1-g0s/r0s; 1-s0s/r0s            # adaptive
1-r0s/r1s; 1-g0s/g1s; 1-s0s/s1s # MC

# correct classification rates
clsfp = cbind(catShEfp$cls,catGDIfp$cls,catRNDfp$cls)
clsfd = cbind(catShEfd$cls,catGDIfd$cls,catRNDfd$cls)
apply(clsfp==postfp$cls,2,mean); apply(clsfd==postfp$cls,2,mean)
apply(clsfp==postfd$cls,2,mean); apply(clsfd==postfd$cls,2,mean)
mean(postfp$cls==postfd$cls)
```

```
c(mean(postfp$cls==catRNDfp$cls),mean(postfd$cls==catRNDfd$cls),
  mean(postfp$cls==catGDIfp$cls),mean(postfd$cls==catGDIfd$cls),
  mean(postfp$cls==catShEfp$cls),mean(postfd$cls==catShEfd$cls))
mean(postfp$cls==postfd$cls)

# ------------------------------------------------------------------------
# ITEM CHARACTERISTICS (RI, NRP, ID)

ichar = array(NA,dim=c(nj,3)); colnames(ichar) = c("RI","NPR","ID")
for(j in 1:nj){
  tmp = rbr0[rbr0[,1]==j,frlab[[ii]]]
  ichar[j,1] = sum(apply(tmp,1,sum)>0)
  ichar[j,2] = sum(apply(tmp,2,sum)>0)
  ichar[j,3] = mean(fp[[ii]]$ppi[j,,][tmp==1])
}
```

B.3 sub.r

```
# ------------------------------------------------------------------------
# sub.r
# R subroutines for LCM and adaptive testing
# lllcaNP, lllcaNPC, pblllcaNP, pblllcaNPC, getfreq, cdi, gdi, postcls,
# catclsGDIsim, catclsSEsim, catclsRNDsim

# ------------------------------------------------------------------------
# lllcaNP
# ML estimation for Linear Logistic LCM (NO PERSON COVARIATE)
# requires: getfreq(), lllca.c
# input values:
#   x        data matrix (ni * nj)
#   nk       # of categories for each item (nj * 1)
#            if a single number is given it is replicated nj times
#   nl       # of latent classes
#   y        design matrix for class probabilities (ni*nl) * p
#   yidx     subject index vector [il] ((ni*nl) * 1)
#   z        design matrix for conditional probabilities (ni*nj*nl*nk) * q
#   zidx     index matrix [ijkl] ((ni*nj*nl*nk) * 4)
#   alpha    vector of alpha parameters
#   beta     vector of beta parameters
#   maxiter  maximum number of EM iterations
#   maxiterm maximum number of M-step iterations
#   ep       convergence criterion for parameters
#   ed       convergence criterion for derivatives
#   el       convergence criterion for likelihood
#   rstart   TRUE = random starting values, FALSE = fixed starting values
# returned values:
#   beta  LOGIT parameter estimates for class probabilities (p*1)
#   alpha LOGIT parameter estimates for item response probabilities (q*1)
#   y, yidx
#   z, zidx
#   omega parameter estimates for class probabilities (p*1)
#   ppi   parameter estimates for item response probabilities (q*1)
#   ll    log-likelihood
#   g2    G2 statistic
```

```
#   np    # of parameters
#   aic   AIC
#   bic   BIC
#   conv  convergence flag
lllcaNP = function(x,nk,nl,
                   y=NULL,yidx=NULL,z=NULL,zidx=NULL,
                   alpha=NULL,beta=NULL,maxiter=1000,maxiterm=20,
                   ep=1.0e-03,ed=1.0e-03,el=1.0e-04,rstart=FALSE){
ni = nrow(x); nj = ncol(x)
ll = 0; conv = 0
if(length(nk)==1) nk = rep(nk,nj)
if(!is.null(y) & is.null(yidx)) stop("no yidx provided for y")

if(is.null(y)){ # create y and yidx for unconstrained model
  yidx = 1:nl
  y = rbind(diag(1,nl-1),rep(0,nl-1))
}
if(!is.null(z) & is.null(zidx)) stop("no zidx provided for z")

zidx = cbind(rep(1:nj,times=nk*nl),rep(rep(1:nl,nj),
             times=rep(nk,each=nl)),sequence(nk))
if(is.null(z)){ # create z and zidx for unconstrained model
  zidx = cbind(rep(1:nj,times=nk*nl),rep(rep(1:nl,nj),
               times=rep(nk,each=nl)),sequence(nk))
  na = sum(nk-1)*nl
  z = array(0,dim=c(sum(nk)*nl,sum(nk-1)*nl))
  s = t = 1
  for(j in 1:nj)
    for(l in 1:nl)
      for(k in 1:nk[j]){
        if(k<nk[j]){ z[s,t] = 1; t = t+1 }
        s = s+1
      }}

# remove zero columns in y and z if any
for(j in ncol(y)) y = y[,!apply(y,2,function(x) all(x==0))]
for(j in ncol(z)) z = z[,!apply(z,2,function(x) all(x==0))]

nb = ncol(y); ny = nrow(y)
if(is.null(beta) | length(beta)!=nb) beta = rep(0,nb)
na = ncol(z); nz = nrow(z)

# starting values
if(!is.null(alpha)){
  omega = exp(y%*%beta); omega = omega/sum(omega)
  ppi = array(0,dim=c(nj,max(nk),nl))
  hlx = array(0,dim=c(nl,ni))
  nkk = c(0,cumsum(nk)[-nj])
  for(l in 1:nl)
    for(j in 1:nj){
      zs = nkk[j]*nl+nk[j]*(l-1)+1; ze = zs + nk[j] - 1
      ppit = exp(z[zs:ze,]%*%alpha)
      ppi[j,1:nk[j],l] = ppit/sum(ppit)
    }
  hlx = t(postcls(x,ppi,omega)$pclass)
} else{
```

```
  hlx = matrix(rgamma(nl*ni,1,1),nrow=nl,ncol=ni)
  if(!rstart){
    xidx = order(getfreq(x)$freq,decreasing=TRUE)[1:nl]
    v = (getfreq(x)$x)[xidx,]
    for(l in 1:nl) hlx[l,] = apply(t(x)==v[l,],2,sum) + 1
  }
  shlx = apply(hlx,2,sum)
  hlx = t(t(hlx)/shlx)
}
if(is.null(alpha) | length(alpha)!=na) alpha = rep(0,na)

# call the EM algorithm routine
dyn.load(file.path(pathr,paste("lllca",.Platform$dynlib.ext,sep="")))
  f = .C("lllcaNP"
  x=as.integer(as.vector(x)),ni=as.integer(ni),nj=as.integer(nj),
  nk=as.integer(nk),nl=as.integer(nl),na=as.integer(na),nb=as.integer(nb),
  maxiter=as.integer(maxiter),maxiterm=as.integer(maxiterm),
  ep=as.double(ep),ed=as.double(ed),el=as.double(el),
  y=as.double(as.vector(t(y))),z=as.double(as.vector(t(z))),
  alpha=as.double(alpha),beta=as.double(beta),
  hlx=as.double(as.vector(hlx)),ll=as.double(ll),conv=as.integer(conv))
dyn.unload(file.path(pathr,paste("lllca,.Platform$dynlib.ext,sep="")))

np = na+nb
aic = -2*f$ll+2*np
bic = -2*f$ll+np*log(ni)
omega = exp(y%*%f$beta); omega = omega/sum(omega)
ppi = array(0,dim=c(nj,max(nk),nl))

for(j in nj:1) x = x[order(x[,j]),] # sort
idx = duplicated(x)
xx = unique(x); nr = nrow(xx); nx = rep(1,nr)
ii = 1
for(i in 2:ni)
  if(all(x[i,]==x[i-1,])){ nx[ii] = nx[ii]+1 } else{ ii = ii+1 }

hlx = array(0,dim=c(nl,nr))
nkk = c(0,cumsum(nk)[-nj])
for(i in 1:nr){
  for(l in 1:nl)
    for(j in 1:nj){
      zs = nkk[j]*nl+nk[j]*(l-1)+1; ze = zs + nk[j] - 1
      ppit = exp(z[zs:ze,]%*%f$alpha)
      ppi[j,1:nk[j],l] = ppit/sum(ppit)
      hlx[l,i] = hlx[l,i] + log(ppi[j,xx[i,j],l])
    }
  hlx[,i] = hlx[,i] + log(omega)
} #i
hlx = exp(hlx); shlx = apply(hlx,2,sum)
hlx = t(t(hlx)/shlx)
g2 = 2*( nx%*%(log(nx)-log(shlx)) )

return(list(beta=f$beta,alpha=f$alpha,
            y=y,yidx=yidx,z=z,zidx=zidx,
            omega=omega,ppi=ppi,
```

```
            ll=f$ll,g2=g2,np=np,aic=aic,bic=bic,conv=f$conv))
} # lllcaNP

# ------------------------------------------------------------------------
# lllcaNPC
# ML estimation for Linear Logistic LCM (NO PERSON COVARIATE;
# FIXED ALPHA ALLOWED)
# requires: getfreq(), lllca.c
# input values:
#   x        raw data matrix (ni * nj)
#   nk       # of categories for each item (nj * 1)
#            if a single number is given it is replicated nj times
#   nl       # of latent classes
#   y        design matrix for class probabilities (ni*nl) * p
#   yidx     subject index vector [il] ((ni*nl) * 1)
#   betaf    vector of flags for fixed beta parameters
#   beta     vector of fixed values of beta parameters
#   z        design matrix for conditional probabilities (ni*nj*nl*nk) * q
#   zidx     index matrix [ijkl] ((ni*nj*nl*nk) * 4)
#   alphaf   vector of flags for fixed alpha parameters
#   alpha    vector of fixed values of alpha parameters
#   maxiter  max number of EM iterations
#   maxiterm max number of M-step iterations
#   ep       convergence criterion for parameters
#   ed       convergence criterion for derivatives
#   el       convergence criterion for likelihood
#   rstart   TRUE = random starting values, FALSE = fixed starting values
# returned values:
#   beta  LOGIT parameter estimates for class probabilities (p*1)
#   alpha LOGIT parameter estimates for item response probabilities (q*1)
#   y, yidx
#   z, zidx
#   omega parameter estimates for class probabilities (p*1)
#   ppi   parameter estimates for item response probabilities (q*1)
#   ll    log-likelihood
#   g2    G2 statistic
#   np    # of parameters
#   aic   AIC
#   bic   BIC
#   conv  convergence flag
lllcaNPC = function(x,nk,nl,
                    y=NULL,yidx=NULL,betaf=NULL,beta=NULL,
                    z,zidx,alphaf,alpha,
                    maxiter=1000,maxiterm=20,
                    ep=1.0e-03,ed=1.0e-03,el=1.0e-04,rstart=FALSE){
ni = nrow(x); nj = ncol(x)
ll = 0; conv = 0

if(length(nk)==1) nk = rep(nk,nj)
if(!is.null(y) & is.null(yidx)) stop("yidx must be provided with y")

if(is.null(y)){ # create y and yidx for unconstrained model
  yidx = 1:nl
  y = rbind(diag(1,nl-1),rep(0,nl-1))
}
if(is.null(z) | is.null(zidx)) stop("both z and aidx must be provided")

