Applied Psychological Measurement
http://apm.sagepub.com/

Diagnosis of Subtraction Bugs Using Bayesian Networks

Jihyun Lee and James E. Corter

Applied Psychological Measurement 2011 35: 27 originally published online 7 October 2010
DOI: 10.1177/0146621610377079

Article

Applied Psychological Measurement
35(1) 27–47
© The Author(s) 2011
Reprints and permission:
sagepub.com/journalsPermissions.nav
DOI: 10.1177/0146621610377079
http://apm.sagepub.com

Diagnosis of Subtraction Bugs Using Bayesian Networks

Jihyun Lee[1] and James E. Corter[1]

Abstract

Diagnosis of misconceptions or “bugs” in procedural skills is difficult because of their unstable nature. This study addresses this problem by proposing and evaluating a probability-based approach to the diagnosis of bugs in children’s multicolumn subtraction performance using Bayesian networks. This approach assumes a causal network relating hypothesized subtraction bugs to the observed test items. Two research questions are tested within this framework. First, it is investigated whether more reliable assessment of latent subtraction bugs can be achieved by hypothesizing and using subskill nodes in the Bayesian network as causal factors affecting bugs. Second, network performance is evaluated using two types of testing situations, one using binary data (items scored as correct or incorrect) and the other simulating a multiple-choice test format with diagnostic use of specific wrong answers. The resulting four types of Bayesian networks are evaluated for their effectiveness in bug diagnosis. All four networks show good performance, with even the simplest network (bug nodes only, binary data) giving overall bug diagnosis rates of at least 85%. Prediction is best with the most complex network (bug and subskill nodes, diagnostic use of specific wrong answers), for which the correct diagnosis rate reaches 99%. These results suggest that stable and reliable bug diagnosis can be achieved using a Bayesian network framework, but that the stability and effectiveness of diagnosis is increased when the network includes latent subskills in addition to bugs as causal factors, and when specific wrong answers are used for diagnostic purposes.

Keywords

test theory, psychometrics, cognitive psychology, computerized testing, diagnostic testing, Bayesian networks, latent class models, misconceptions, subtraction bugs, procedural skills

Studies using large-scale data, such as the National Assessment of Educational Progress (NAEP), have consistently shown that U.S. children have trouble with subtraction operations. Particularly problematic is the operation of borrowing in multicolumn subtraction (Carpenter, Franke, Jacobs, Fennema, & Empson, 1998; Young & O’Shea, 1981). In one study, approximately 50% of U.S. fourth-graders and 15% of eighth-graders were not able to subtract a two-digit number from a three-digit number with borrowing (Kouba, Zawojewski, & Strutchens, 1997).

[1] Columbia University, New York, New York, USA

Corresponding Author:
James E. Corter, Teachers College, 525 W. 120th Street, New York, NY 10027, USA
Email: jec34@columbia.edu

One way to investigate why subtraction with borrowing is difficult for school children is through cognitively oriented error analysis. A common assumption in such research is that each individual has systematic and persistent error patterns because individuals follow faulty procedures that they have learned or invented (J. S. Brown & Burton, 1978; Cox, 1975; Sleeman, 1984). In practice, it is difficult to diagnose the existence of these ‘‘bugs’’ in an individual’s procedural skills. First of all, only a limited set of bugs (e.g., about a dozen bugs for subtraction skills) have been found frequently enough to confirm their existence (J. S. Brown & VanLehn, 1980). Second, whether a bug is exhibited can depend on specific problem contexts. Third, computational errors can occur when one is executing a buggy algorithm, just as they can when one is executing a correct algorithm, injecting noise into the process of bug diagnosis. For these reasons, bugs are unstable, meaning that a bug symptom observed in an individual might not show up in subsequent items or in a subsequent testing session. Consequently, some researchers have warned that reliable diagnosis of bugs based on students’ performance should not be expected (J. S. Brown & Burton, 1978; J. S. Brown & VanLehn, 1980).

The present study investigates whether reliable bug diagnosis can be achieved using a probabilistic approach to diagnosis that can accommodate the unstable nature of the bugs. Such a methodology is available in the form of Bayesian networks (e.g., Jensen, 1996; Mislevy, 1995; Pearl, 1988; Russell & Norvig, 1995) where uncertainties of an event occurrence are managed efficiently by use of conditional probabilities. Bayesian networks have been successfully applied to assess subskills and competencies in various domains, including physics problem solving (Martin & VanLehn, 1995; VanLehn & Martin, 1998), troubleshooting (Gitomer, Steinberg, & Mislevy, 1995), mixed-number subtraction (Mislevy, 1995), proportional reasoning (Beland & Mislevy, 1996), knowledge of English (Almond & Mislevy, 1999), dental hygiene (Mislevy, Steinberg, Breyer, & Almond, 2002), and student knowledge in computer-based instruction systems (e.g., Conati, Gertner, & VanLehn, 2002; Pardos, Heffernan, Anderson, & Heffernan, 2007; VanLehn & Niu, 2001). In recent years, interest in applying Bayesian networks in educational assessments seems to be growing (e.g., Desmarais & Pu, 2005; Heiner, Heffernan, & Barnes, 2007).