# remove zero columns in y and z if any
```

```
for(j in ncol(y)) y = y[,!apply(y,2,function(x) all(x==0))]
for(j in ncol(z)) z = z[,!apply(z,2,function(x) all(x==0))]

nb = ncol(y); ny = nrow(y)
if(is.null(beta) | length(beta)!=nb) beta = rep(0,nb)
na = ncol(z); nz = nrow(z)

# starting values
if(!is.null(alpha)){
  omega = exp(y%*%beta); omega = omega/sum(omega)
  ppi = array(0,dim=c(nj,max(nk),nl))
  hlx = array(0,dim=c(nl,ni))
  nkk = c(0,cumsum(nk)[-nj])
  for(l in 1:nl)
    for(j in 1:nj){
      zs = nkk[j]*nl+nk[j]*(l-1)+1; ze = zs + nk[j] - 1
      ppit = exp(z[zs:ze,]%*%alpha)
      ppi[j,1:nk[j],l] = ppit/sum(ppit)
    }
  hlx = t(postcls(x,ppi,omega)$pclass)
} else{
  hlx = matrix(rgamma(nl*ni,1,1),nrow=nl,ncol=ni)
  if(!rstart){
    xidx = order(getfreq(x)$freq,decreasing=TRUE)[1:nl]
    v = (getfreq(x)$x)[xidx,]
    for(l in 1:nl) hlx[l,] = apply(t(x)==v[l,],2,sum) + 1
  }
  shlx = apply(hlx,2,sum)
  hlx = t(t(hlx)/shlx)
}

# call the EM algorithm routine
dyn.load(file.path(pathr,paste("lllca",.Platform$dynlib.ext,sep="")))
  f = .C("lllcaNPC",
  x=as.integer(as.vector(x)),ni=as.integer(ni),nj=as.integer(nj),
  nk=as.integer(nk),nl=as.integer(nl),na=as.integer(na),nb=as.integer(nb),
  maxiter=as.integer(maxiter),maxiterm=as.integer(maxiterm),
  ep=as.double(ep),ed=as.double(ed),el=as.double(el),
  y=as.double(as.vector(t(y))),z=as.double(as.vector(t(z))),
  alpha=as.double(alpha),alphaf=as.integer(alphaf),beta=as.double(beta),
  hlx=as.double(as.vector(hlx)),ll=as.double(ll),conv=as.integer(conv))
dyn.unload(file.path(pathr,paste("lllca",.Platform$dynlib.ext,sep="")))

np = (na-sum(alphaf))+nb
aic = -2*f$ll+2*np
bic = -2*f$ll+np*log(ni)
omega = exp(y%*%f$beta); omega = omega/sum(omega)
ppi = array(0,dim=c(nj,max(nk),nl))

for(j in nj:1) x = x[order(x[,j]),] # sort
idx = duplicated(x)
xx = unique(x); nr = nrow(xx); nx = rep(1,nr)
ii = 1
for(i in 2:ni)
  if(all(x[i,]==x[i-1,])){ nx[ii] = nx[ii]+1 } else{ ii = ii+1 }
```

```
hlx = array(0,dim=c(nl,nr))
nkk = c(0,cumsum(nk)[-nj])
for(i in 1:nr){
  for(l in 1:nl)
    for(j in 1:nj){
      zs = nkk[j]*nl+nk[j]*(l-1)+1; ze = zs + nk[j] - 1
      ppit = exp(z[zs:ze,]%*%f$alpha)
      ppi[j,1:nk[j],l] = ppit/sum(ppit)
      hlx[l,i] = hlx[l,i] + log(ppi[j,xx[i,j],l])
    }
  hlx[,i] = hlx[,i] + log(omega)
}
hlx = exp(hlx); shlx = apply(hlx,2,sum)
hlx = t(t(hlx)/shlx)
g2 = 2*( nx%*%(log(nx)-log(shlx)) )

return(list(beta=f$beta,alpha=f$alpha,
            y=y,yidx=yidx,z=z,zidx=zidx,
            omega=omega,ppi=ppi,
            ll=f$ll,g2=g2,np=np,aic=aic,bic=bic,conv=f$conv))
} # lllcaNPC

# ------------------------------------------------------------------------
# pblllcaNP
# parametric bootstrap for lllcaNP (NO PERSON COVARIATE)
# requires: lllcaNP(), lllca.c
# input values:
#   f     a lllcaNP object
#   x     original data matrix
#   nk    # of response options
#   nl    # of classes
#   nboot bootstrap sample size
# returned values:
#   alpha    original alpha estimates
#   se.alpha standard error estimates for alpha
#   beta     original beta estimates
#   se.beta  standard error estimates for beta
#   p.g2     p-value for the G2 statistic
#   g2       vector of all bootstrap G2 statistics
pblllcaNP = function(f,x,nk,nl,nboot=100){
nn = nrow(x); nj = ncol(x); x0 = x
alpha = array(0,dim=c(nboot,length(f$alpha)))
beta = array(0,dim=c(nboot,length(f$beta)))
g2 = rep(0,nboot)

s = 1; tott = 1
while(s <= nboot){
  cc = t((1:nl)%*%rmultinom(nn,1,f$omega))
  for(i in 1:nn)
    for(j in 1:nj)
      x0[i,j] = (1:nk[j])%*%rmultinom(1,1,f$ppi[j,1:nk[j],cc[i]])
  f0 = lllcaNP(x0,nk,nl,y=NULL,yidx=NULL,z=f$z,zidx=f$zidx,
               alpha=f$alpha,beta=f$beta)

  if(f0$conv==1){
    alpha[s,] = (f0$alpha-f$alpha)^2
```

```
    beta[s,] = (f0$beta-f$beta)^2
    g2[s] = f0$g2
    s = s + 1
  }
  tott = tott + 1
} #s

p.g2 = sum(g2>as.vector(f$g2))/nboot
se.alpha = sqrt(apply(alpha,2,mean))
se.beta = sqrt(apply(beta,2,mean))

return(list(alpha=f$alpha,se.alpha=se.alpha,
            beta=f$beta,se.beta=se.beta,p.g2=p.g2,g2=g2))
} # pblllcaNP

# ------------------------------------------------------------------------
# pblllcaNPC
# parametric bootstrap for lllcaNPC (NO PERSON COVARIATE;
# FIXED ALPHA ALLOWED)
# input values:
#   f      lllcaNP object
#   x      original data matrix
#   nk     # of response options
#   nl     # of classes
#   alphav alpha parameters
#   alphaf vector of flags for fixed alpha parameters
#   betav  beta parameters
#   nboot  bootstrap sample size
# returned values:
#   alpha    original alpha estimates
#   se.alpha standard error estimates for alpha
#   beta     original beta estimates
#   se.beta  standard error estimates for beta
#   p.g2     p-value for the G2 statistic
#   g2       vector of all bootstrap G2 statistics
pblllcaNPC = function(f,x,nk,nl,alphav,alphaf,betav,nboot=100){
nn = nrow(x); nj = ncol(x); x0 = x
alpha = array(0,dim=c(nboot,length(f$alpha)))
beta = array(0,dim=c(nboot,length(f$beta)))
g2 = rep(0,nboot)

s = 1; tott = 1
while(s <= nboot){
  cc = t((1:nl)%*%rmultinom(nn,1,f$omega))
  for(i in 1:nn)
    for(j in 1:nj)
      x0[i,j] = (1:nk[j])%*%rmultinom(1,1,f$ppi[j,1:nk[j],cc[i]])

  f0 = lllcaNPC(x0,nk,nl,y=NULL,yidx=NULL,beta=f$beta,
                z=f$z,zidx=f$zidx,alphaf=alphaf,alpha=alphav)
# rearrange class probabilities
  idx = 1:nl
  for(l in 1:nl){
    q = rep(0,nl)
    for(ll in 1:nl) q[ll] = sum(abs(f0$ppi[,,l]-f$ppi[,,ll]))
    idx[l] = which.min(q)
  }
  if(!all(sort(idx)==(1:nl))){
```

```
    idxx = permutations(nl,nl); idx = idxx[1,]
    q = sum(abs(f0$ppi-f$ppi))
    for(k in 2:nrow(idxx)){
      idx0 = as.vector(idxx[k,])
      q0 = sum(abs(f0$ppi[,,idx0]-f$ppi[,,]))
      if(q0<q){ q = q0; idx = idx0 }
    }}

  tmp = f0$ppi; tmpp = f0$omega;
  tmppp = f0$alpha; tmpppp = f0$beta
  for(l in 1:nl){
    f0$ppi[,,idx[l]] = tmp[,,l]
    f0$omega[idx[l]] = tmpp[l]
  }

  for(l in 1:nl)
    f0$alpha[f0$zidx[f0$zidx[,3]!=nk[1],2]==idx[l]]
                      = tmppp[f0$zidx[f0$zidx[,3]!=nk[1],2]==l]

  betav = c(f0$beta,0); betabse = betav[idx==nl]
  for(l in 1:nl) betav[idx[l]] = (c(tmpppp,0)-betabse)[l]
  f0$beta = betav[-nl]
  if(f0$conv==1){
    alpha[s,] = (f0$alpha-f$alpha)^2
    beta[s,] = (f0$beta-f$beta)^2
    g2[s] = f0$g2
    s = s + 1
  }
  tott = tott + 1
} #s

p.g2 = sum(g2>as.vector(f$g2))/nboot
se.alpha = sqrt(apply(alpha,2,mean))
se.beta = sqrt(apply(beta,2,mean))

return(list(alpha=f$alpha,se.alpha=se.alpha,
            beta=f$beta,se.beta=se.beta,p.g2=p.g2,g2=g2))
} # pblllcaNPC

# ------------------------------------------------------------------------
# getfreq
# get unique response patterns and corresponding observed frequencies for
# a given MC/dichotomous response data matrix
# input value:
#   x    data frame or matrix
# returned values:
#   x    unique observed response patterns
#   freq frequency vector corresponding to x
getfreq = function(x){
x = as.matrix(x); nc = ncol(x); nr = nrow(x)
for(j in nc:1) x = x[order(x[,j]),] # sort
idx = duplicated(x); xuniq = unique(x)
freq = pnum = rep(1,nrow(xuniq))
i = j = 1
while(j<=nr){
  if(idx[j]==FALSE){ i = i+1 } else{ freq[i-1] = freq[i-1]+1 }
  j = j+1
```

```
}
return(list(x=xuniq,freq=freq))
} # getfreq

# ------------------------------------------------------------------------
# cdi
# cognitive diagnosis index matrix
# input values:
#   pp      conditional response probability matrix (nj * max(nk) * nl)
#   nk      # of categories for each item (nj * 1)
#   base    base for the logarithm in KL divergence
# returned values:
#   CDI matrices (nl * nl * nj)
cdi = function(pp,nk,base=exp(1)){
nj = dim(pp)[1]; nl = dim(pp)[3]
cdimat = array(0,dim=c(nl,nl,nj))
for(j in 1:nj)
  for(l in 1:nl)
    cdimat[l,,j] = apply(-pp[j,,l][1:nk[j]]*(log(pp[j,,][1:nk[j],],
                         base=base)-log(pp[j,,l][1:nk[j]],base=base)),
                         2,sum)
return(cdimat)
} # cdi

# ------------------------------------------------------------------------
# gdi
# compute global cognitive diagnosis index
# input values:
#   cdi output from cdi()
#   w   weight vector
# returned values:
#   GDI matrix (nl * nj)
gdi = function(cdi,w=rep(1,dim(cdi)[1])){
nl = dim(cdi)[1]; nj = dim(cdi)[3]
gdimat = array(0,dim=c(nl,nj))
for(j in 1:nj)
  for(l in 1:nl)
    gdimat[l,j] = cdi[l,-l,j] %*% w[-l]
return(gdimat)
} # gdi

# ------------------------------------------------------------------------
# postcls
# calculate posterior class probabilities for polytomous LCM
#   x      response matrix (ni * nj)
#   nk     # of categories for each item (nj * 1)
#   ppi    conditional response probability matrix (nj * max(nk) * nl)
#   omega  prior class probability (nl * 1)
# returned values:
#   pclass posterior probability matrix (ni * nl)
#   cls    MAP classification (ni * 1)
postcls = function(x,ppi,omega=rep(1/nl,nl)){
x = as.matrix(x)
ni = nrow(x); nj = ncol(x); nl = dim(ppi)[3]
cls = rep(0,ni)
pclass = array(0,dim=c(ni,nl))
```

```
for(i in 1:ni){
  for(l in 1:nl){
    pclass[i,l] = 0
    for(j in 1:nj)
      pclass[i,l] = pclass[i,l] + log(ppi[j,x[i,j],l])
  }
  pclass[i,] = pclass[i,] + log(omega)
}
pclass = exp(pclass)
spclass = apply(pclass,1,sum)
pclass = pclass/spclass
for(i in 1:ni) cls[i] = which.max(pclass[i,])
return(list(pclass=pclass,cls=cls))
} # postcls

# ------------------------------------------------------------------------
# catclsGDIsim
# perform CAT classification using GDI
#   x      response matrix (ni * nj)
#   gdi    global diagnostic index (nl * nj)
#   ppi    item response probability matrix (nj * max(nk) * nl)
#   pcmin  stop if posterior > pcmin
#   omega  prior class probability (nl * 1)
#   first  list of items to be administered FIRST (ni * 1)
#   rseed  seed for initial random item selection
#          (valid only when first == NULL)
# returned values:
#   pclass posterior probability matrix (ni * nl)
#   iidx   items which were administered (ni * nj)
#   cls    MAP classification (ni * 1)
#   clsd   whether njadm < nj (ni * 1)
#   njadm  number of items administered (ni * 1)
catclsGDIsim = function(x,gdi,ppi,pcmin=.90,
                        omega=rep(1/nl,nl),first=NULL,rseed=NULL){
x = as.matrix(x)
ni = nrow(x); nj = ncol(x); nl = dim(ppi)[3]
cls = clsd = rep(0,ni)
njadm = rep(0,ni)
pclass = array(0,dim=c(ni,nl))
iidx = array(0,dim=c(ni,nj))

# first item = random selection
if(!is.null(rseed)) set.seed(rseed)
if(is.null(first)) for(i in 1:ni) iidx[i,sample(1:nj,1)] = 1
else for(i in 1:ni) iidx[i,first[i]] = 1

for(i in 1:ni){
  while(clsd[i]==0 & njadm[i]<nj){
    njadm[i] = njadm[i] + 1
    for(l in 1:nl){ # update the posterior class probabilities
      pclass[i,l] = 0
      for(j in (1:nj)[iidx[i,]>0]){
        pclass[i,l] = pclass[i,l] + log(ppi[j,x[i,j],l])
        cls[i] = j
      }}
```

```
    pclass[i,] = exp(pclass[i,] + log(omega))
    spclass = sum(pclass[i,])
    pclass[i,] = pclass[i,]/spclass
    cls[i] = which.max(pclass[i,])
    if(pclass[i,cls[i]]>pcmin){ clsd[i] = 1 # reached a diagnosis
    } else{ # select a next item with largest GDI
      if(njadm[i]<nj){
        iidx[i,(1:nj)[iidx[i,]==0][which.max(gdi[cls[i],
                                            iidx[i,]==0])]] = njadm[i]+1
    }}}}

return(list(pclass=pclass,iidx=iidx,cls=cls,clsd=clsd,njadm=njadm))
} # catclsGDIsim

# ------------------------------------------------------------------------
# catclsSEsim
# perform CAT classification using pre-posterior Shannon Entropy
#   x      response matrix (ni * nj)
#   ppi    conditional response probability matrix (nj * max(nk) * nl)
#   pcmin  stop if posterior > pcmin
#   omega  prior class probability (nl * 1)
#   first  list of items to be administered FIRST (ni * 1)
#   rseed  seed for initial random item selection
#          (valid only when first == NULL)
# returned values:
#   pclass posterior probability matrix (ni * nl)
#   iidx   items which were administered (ni * nj)
#   cls    MAP classification (ni * 1)
#   clsd   whether njadm < nj (ni * 1)
#   njadm  number of items administered (ni * 1)
catclsSEsim = function(x,ppi,pcmin=.90,
                       omega=rep(1/nl,nl),first=NULL,rseed=NULL){
x = as.matrix(x)
ni = nrow(x); nj = ncol(x); nl = dim(ppi)[3]
cls = clsd = rep(0,ni)
njadm = rep(0,ni)
pclass = array(0,dim=c(ni,nl))
iidx = array(0,dim=c(ni,nj))

# first item = random selection
if(!is.null(rseed)) set.seed(rseed)
if(is.null(first)) for(i in 1:ni) iidx[i,sample(1:nj,1)] = 1
else for(i in 1:ni) iidx[i,first[i]] = 1

for(i in 1:ni){
  while(clsd[i]==0 & njadm[i]<nj){
    njadm[i] = njadm[i] + 1
    for(l in 1:nl){ # update the posterior class probabilities
      pclass[i,l] = 0
      for(j in (1:nj)[iidx[i,]>0]){
        pclass[i,l] = pclass[i,l] + log(ppi[j,x[i,j],l])
        cls[i] = j
      }}
    pclass[i,] = exp(pclass[i,] + log(omega))
    spclass = sum(pclass[i,])
```

```
    pclass[i,] = pclass[i,]/spclass # current posterior probability
    cls[i] = which.max(pclass[i,]) # current class

    if(pclass[i,cls[i]]>pcmin){ clsd[i] = 1 # reached a diagnosis
    } else{ # select a next item with largest NEGATIVE ShE
      if(njadm[i]<nj){
        ShE = rep(NA,nj)
        for(j in (1:nj)[iidx[i,]==0]){
          ppclass = exp(log(pclass[i,])+t(log(ppi[j,,])))
          ppclass = t(ppclass)/apply(ppclass,2,sum)
          ShE[j] = apply(log(ppclass)*ppclass,
                         1,sum)%*%ppi[j,,]%*%pclass[i,]
        }
        iidx[i,which.max(ShE)] = njadm[i]+1
      }}}}

return(list(pclass=pclass,iidx=iidx,cls=cls,clsd=clsd,njadm=njadm))
} # catclsSEsim

# ------------------------------------------------------------------------
# catclsRNDsim
# perform CAT classification using pre-posterior Shannon Entropy
#   x      response matrix (ni * nj)
#   nk     # of categories for each item (nj * 1)
#   ppi    conditional response probability matrix (nj * max(nk) * nl)
#   pcmin  stop if posterior > pcmin
#   omega  prior class probability (nl * 1)
#   first  list of items to be administered FIRST (ni * 1)
#   rseed  seed for initial random item selection
#          (valid only when first == NULL)
# returned values:
#   pclass posterior probability matrix (ni * nl)
#   iidx   items which were administered (ni * nj)
#   cls    MAP classification (ni * 1)
#   clsd   whether njadm < nj (ni * 1)
#   njadm  number of items administered (ni * 1)
catclsRNDsim = function(x,ppi,pcmin=.90,
                        omega=rep(1/nl,nl),first=NULL,rseed=NULL){
x = as.matrix(x)
ni = nrow(x); nj = ncol(x); nl = dim(ppi)[3]
cls = clsd = rep(0,ni)
njadm = rep(0,ni)
pclass = array(0,dim=c(ni,nl))
iidx = array(0,dim=c(ni,nj))

# first item = random selection
if(!is.null(rseed)) set.seed(rseed)
if(is.null(first)) for(i in 1:ni) iidx[i,sample(1:nj,1)] = 1
else for(i in 1:ni) iidx[i,first[i]] = 1

for(i in 1:ni){
  while(clsd[i]==0 & njadm[i]<nj){
    njadm[i] = njadm[i] + 1
    for(l in 1:nl){ # update the posterior class probabilities
      pclass[i,l] = 0
```

```
      for(j in (1:nj)[iidx[i,]>0]){
        pclass[i,l] = pclass[i,l] + log(ppi[j,x[i,j],l])
        cls[i] = j
      }}
    pclass[i,] = exp(pclass[i,] + log(omega))
    spclass = sum(pclass[i,])
    pclass[i,] = pclass[i,]/spclass # current posterior probability
    cls[i] = which.max(pclass[i,]) # current class

    if(pclass[i,cls[i]]>pcmin){ clsd[i] = 1 # reached a diagnosis
    } else{ # select a next item RANDOMLY
      if(njadm[i]<nj){
        if(sum(iidx[i,]==0)>1) temp = sample((1:nj)[iidx[i,]==0],1)
        else temp = (1:nj)[iidx[i,]==0]
        iidx[i,temp] = njadm[i]+1
      }}}}

return(list(pclass=pclass,iidx=iidx,cls=cls,clsd=clsd,njadm=njadm))
} # catclsRNDsim
```

B.4 lllca.c

```
/* ------------------------------------------------------------------------
lllca.c
Parameter estimation for linear logistic LCM without person covariates
------------------------------------------------------------------------ */
#include <R.h>
#include <Rmath.h>
#include <R_ext/blas.h>
#define SSIZE .50

/* LLLCA */
void lllcaNP(int *x, int *ni, int *nj, int *nk, int *nl, int *na, int *nb,
int *maxiter, int *maxiterm, double *ep, double *ed, double *el,
double *y, double *z, double *alpha, double *beta,
double *hlx, double *ll, int *conv)
{
  int i,j,k,l,jlk,u,v,uv,t=0,tm=0,convm=0,convs=0,nz;
  int incx=1,incy=1;
  int *nnk;
  char uplo='u', trst='t';
  double ll0, tmpd=0.0, tmpdd=0.0, a=1.0, b=0.0, s, ss=SSIZE;
  double q_alpha=0.0, q_beta=0.0, q_alpha0=0.0, q_beta0=0.0;
  double *shlx, *omega, *lomega, *ppi, *lppi;
  double *alpha0,*alpham,*alpham0;
  double *alphad,*alphad0,*alphah,*alphaht,*d_alpha,*d_alphad;
  double *beta0,*betam,*betam0;
  double *betad,*betad0,*betah,*betaht,*d_beta,*d_betad;

  /* INITIALIZATION */

  nnk = (int *) malloc(nj[0] * sizeof(int));
  nnk[0] = 0; nz = nk[0];
  for(j=1; j<nj[0]; j++){
    nnk[j] = nnk[j-1]+nk[j-1]*nl[0];
```

```
    nz += nk[j];
  }
  nz *= nl[0];

  alpha0   = (double *) malloc(na[0] * sizeof(double));
  alpham   = (double *) malloc(na[0] * sizeof(double));
  alpham0  = (double *) malloc(na[0] * sizeof(double));
  alphad   = (double *) malloc(na[0] * sizeof(double));
  alphad0  = (double *) malloc(na[0] * sizeof(double));
  alphah   = (double *) malloc(na[0]*na[0] * sizeof(double));
  alphaht  = (double *) malloc(na[0]*na[0] * sizeof(double));
  d_alpha  = (double *) malloc(na[0] * sizeof(double));
  d_alphad = (double *) malloc(na[0] * sizeof(double));
  beta0    = (double *) malloc(nb[0] * sizeof(double));
  betam    = (double *) malloc(nb[0] * sizeof(double));
  betam0   = (double *) malloc(nb[0] * sizeof(double));
  betad    = (double *) malloc(nb[0] * sizeof(double));
  betad0   = (double *) malloc(nb[0] * sizeof(double));
  betah    = (double *) malloc(nb[0]*nb[0] * sizeof(double));
  betaht   = (double *) malloc(nb[0]*nb[0] * sizeof(double));
  d_beta   = (double *) malloc(nb[0] * sizeof(double));
  d_betad  = (double *) malloc(nb[0] * sizeof(double));
  omega    = (double *) malloc(nl[0] * sizeof(double));
  lomega   = (double *) malloc(nl[0] * sizeof(double));
  ppi      = (double *) malloc(nz * sizeof(double));
  lppi     = (double *) malloc(nz * sizeof(double));
  shlx     = (double *) malloc(nl[0] * sizeof(double));

  for(u=0; u<nb[0]; u++){
    beta0[u] = beta[u]; betad0[u] = beta[u];
  }
  for(u=0; u<na[0]; u++){
    alpha0[u] = alpha[u]; alphad0[u] = alpha[u];
  }
  ll0 = 0.0;
  for(j=0; j<nj[0]; j++) ll0 += log(1.0/nk[j]);
  ll0 *= ni[0]; *ll = ll0;

  Rprintf("--------------------------------------------\n");
  Rprintf("EM Algorithm for LLLCM (No Person Covariates)\n");
  Rprintf("Initial Log-Likelihood = %f\n",ll0);
  Rprintf("--------------------------------------------\n");

/* EM algorithm */

*conv = 0;
while(*conv==0 && t<maxiter[0]){

  F77_CALL(dcopy)(nb,beta,&incx,beta0,&incy);
  F77_CALL(dcopy)(na,alpha,&incx,alpha0,&incy);

// E-step
  tmpd = 0.0;
  F77_CALL(dgemv)(&trst,nb,nl,&a,y,nb,beta,&incx,&b,lomega,&incy);
  for(l=0; l<nl[0]; l++){
    shlx[l] = 0.0;
    for(i=0; i<ni[0]; i++) shlx[l] += hlx[l + nl[0]*i];
```

```
    omega[l] = exp(lomega[l]);
    tmpd += omega[l];
  }
  q_beta0 = 0.0;
  for(l=0; l<nl[0]; l++){
    omega[l] /= tmpd;
    lomega[l] -= log(tmpd);
    q_beta0 += shlx[l]*lomega[l];
  }

  F77_CALL(dgemv)(&trst,na,&nz,&a,z,na,alpha,&incx,&b,lppi,&incy);
  for(j=0; j<nj[0]; j++){
    for(l=0; l<nl[0]; l++){
      tmpd = 0.0;
      for(k=0; k<nk[j]; k++){
        jlk = k + nk[j]*l + nnk[j];
        ppi[jlk] = exp(lppi[jlk]);
        tmpd += ppi[jlk];
      }
      for(k=0; k<nk[j]; k++){
        jlk = k + nk[j]*l + nnk[j];
        ppi[jlk] /= tmpd;
        lppi[jlk] -= log(tmpd);
      }
    }
  }
  q_alpha0 = 0.0;
  for(i=0; i<ni[0]; i++)
    for(l=0; l<nl[0]; l++)
      for(j=0; j<nj[0]; j++)
        q_alpha0 +=
          hlx[l + nl[0]*i]*lppi[ (x[i+ni[0]*j]-1) + nk[j]*l + nnk[j] ];

// M-step: beta
  for(u=0; u<nb[0]; u++)
    for(v=0; v<nb[0]; v++) betah[u + nb[0]*v] = (u==v)*1.0;

  F77_CALL(dcopy)(nb,beta,&incx,betam,&incy);
  for(u=0; u<nb[0]; u++){
    betad[u] = 0.0;
    for(l=0; l<nl[0]; l++) betad[u] +=
      y[nb[0]*l + u]*(shlx[l] - ni[0]*omega[l]);
  }
  tm = 0; convm = 0;

while(convm==0){
  F77_CALL(dcopy)(nb,betam,&incx,betam0,&incy);

// M-step: beta: convergence check
  tmpd = fabs(betad[F77_CALL(idamax)(nb,betad,&incx)-1]);
  if(tmpd<ed[0]){
    convm = 1;
    F77_CALL(dcopy)(nb,betam,&incx,beta,&incy);
  }

// M-step: beta: optimization by BFGS
```

```
  if(convm==0){
    convs = 0;
    if(tmpd>1.0/ss) s = 1.0 / tmpd; else s = ss;

    while(convs==0){
      F77_CALL(dsymv)(&uplo,nb,&s,betah,nb,betad,&incx,&b,betam,&incy);
      tmpdd = fabs(betam[F77_CALL(idamax)(nb,betam,&incx)-1]);
      F77_CALL(daxpy)(nb,&a,betam0,&incx,betam,&incy);

// renew the Q function
      tmpd = 0.0;
      F77_CALL(dgemv)(&trst,nb,nl,&a,y,nb,betam,&incx,&b,lomega,&incy);
      for(l=0; l<nl[0]; l++){
        omega[l] = exp(lomega[l]);
        tmpd += omega[l];
      }
      q_beta = 0.0;
      for(l=0; l<nl[0]; l++){
        omega[l] /= tmpd;
        lomega[l] -= log(tmpd);
        q_beta += shlx[l]*lomega[l];
      }

      if(q_beta>=q_beta0){
        convs = 1;
        q_beta0 = q_beta;
      }
      else{
        if(tmpdd<ep[0]){ // failed to increase Q -> exit
          convs = 1; convm = 1;
          F77_CALL(dcopy)(nb,betam0,&incx,betam,&incy);
          q_beta = q_beta0;
        }
        else s *= SSIZE;
      }
    } // M-step: while(convs==0)

// renew H
    if(convm==0){
      for(u=0; u<nb[0]; u++){
        betad0[u] = betad[u];
        betad[u]  = 0.0;
        for(l=0; l<nl[0]; l++) betad[u] +=
          y[nb[0]*l + u]*(shlx[l] - ni[0]*omega[l]);
        d_beta[u]  = betam[u] - betam0[u];
        d_betad[u] = betad[u] - betad0[u];
      }

      tmpd = F77_CALL(ddot)(nb,d_beta,&incx,d_betad,&incy);
        for(u=0; u<nb[0]; u++){
          for(v=0; v<nb[0]; v++){
            uv = u + nb[0]*v;
            betaht[uv] = 0.0;
            for(k=0; k<nb[0]; k++) betaht[uv] +=
              ((u==k)*1.0 - d_beta[u]*d_betad[k]/tmpd)*betah[k + nb[0]*v];
          }
        }
        for(u=0; u<nb[0]; u++){
```

```
          for(v=u; v<nb[0]; v++){
            uv = u + nb[0]*v;
            betah[uv] = d_beta[u]*d_beta[v]/tmpd;
            for(k=0; k<nb[0]; k++) betah[uv] +=
              betaht[u + nb[0]*k]*((k==v)*1.0-d_betad[k]*d_beta[v]/tmpd);
            betah[v + nb[0]*u] = betah[uv];
          }
        }
        F77_CALL(dcopy)(nb,betad,&incx,betad0,&incy);
    } // M-step: if(convm==0)
  } // M-step: if(convm==0)

  if(tm>=maxiterm[0]-1) convm = 1;
  tm += 1;
  if(convm==1) F77_CALL(dcopy)(nb,betam,&incx,beta,&incy);
} // M-step: beta: END: while(convm==0)

// M-step: alpha
  for(u=0; u<na[0]; u++)
    for(v=0; v<na[0]; v++) alphah[u + na[0]*v] = (u==v)*1.0;

  F77_CALL(dcopy)(na,alpha,&incx,alpham,&incy);
  for(u=0; u<na[0]; u++){
    alphad[u] = 0.0;
    for(i=0; i<ni[0]; i++){
      for(l=0; l<nl[0]; l++){
        tmpd = 0.0;
        for(j=0; j<nj[0]; j++){
          for(k=0; k<nk[j]; k++){
            jlk = k + nk[j]*l + nnk[j];
            tmpd += z[na[0]*jlk + u]*((x[i + ni[0]*j]==k+1) - ppi[jlk]);
          }
        }
        alphad[u] += tmpd * hlx[l + nl[0]*i];
      }
    }
  }
  tm = 0; convm = 0;

while(convm==0){
  F77_CALL(dcopy)(na,alpham,&incx,alpham0,&incy);

// M-step: alpha: convergence check
  tmpd = fabs(alphad[F77_CALL(idamax)(na,alphad,&incx)-1]);
  if(tmpd<ed[0]){
    convm = 1;
    F77_CALL(dcopy)(na,alpham,&incx,alpha,&incy);
  }

// M-step: alpha: optimization by BFGS
  if(convm==0){
    convs = 0;
    if(tmpd>1.0/ss) s = 1.0 / tmpd; else s = ss;

    while(convs==0){
      F77_CALL(dsymv)(&uplo,na,&s,alphah,na,alphad,&incx,&b,alpham,&incy);
      tmpdd = fabs(alpham[F77_CALL(idamax)(na,alpham,&incx)-1]);
```

```
      F77_CALL(daxpy)(na,&a,alpham0,&incx,alpham,&incy);

// M-step: alpha: renew the Q functions
      F77_CALL(dgemv)(&trst,na,&nz,&a,z,na,alpham,&incx,&b,lppi,&incy);
      for(j=0; j<nj[0]; j++){ // pi
        for(l=0; l<nl[0]; l++){
          tmpd = 0.0;
          for(k=0; k<nk[j]; k++){
            jlk = k + nk[j]*l + nnk[j];
            ppi[jlk] = exp(lppi[jlk]);
            tmpd += ppi[jlk];
          }
          for(k=0; k<nk[j]; k++){
            jlk = k + nk[j]*l + nnk[j];
            ppi[jlk] /= tmpd;
            lppi[jlk] -= log(tmpd);
          }
        }
      }
      q_alpha = 0.0;
      for(i=0; i<ni[0]; i++)
        for(l=0; l<nl[0]; l++)
          for(j=0; j<nj[0]; j++)
            q_alpha +=
              hlx[l+nl[0]*i]*lppi[ (x[i+ni[0]*j]-1) + nk[j]*l + nnk[j] ];

      if(q_alpha>=q_alpha0){
        convs = 1;
        q_alpha0 = q_alpha;
      }
      else{
        if(tmpdd<ep[0]){ // failed to increase Q -> exit
          convs = 1; convm = 1;
          F77_CALL(dcopy)(nb,alpham0,&incx,alpham,&incy);
          q_alpha = q_alpha0;
        }
        else s *= SSIZE;
      }
    } // M-step: while(convs==0)

// M-step: alpha: renew H
    if(convm==0){
      for(u=0; u<na[0]; u++){
        alphad0[u] = alphad[u];
        alphad[u]  = 0.0;
        for(i=0; i<ni[0]; i++){
          for(l=0; l<nl[0]; l++){
            tmpd = 0.0;
            for(j=0; j<nj[0]; j++){
              for(k=0; k<nk[j]; k++){
                jlk = k + nk[j]*l + nnk[j];
                tmpd += z[na[0]*jlk+u]*((x[i + ni[0]*j]==k+1) - ppi[jlk]);
              }
            }
            alphad[u] += tmpd * hlx[l + nl[0]*i];
          }
```

```
        }
        d_alpha[u]  = alpham[u] - alpham0[u];
        d_alphad[u] = alphad[u] - alphad0[u];
      }

      tmpd = F77_CALL(ddot)(na,d_alpha,&incx,d_alphad,&incy);
        for(u=0; u<na[0]; u++){
          for(v=0; v<na[0]; v++){
            uv = u + na[0]*v;
            alphaht[uv] = 0.0;
            for(k=0; k<na[0]; k++) alphaht[uv] +=
              ((u==k)*1.0-d_alpha[u]*d_alphad[k]/tmpd)*alphah[k+na[0]*v];
          }
        }
        for(u=0; u<na[0]; u++){
          for(v=u; v<na[0]; v++){
            uv = u + na[0]*v;
            alphah[uv] = d_alpha[u]*d_alpha[v]/tmpd;
            for(k=0; k<na[0]; k++) alphah[uv] +=
              alphaht[u+na[0]*k]*((k==v)*1.0-d_alphad[k]*d_alpha[v]/tmpd);
            alphah[v + na[0]*u] = alphah[uv];
          }
        }
        F77_CALL(dcopy)(na,alphad,&incx,alphad0,&incy);
    } // M-step: if(convm==0)
  } // M-step: if(convm==0)

  if(tm>=maxiterm[0]-1) convm = 1;
  tm += 1;
  if(convm==1) F77_CALL(dcopy)(na,alpham,&incx,alpha,&incy);
} // M-step: alpha: END: while(convm==0)

// renew the log-likelihood & hlx
  *ll = 0.0;
  for(l=0; l<nl[0]; l++) shlx[l] = 0.0;
  for(i=0; i<ni[0]; i++){ // hlx
    tmpd = 0.0;
    for(l=0; l<nl[0]; l++){
      tmpdd = lomega[l];
      for(j=0; j<nj[0]; j++) tmpdd
        += lppi[ (x[i+ni[0]*j]-1) + nk[j]*l + nnk[j] ];
      hlx[l + nl[0]*i] = exp(tmpdd);
      tmpd += hlx[l + nl[0]*i];
    }
    for(l=0; l<nl[0]; l++){
      hlx[l + nl[0]*i] /= tmpd;
      shlx[l] += hlx[l + nl[0]*i];
    }
    *ll += log(tmpd);
  }
  Rprintf("Iteration %d: Log-Likelihood = %f\n",t+1,*ll);

// EM: convergence check
  for(u=0; u<nb[0]; u++) d_beta[u] = beta[u] - beta0[u];
  tmpd = fabs(d_beta[F77_CALL(idamax)(nb,d_beta,&incx)-1]);
  for(u=0; u<na[0]; u++) d_alpha[u] = alpha[u] - alpha0[u];
```

```
  tmpdd = fabs(d_alpha[F77_CALL(idamax)(na,d_alpha,&incx)-1]);
  if((tmpd<ep[0] && tmpdd<ep[0]) || fabs(*ll-ll0)<el[0]) *conv = 1;
  else ll0 = *ll;

  t += 1;
  if(*conv==0 && t>=maxiter[0])
    Rprintf("WARNING: EM reached the maximum number of
      iterations without convergence\n");
} /* EM algorithm END: while(conv==0) */

  if(*conv==1) Rprintf("\n***** EM algorithm converged *****\n");
  Rprintf("Estimates:\n");
  Rprintf("beta:\n");
  for(u=0; u<nb[0]; u++) Rprintf("%10.6f\n",beta[u]);
  Rprintf("alpha:\n");
  for(u=0; u<na[0]; u++) Rprintf("%10.6f\n",alpha[u]);
  Rprintf("\nLog-Likelihood = %f\n",*ll);
  Rprintf("Number of parameters = %d\n",na[0]+nb[0]);
  Rprintf("Number of EM iterations = %d\n",t);

/* CLEANING UP */

  free(alpha0); free(alpham); free(alpham0);
  free(alphad); free(alphad0);
  free(alphah); free(alphaht);
  free(d_alpha); free(d_alphad);
  free(beta0); free(betam); free(betam0);
  free(betad); free(betad0);
  free(betah); free(betaht);
  free(d_beta); free(d_betad);
  free(omega); free(lomega); free(ppi); free(lppi);
  free(shlx); free(nnk);

} // lllcaNP

/* LLLCA with fixed parameter constraints */
void lllcaNPC(int *x, int *ni, int *nj, int *nk, int *nl, int *na,
int *nb, int *maxiter, int *maxiterm, double *ep, double *ed, double *el,
double *y, double *z, double *alpha, int *alphaf, double *beta,
double *hlx, double *ll, int *conv)
{
  int i,j,k,l,jlk,u,v,uv,t=0,tm=0,convm=0,convs=0,nz;
  int incx=1,incy=1;
  int *nnk;
  char uplo='u', trst='t';
  double ll0, tmpd=0.0, tmpdd=0.0, a=1.0, b=0.0, s, ss=SSIZE;
  double q_alpha=0.0, q_beta=0.0, q_alpha0=0.0, q_beta0=0.0;
  double *shlx, *omega, *lomega, *ppi, *lppi;
  double *alpha0,*alpham,*alpham0;
  double *alphad,*alphad0,*alphah,*alphaht,*d_alpha,*d_alphad;
  double *beta0,*betam,*betam0;
  double *betad,*betad0,*betah,*betaht,*d_beta,*d_betad;

  /* INITIALIZATION */

  nnk = (int *) malloc(nj[0] * sizeof(int));
  nnk[0] = 0; nz = nk[0];
```

```
  for(j=1; j<nj[0]; j++){
    nnk[j] = nnk[j-1]+nk[j-1]*nl[0];
    nz += nk[j];
  }
  nz *= nl[0];

  alpha0   = (double *) malloc(na[0] * sizeof(double));
  alpham   = (double *) malloc(na[0] * sizeof(double));
  alpham0  = (double *) malloc(na[0] * sizeof(double));
  alphad   = (double *) malloc(na[0] * sizeof(double));
  alphad0  = (double *) malloc(na[0] * sizeof(double));
  alphah   = (double *) malloc(na[0]*na[0] * sizeof(double));
  alphaht  = (double *) malloc(na[0]*na[0] * sizeof(double));
  d_alpha  = (double *) malloc(na[0] * sizeof(double));
  d_alphad = (double *) malloc(na[0] * sizeof(double));
  beta0    = (double *) malloc(nb[0] * sizeof(double));
  betam    = (double *) malloc(nb[0] * sizeof(double));
  betam0   = (double *) malloc(nb[0] * sizeof(double));
  betad    = (double *) malloc(nb[0] * sizeof(double));
  betad0   = (double *) malloc(nb[0] * sizeof(double));
  betah    = (double *) malloc(nb[0]*nb[0] * sizeof(double));
  betaht   = (double *) malloc(nb[0]*nb[0] * sizeof(double));
  d_beta   = (double *) malloc(nb[0] * sizeof(double));
  d_betad  = (double *) malloc(nb[0] * sizeof(double));
  omega    = (double *) malloc(nl[0] * sizeof(double));
  lomega   = (double *) malloc(nl[0] * sizeof(double));
  ppi      = (double *) malloc(nz * sizeof(double));
  lppi     = (double *) malloc(nz * sizeof(double));
  shlx     = (double *) malloc(nl[0] * sizeof(double));

  for(u=0; u<nb[0]; u++){
    beta0[u] = beta[u]; betad0[u] = beta[u];
  }
  for(u=0; u<na[0]; u++){
    alpha0[u] = alpha[u]; alphad0[u] = alpha[u];
  }
  ll0 = 0.0;
  for(j=0; j<nj[0]; j++) ll0 += log(1.0/nk[j]);
  ll0 *= ni[0]; *ll = ll0;

  Rprintf("--------------------------------------------\n");
  Rprintf("EM Algorithm for LLLCM (No Person Covariates)\n");
  Rprintf("Initial Values:\n");
  Rprintf("beta:\n");
  for(u=0; u<nb[0]; u++) Rprintf("%10.6f\n",beta[u]);
  Rprintf("alpha:\n");
  for(u=0; u<na[0]; u++) Rprintf("%10.6f %d \n",alpha[u],alphaf[u]);
  Rprintf("Initial Log-Likelihood = %f\n",ll0);
  Rprintf("--------------------------------------------\n");

/* EM algorithm */

*conv = 0;
while(*conv==0 && t<maxiter[0]){

  F77_CALL(dcopy)(nb,beta,&incx,beta0,&incy);
```

```
  F77_CALL(dcopy)(na,alpha,&incx,alpha0,&incy);

// E-step
  tmpd = 0.0;
  F77_CALL(dgemv)(&trst,nb,nl,&a,y,nb,beta,&incx,&b,lomega,&incy);
  for(l=0; l<nl[0]; l++){
    shlx[l] = 0.0;
    for(i=0; i<ni[0]; i++) shlx[l] += hlx[l + nl[0]*i];
    omega[l] = exp(lomega[l]);
    tmpd += omega[l];
  }
  q_beta0 = 0.0;
  for(l=0; l<nl[0]; l++){
    omega[l] /= tmpd;
    lomega[l] -= log(tmpd);
    q_beta0 += shlx[l]*lomega[l];
  }

  F77_CALL(dgemv)(&trst,na,&nz,&a,z,na,alpha,&incx,&b,lppi,&incy);
  for(j=0; j<nj[0]; j++){
    for(l=0; l<nl[0]; l++){
      tmpd = 0.0;
      for(k=0; k<nk[j]; k++){
        jlk = k + nk[j]*l + nnk[j];
        ppi[jlk] = exp(lppi[jlk]);
        tmpd += ppi[jlk];
      }
      for(k=0; k<nk[j]; k++){
        jlk = k + nk[j]*l + nnk[j];
        ppi[jlk] /= tmpd;
        lppi[jlk] -= log(tmpd);
      }
    }
  }
  q_alpha0 = 0.0;
  for(i=0; i<ni[0]; i++)
    for(l=0; l<nl[0]; l++)
      for(j=0; j<nj[0]; j++)
        q_alpha0 +=
          hlx[l + nl[0]*i]*lppi[ (x[i+ni[0]*j]-1) + nk[j]*l + nnk[j] ];

// M-step: beta
  for(u=0; u<nb[0]; u++)
    for(v=0; v<nb[0]; v++) betah[u + nb[0]*v] = (u==v)*1.0;

  F77_CALL(dcopy)(nb,beta,&incx,betam,&incy);
  for(u=0; u<nb[0]; u++){
    betad[u] = 0.0;
    for(l=0; l<nl[0]; l++) betad[u] +=
      y[nb[0]*l + u]*(shlx[l] - ni[0]*omega[l]);
  }
  tm = 0; convm = 0;

while(convm==0){
  F77_CALL(dcopy)(nb,betam,&incx,betam0,&incy);
```

```
// M-step: beta: convergence check
  tmpd = fabs(betad[F77_CALL(idamax)(nb,betad,&incx)-1]);
  if(tmpd<ed[0]){
    convm = 1;
    F77_CALL(dcopy)(nb,betam,&incx,beta,&incy);
  }

// M-step: beta: optimization by BFGS
  if(convm==0){
    convs = 0;
    if(tmpd>1.0/ss) s = 1.0 / tmpd; else s = ss;

    while(convs==0){
      F77_CALL(dsymv)(&uplo,nb,&s,betah,nb,betad,&incx,&b,betam,&incy);
      tmpdd = fabs(betam[F77_CALL(idamax)(nb,betam,&incx)-1]);
      F77_CALL(daxpy)(nb,&a,betam0,&incx,betam,&incy);

// renew the Q function
      tmpd = 0.0;
      F77_CALL(dgemv)(&trst,nb,nl,&a,y,nb,betam,&incx,&b,lomega,&incy);
      for(l=0; l<nl[0]; l++){ // omega
        omega[l] = exp(lomega[l]);
        tmpd += omega[l];
      }
      q_beta = 0.0;
      for(l=0; l<nl[0]; l++){
        omega[l] /= tmpd;
        lomega[l] -= log(tmpd);
        q_beta += shlx[l]*lomega[l];
      }

      if(q_beta>=q_beta0){
        convs = 1;
        q_beta0 = q_beta;
      }
      else{
        if(tmpdd<ep[0]){ // failed to increase Q -> exit
          convs = 1; convm = 1;
          F77_CALL(dcopy)(nb,betam0,&incx,betam,&incy);
          q_beta = q_beta0;
        }
        else s *= SSIZE;
      }
    } // M-step: while(convs==0)

// renew H
    if(convm==0){
      for(u=0; u<nb[0]; u++){
        betad0[u] = betad[u];
        betad[u]  = 0.0;
        for(l=0; l<nl[0]; l++) betad[u] +=
          y[nb[0]*l + u]*(shlx[l] - ni[0]*omega[l]);
        d_beta[u]  = betam[u] - betam0[u];
        d_betad[u] = betad[u] - betad0[u];
      }

      tmpd = F77_CALL(ddot)(nb,d_beta,&incx,d_betad,&incy);
```

```
        for(u=0; u<nb[0]; u++){
          for(v=0; v<nb[0]; v++){
            uv = u + nb[0]*v;
            betaht[uv] = 0.0;
            for(k=0; k<nb[0]; k++) betaht[uv] +=
              ((u==k)*1.0 - d_beta[u]*d_betad[k]/tmpd)*betah[k + nb[0]*v];
          }
        }
        for(u=0; u<nb[0]; u++){
          for(v=u; v<nb[0]; v++){
            uv = u + nb[0]*v;
            betah[uv] = d_beta[u]*d_beta[v]/tmpd;
            for(k=0; k<nb[0]; k++) betah[uv] +=
              betaht[u+nb[0]*k]*((k==v)*1.0 - d_betad[k]*d_beta[v]/tmpd);
            betah[v + nb[0]*u] = betah[uv];
          }
        }
        F77_CALL(dcopy)(nb,betad,&incx,betad0,&incy);
    } // M-step: if(convm==0)
  } // M-step: if(convm==0)

  if(tm>=maxiterm[0]-1) convm = 1;
  tm += 1;
  if(convm==1) F77_CALL(dcopy)(nb,betam,&incx,beta,&incy);
} // M-step: beta: END: while(convm==0)

// M-step: alpha
  for(u=0; u<na[0]; u++)
    for(v=0; v<na[0]; v++) alphah[u + na[0]*v] = (u==v)*1.0;

  F77_CALL(dcopy)(na,alpha,&incx,alpham,&incy);
  for(u=0; u<na[0]; u++){
    alphad[u] = 0.0;
    if(alphaf[u]==0){ // only for unconstrained alphas
      for(i=0; i<ni[0]; i++){
        for(l=0; l<nl[0]; l++){
          tmpd = 0.0;
          for(j=0; j<nj[0]; j++){
            for(k=0; k<nk[j]; k++){
              jlk = k + nk[j]*l + nnk[j];
              tmpd += z[na[0]*jlk + u]*((x[i + ni[0]*j]==k+1) - ppi[jlk]);
            }
          }
          alphad[u] += tmpd * hlx[l + nl[0]*i];
        }
      }
    }
  }
  tm = 0; convm = 0;

while(convm==0){
  F77_CALL(dcopy)(na,alpham,&incx,alpham0,&incy);

// M-step: alpha: convergence check
  tmpd = fabs(alphad[F77_CALL(idamax)(na,alphad,&incx)-1]);
  if(tmpd<ed[0]){
    convm = 1;
```

```
    F77_CALL(dcopy)(na,alpham,&incx,alpha,&incy);
  }

// M-step: alpha: optimization by BFGS
  if(convm==0){
    convs = 0;
    if(tmpd>1.0/ss) s = 1.0 / tmpd; else s = ss;

    while(convs==0){
      F77_CALL(dsymv)(&uplo,na,&s,alphah,na,alphad,&incx,&b,alpham,&incy);
      tmpdd = fabs(alpham[F77_CALL(idamax)(na,alpham,&incx)-1]);
      F77_CALL(daxpy)(na,&a,alpham0,&incx,alpham,&incy);

// M-step: alpha: renew the Q functions
      F77_CALL(dgemv)(&trst,na,&nz,&a,z,na,alpham,&incx,&b,lppi,&incy);
      for(j=0; j<nj[0]; j++){ // pi
        for(l=0; l<nl[0]; l++){
          tmpd = 0.0;
          for(k=0; k<nk[j]; k++){
            jlk = k + nk[j]*l + nnk[j];
            ppi[jlk] = exp(lppi[jlk]);
            tmpd += ppi[jlk];
          }
          for(k=0; k<nk[j]; k++){
            jlk = k + nk[j]*l + nnk[j];
            ppi[jlk] /= tmpd;
            lppi[jlk] -= log(tmpd);
          }
        }
      }
      q_alpha = 0.0;
      for(i=0; i<ni[0]; i++)
        for(l=0; l<nl[0]; l++)
          for(j=0; j<nj[0]; j++)
            q_alpha +=
              hlx[l+nl[0]*i]*lppi[ (x[i+ni[0]*j]-1) + nk[j]*l + nnk[j] ];

      if(q_alpha>=q_alpha0){
        convs = 1;
        q_alpha0 = q_alpha;
      }
      else{
        if(tmpdd<ep[0]){ // failed to increase Q -> exit
          convs = 1; convm = 1;
          F77_CALL(dcopy)(nb,alpham0,&incx,alpham,&incy);
          q_alpha = q_alpha0;
        }
        else s *= SSIZE;
      }
    } // M-step: while(convs==0)

// M-step: alpha: renew H
    if(convm==0){
      for(u=0; u<na[0]; u++){
        alphad0[u] = alphad[u];
        alphad[u]  = 0.0;
        if(alphaf[u]==0){ // only for unconstrained alphas
```

```
          for(i=0; i<ni[0]; i++){
            for(l=0; l<nl[0]; l++){
              tmpd = 0.0;
              for(j=0; j<nj[0]; j++){
                for(k=0; k<nk[j]; k++){
                  jlk = k + nk[j]*l + nnk[j];
                  tmpd += z[na[0]*jlk+u]*((x[i+ni[0]*j]==k+1) - ppi[jlk]);
                }
              }
              alphad[u] += tmpd * hlx[l + nl[0]*i];
            }
          }
        }
        d_alpha[u]  = alpham[u] - alpham0[u];
        d_alphad[u] = alphad[u] - alphad0[u];
      }

      tmpd = F77_CALL(ddot)(na,d_alpha,&incx,d_alphad,&incy);
        for(u=0; u<na[0]; u++){
          for(v=0; v<na[0]; v++){
            uv = u + na[0]*v;
            alphaht[uv] = 0.0;
            for(k=0; k<na[0]; k++) alphaht[uv] +=
              ((u==k)*1.0-d_alpha[u]*d_alphad[k]/tmpd)*alphah[k+na[0]*v];
          }
        }
        for(u=0; u<na[0]; u++){
          for(v=u; v<na[0]; v++){
            uv = u + na[0]*v;
            alphah[uv] = d_alpha[u]*d_alpha[v]/tmpd;
            for(k=0; k<na[0]; k++) alphah[uv] +=
              alphaht[u+na[0]*k]*((k==v)*1.0-d_alphad[k]*d_alpha[v]/tmpd);
            alphah[v + na[0]*u] = alphah[uv];
          }
        }
        F77_CALL(dcopy)(na,alphad,&incx,alphad0,&incy);

    } // M-step: if(convm==0)
  } // M-step: if(convm==0)

  if(tm>=maxiterm[0]-1) convm = 1;
  tm += 1;
  if(convm==1) F77_CALL(dcopy)(na,alpham,&incx,alpha,&incy);
} // M-step: alpha: END: while(convm==0)

// renew the log-likelihood & hlx
  *ll = 0.0;
  for(l=0; l<nl[0]; l++) shlx[l] = 0.0;
  for(i=0; i<ni[0]; i++){ // hlx
    tmpd = 0.0;
    for(l=0; l<nl[0]; l++){
      tmpdd = lomega[l];
      for(j=0; j<nj[0]; j++) tmpdd +=
        lppi[ (x[i+ni[0]*j]-1) + nk[j]*l + nnk[j] ];
      hlx[l + nl[0]*i] = exp(tmpdd);
      tmpd += hlx[l + nl[0]*i];
```

```
    }
    for(l=0; l<nl[0]; l++){
      hlx[l + nl[0]*i] /= tmpd;
      shlx[l] += hlx[l + nl[0]*i];
    }
    *ll += log(tmpd);
  }
  Rprintf("Iteration %d: Log-Likelihood = %f\n",t+1,*ll);

// EM: convergence check
  for(u=0; u<nb[0]; u++) d_beta[u] = beta[u] - beta0[u];
  tmpd = fabs(d_beta[F77_CALL(idamax)(nb,d_beta,&incx)-1]);
  for(u=0; u<na[0]; u++) d_alpha[u] = alpha[u] - alpha0[u];
  tmpdd = fabs(d_alpha[F77_CALL(idamax)(na,d_alpha,&incx)-1]);
  if((tmpd<ep[0] && tmpdd<ep[0]) || fabs(*ll-ll0)<el[0]) *conv = 1;
  else ll0 = *ll;

  t += 1;
  if(*conv==0 && t>=maxiter[0])
    Rprintf("WARNING: EM reached the maximum number of iterations
      without convergence\n");
} /* EM algorithm END: while(conv==0) */

  nz = 0;
  for(u=0; u<na[0]; u++) nz += alphaf[u];
  if(*conv==1) Rprintf("\n***** EM algorithm converged *****\n");
  Rprintf("Estimates:\n");
  Rprintf("beta:\n");
  for(u=0; u<nb[0]; u++) Rprintf("%10.6f ",beta[u]);
  Rprintf("\nalpha:\n");
  for(u=0; u<na[0]; u++) Rprintf("%10.6f ",alpha[u]);
  Rprintf("\nLog-Likelihood = %f\n",*ll);
  Rprintf("Number of parameters = %d\n",na[0]-nz+nb[0]);
  Rprintf("Number of EM iterations = %d\n",t);

/* CLEANING UP */

  free(alpha0); free(alpham); free(alpham0);
  free(alphad); free(alphad0);
  free(alphah); free(alphaht);
  free(d_alpha); free(d_alphad);
  free(beta0); free(betam); free(betam0);
  free(betad); free(betad0);
  free(betah); free(betaht);
  free(d_beta); free(d_betad);
  free(omega); free(lomega); free(ppi); free(lppi);
  free(shlx); free(nnk);

} // lllcaNPC
```

Appendix C

Estimates of Item Response Probabilities in Study 2

This appendix shows tables of item response probability estimates from which Figure 5.2 was drawn. In the following, each table shows a set of item response probability estimates ($\hat{\boldsymbol{\pi}}$) for a class (rule) and contains those estimates for both multiple choice and dichotomous models.

Table C.1: Estimates of Item Response Probabilities for Random

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
2	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
3	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
4	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
5	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
6	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
7	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
8	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
9	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
10	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
11	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
12	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
13	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
14	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
15	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
16	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
17	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
18	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
19	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
20	B	–	0.3333	0.3333	0.3333	0.6667	0.3333

Note. L = Left; R = Right; B = Balance. No particular response is predicted by this rule. All probability values were fixed and no parameters were estimated in this class.

Table C.2: Estimates of Item Response Probabilities for Rule 1

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	B	0.0002	0.9996	0.0002	0.9405	0.0595
2	L	B	0.0003	0.9993	0.0003	0.9340	0.0660
3	L	B	0.0122	0.9757	0.0122	0.9661	0.0339
4	L	B	0.0105	0.9790	0.0105	0.9518	0.0482
5	L	B	0.0003	0.9995	0.0003	0.9690	0.0310
6	L	L	1.0000	0.0000	0.0000	0.0000	1.0000
7	L	L	0.9895	0.0052	0.0052	0.0169	0.9831
8	L	L	1.0000	0.0000	0.0000	0.0000	1.0000
9	L	L	1.0000	0.0000	0.0000	0.0000	1.0000
10	L	L	0.9893	0.0053	0.0053	0.0088	0.9912
11	L	R	0.0065	0.0065	0.9870	0.9825	0.0175
12	L	R	0.0064	0.0064	0.9871	1.0000	0.0000
13	L	R	0.0000	0.0000	1.0000	1.0000	0.0000
14	L	R	0.0049	0.0049	0.9902	0.9916	0.0084
15	L	R	0.0050	0.0050	0.9899	0.9916	0.0084
16	B	L	0.9897	0.0051	0.0051	1.0000	0.0000
17	B	L	0.9895	0.0053	0.0053	0.9927	0.0073
18	B	L	0.9784	0.0108	0.0108	0.9915	0.0085
19	B	L	0.9998	0.0001	0.0001	1.0000	0.0000
20	B	L	1.0000	0.0000	0.0000	1.0000	0.0000

Note. L = Left; R = Right; B = Balance.

Table C.3: Estimates of Item Response Probabilities for Rule 2

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	L	0.8501	0.0749	0.0749	0.0004	0.9996
2	L	L	0.8710	0.0645	0.0645	0.0653	0.9347
3	L	L	0.9631	0.0185	0.0185	0.0001	0.9999
4	L	L	0.9461	0.0269	0.0269	0.0210	0.9790
5	L	L	0.9525	0.0238	0.0238	0.0189	0.9811
6	L	L	0.9801	0.0099	0.0099	0.0336	0.9664
7	L	L	0.8730	0.0635	0.0635	0.0501	0.9499
8	L	L	0.8639	0.0680	0.0680	0.0209	0.9791
9	L	L	1.0000	0.0000	0.0000	0.0000	1.0000
10	L	L	0.9741	0.0130	0.0130	0.0000	1.0000
11	L	R	0.0699	0.0699	0.8602	0.9887	0.0113
12	L	R	0.1809	0.1809	0.6381	0.9307	0.0693
13	L	R	0.0978	0.0978	0.8045	0.9677	0.0323
14	L	R	0.2626	0.2626	0.4748	0.8900	0.1100
15	L	R	0.2691	0.2691	0.4618	0.9001	0.0999
16	B	L	0.9698	0.0151	0.0151	0.9810	0.0190
17	B	L	0.7392	0.1304	0.1304	0.7548	0.2452
18	B	L	0.7743	0.1128	0.1128	0.7538	0.2462
19	B	L	0.9685	0.0158	0.0158	0.9999	0.0001
20	B	L	0.5967	0.2016	0.2016	0.7328	0.2672

Note. L = Left; R = Right; B = Balance.

Table C.4: Estimates of Item Response Probabilities for Rule 3

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	L	0.9996	0.0002	0.0002	0.0004	0.9996
2	L	L	0.9986	0.0007	0.0007	0.0981	0.9019
3	L	L	0.8613	0.0694	0.0694	0.0719	0.9281
4	L	L	0.7902	0.1049	0.1049	0.2316	0.7684
5	L	L	0.9077	0.0462	0.0462	0.1362	0.8638
6	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
7	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
8	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
9	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
10	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
11	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
12	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
13	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
14	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
15	L	–	0.3333	0.3333	0.3333	0.6667	0.3333
16	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
17	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
18	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
19	B	–	0.3333	0.3333	0.3333	0.6667	0.3333
20	B	–	0.3333	0.3333	0.3333	0.6667	0.3333

Note. L = Left; R = Right; B = Balance. No particular response is predicted for items 6 through 20 by this rule. All probability values were fixed and no parameters were estimated for these items.

Table C.5: Estimates of Item Response Probabilities for Rule 4

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	L	0.9999	0.0000	0.0000	0.0001	0.9999
2	L	L	0.9999	0.0000	0.0000	0.0001	0.9999
3	L	L	0.9999	0.0000	0.0000	0.0001	0.9999
4	L	L	0.9679	0.0160	0.0160	0.0001	0.9999
5	L	L	0.9999	0.0000	0.0000	0.0001	0.9999
6	L	L	0.9656	0.0172	0.0172	0.0001	0.9999
7	L	L	0.9998	0.0001	0.0001	0.0303	0.9697
8	L	L	0.8211	0.0894	0.0894	0.1288	0.8712
9	L	L	0.9999	0.0000	0.0000	0.0001	0.9999
10	L	L	0.9634	0.0183	0.0183	0.0328	0.9672
11	L	L	0.8539	0.0731	0.0731	0.2200	0.7800
12	L	L	0.9999	0.0001	0.0001	0.1145	0.8855
13	L	L	0.9450	0.0275	0.0275	0.0653	0.9347
14	L	L	0.9998	0.0001	0.0001	0.0001	0.9999
15	L	L	0.9999	0.0000	0.0000	0.0001	0.9999
16	B	B	0.0804	0.8392	0.0804	0.2401	0.7599
17	B	B	0.0197	0.9606	0.0197	0.0334	0.9666
18	B	B	0.0176	0.9648	0.0176	0.0001	0.9999
19	B	B	0.0444	0.9111	0.0444	0.2162	0.7838
20	B	B	0.0560	0.8880	0.0560	0.0993	0.9007

Note. L = Left; R = Right; B = Balance.

Table C.6: Estimates of Item Response Probabilities for Addition

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	L	0.9372	0.0314	0.0314	0.0959	0.9041
2	L	L	0.9426	0.0287	0.0287	0.0481	0.9519
3	L	L	1.0000	0.0000	0.0000	0.0001	0.9999
4	L	L	0.9428	0.0286	0.0286	0.0293	0.9707
5	L	L	0.9592	0.0204	0.0204	0.0164	0.9836
6	L	L	0.7852	0.1074	0.1074	0.1508	0.8492
7	L	B	0.2227	0.5546	0.2227	0.6171	0.3829
8	L	B	0.1580	0.6839	0.1580	0.8613	0.1387
9	L	L	0.9614	0.0193	0.0193	0.0000	1.0000
10	L	L	0.8694	0.0653	0.0653	0.0797	0.9203
11	L	L	0.1969	0.4015	0.4015	0.8530	0.1470
12	L	R	0.3587	0.3587	0.2827	0.6996	0.3004
13	L	R	0.2558	0.2558	0.4883	0.9997	0.0003
14	L	L	0.4540	0.2730	0.2730	0.6857	0.3143
15	L	L	0.5072	0.2464	0.2464	0.5990	0.4010
16	B	L	0.8778	0.0611	0.0611	0.9275	0.0725
17	B	B	0.1794	0.6411	0.1794	0.3934	0.6066
18	B	B	0.1786	0.6427	0.1786	0.3173	0.6827
19	B	L	0.6734	0.1633	0.1633	0.7059	0.2941
20	B	R	0.3163	0.3163	0.3674	0.4635	0.5365

Note. L = Left; R = Right; B = Balance.

Table C.7: Estimates of Item Response Probabilities for QP

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	L	0.9712	0.0144	0.0144	0.0290	0.9710
2	L	L	1.0000	0.0000	0.0000	0.0386	0.9614
3	L	L	1.0000	0.0000	0.0000	0.0000	1.0000
4	L	L	0.9853	0.0073	0.0073	0.0269	0.9731
5	L	L	0.9854	0.0073	0.0073	0.0130	0.9870
6	L	B	0.1762	0.6477	0.1762	0.6954	0.3046
7	L	B	0.0410	0.9181	0.0410	0.9176	0.0824
8	L	B	0.0420	0.9160	0.0420	0.8727	0.1273
9	L	L	0.9409	0.0295	0.0295	0.1228	0.8772
10	L	L	0.8037	0.0982	0.0982	0.2582	0.7418
11	L	L	0.4826	0.2587	0.2587	0.5805	0.4195
12	L	L	0.7751	0.1124	0.1124	0.2400	0.7600
13	L	L	0.9339	0.0330	0.0330	0.0914	0.9086
14	L	L	0.8936	0.0532	0.0532	0.1698	0.8302
15	L	L	0.9090	0.0455	0.0455	0.1369	0.8631
16	B	B	0.2561	0.4878	0.2561	0.6149	0.3851
17	B	B	0.0640	0.8720	0.0640	0.2935	0.7065
18	B	B	0.0626	0.8748	0.0626	0.2983	0.7017
19	B	L	0.6582	0.1709	0.1709	0.6502	0.3498
20	B	R	0.0264	0.0264	0.9471	0.9066	0.0934

Note. L = Left; R = Right; B = Balance.

Table C.8: Estimates of Item Response Probabilities for SDD

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	R	0.4819	0.4819	0.0361	0.4592	0.5408
2	L	R	0.4451	0.4451	0.1097	0.3259	0.6741
3	L	R	0.4395	0.4395	0.1209	0.1860	0.8140
4	L	R	0.4085	0.4085	0.1829	0.3344	0.6656
5	L	R	0.4221	0.4221	0.1557	0.2882	0.7118
6	L	L	1.0000	0.0000	0.0000	0.0000	1.0000
7	L	L	0.9470	0.0265	0.0265	0.1640	0.8360
8	L	L	0.9825	0.0088	0.0088	0.0874	0.9126
9	L	L	1.0000	0.0000	0.0000	0.0000	1.0000
10	L	L	0.9768	0.0116	0.0116	0.0599	0.9401
11	L	R	0.1049	0.1049	0.7901	0.8816	0.1184
12	L	R	0.1045	0.1045	0.7909	0.8114	0.1886
13	L	R	0.0147	0.0147	0.9705	0.9603	0.0397
14	L	R	0.0706	0.0706	0.8589	0.8255	0.1745
15	L	R	0.0561	0.0561	0.8877	0.8147	0.1853
16	B	L	0.9154	0.0423	0.0423	0.9759	0.0241
17	B	L	0.9586	0.0207	0.0207	0.9590	0.0410
18	B	L	1.0000	0.0000	0.0000	0.9773	0.0227
19	B	L	0.9040	0.0480	0.0480	0.9229	0.0771
20	B	L	0.9532	0.0234	0.0234	0.9153	0.0847

Note. L = Left; R = Right; B = Balance.

Table C.9: Estimates of Item Response Probabilities for Buggy

	Correct	Expected	Multiple Choice Model			Dichotomous Model	
Item	Response	Response	Left	Balance	Right	Incorrect	Correct
1	L	L	0.8887	0.0557	0.0557	0.0511	0.9489
2	L	L	0.8911	0.0545	0.0545	0.0506	0.9494
3	L	L	0.9545	0.0227	0.0227	0.0103	0.9897
4	L	L	0.8950	0.0525	0.0525	0.0776	0.9224
5	L	L	0.9036	0.0482	0.0482	0.0664	0.9336
6	L	L	0.5376	0.2312	0.2312	0.1508	0.8492
7	L	B	0.4140	0.1719	0.4140	0.1651	0.8349
8	L	B	0.4935	0.0130	0.4935	0.3611	0.6389
9	L	L	0.8258	0.0871	0.0871	0.0170	0.9830
10	L	L	0.5564	0.2218	0.2218	0.1815	0.8185
11	L	L	0.3540	0.3230	0.3230	0.7191	0.2809
12	L	L	0.8137	0.0931	0.0931	0.3210	0.6790
13	L	L	0.6701	0.1649	0.1649	0.6553	0.3447
14	L	L	0.9998	0.0001	0.0001	0.0270	0.9730
15	L	L	0.8756	0.0622	0.0622	0.0575	0.9425
16	B	B	0.4238	0.1525	0.4238	0.9065	0.0935
17	B	B	0.4746	0.0508	0.4746	0.7531	0.2469
18	B	B	0.4999	0.0002	0.4999	0.8747	0.1253
19	B	L	0.5033	0.2484	0.2484	0.8612	0.1388
20	B	R	0.0985	0.0985	0.8031	0.8770	0.1230

Note. L = Left; R = Right; B = Balance.