The framework of Bayesian networks seems appropriate to the task of modeling relationships between bugs, subskills, and test items for several reasons. First, causal relationships between bugs and test item performance can be modeled and estimated. Second, the instability of the bugs is expressed in a natural way by the probabilistic framework of a Bayesian network. Third, the varying degrees of strength in the relationships between specific bugs and the items can be expressed by the relevant conditional probabilities. Finally, a priori information about the frequency of various bugs can be used via the prior probabilities of bug incidence in a reference group.

Accordingly, the present study examines the capability of Bayesian networks to diagnose subtraction bugs. It considers and simulates two common types of diagnostic testing scenarios: diagnosis based on binary test data (scoring only item correct or incorrect) and diagnosis based on specific wrong answers, such as might be obtained with open-ended or constructed-response test items, or with multiple-choice test items with distractors specifically constructed to yield cognitively diagnostic information (cf. de la Torre, 2009; Lee & Corter, 2003; Sadler, 1998). It examines whether limited test information on student performance (i.e., correct–incorrect test scoring) is sufficient to diagnose subtraction bugs, or whether more detailed information (in the form of diagnostic use of specific incorrect answers) is needed for reliable bug diagnosis.

Another theoretical question that the present study investigates involves the effects of including subskill variables in the networks to diagnose subtraction bugs. The common assumption in the literature has been that each specific bug arises as a result of the lack of a particular subskill (J. S. Brown & Burton, 1978; J. S. Brown & VanLehn, 1980). Thus, the study investigated whether bug diagnosis can be made more stable and reliable by incorporating a set of hypothesized subskills as causal factors affecting the bugs.

Including subskill nodes in the network connects the present investigation to the growing body of research on cognitive diagnostic testing models. Cognitive diagnostic models generally produce a set of probabilities of individuals' mastery or nonmastery of cognitive attributes or skills, given evidence consisting of an individual's vector of correct or incorrect item performance for a test (Henson & Douglas, 2005; Junker & Sijtsma, 2001). Some of the most widely studied cognitive diagnostic models that use multidimensional approaches include the *rule space model* (Tatsuoka, 1985), *multiple classification latent class models* (Maris, 1999), *multidimensional compensatory IRT models* (Adams, Wilson, & Wang, 1997; Reckase, 1997), the *multicomponent latent trait model* or MLTM (Embretson, 1991), and *componential IRT models* (Sijtsma & Verweij, 1999). More recently, the *general diagnostic model* or GDM (von Davier & Yamamoto, 2004) has been proposed to model attributes at the ordinal level and to incorporate different types of skill dependencies. *Higher order latent trait models* (e.g., de la Torre & Douglas, 2004; Templin, Henson, Templin, & Roussos, 2008) extend the *deterministic inputs, noisy and gate* (DINA) model (DiBello, Stout, & Roussos, 1995; Maris, 1995, 1999) by adding a high-dimensional attribute vector expressed in the joint distributions. In all these models, latent response variables are treated as indicators of the presence or absence of cognitive attributes and assumed to be related to the observed item performance of each individual (Junker & Sijtsma, 2001). The main use of these models is to estimate individuals' subskill mastery patterns (for those methods assuming a fixed Q-matrix) or to classify individuals into one or multiple latent classes in terms of their possession of subskills or attributes.

The three main objectives of the present article are to ask (a) whether accurate bug diagnosis can be achieved using a probabilistic approach (specifically, Bayesian networks), (b) what type of item performance information is sufficient for reliable bug diagnosis (item correctness only vs. diagnostic use of wrong answers), and (c) whether the simultaneous diagnosis of hypothesized subskills (implemented as subskill nodes in the Bayesian network) improves the reliability of bug diagnosis.

Overview of Bayesian Networks

Bayesian networks have been one of the most widely used tools for managing and assessing uncertainty of an event occurrence, with successful applications in artificial intelligence, computer science, decision science, and engineering. Bayesian networks, also called belief networks, causal networks, probabilistic networks, or knowledge maps, implement probabilistic, causal relationships among a set of variables based on a directed acyclic graph (DAG) representation. The underlying theory of Bayesian inference networks was advanced by Pearl (1988) and by Lauritzen and Spiegelhalter (1988), with additional contributions by Jensen, Lauritzen, and Olesen (1990) and Spiegelhalter, Dawid, Lauritzen, and Cowell (1993), among many others. Thorough surveys of research on Bayesian networks are available (e.g., Cowell, Dawid, Lauritzen, & Spiegelhalter, 1999; Jensen, 1996; Korb & Nicholson, 2004).

In a typical application of Bayesian networks, each arc or arrow in the graph represents a causal relationship between the two connected variables. For two discrete variables, the strength of this causal relationship is determined by a set of conditional probabilities, specifically the probabilities of the states of the effect variable, conditional on states of the cause (taking into account their relationships with other variables in the network). These weight parameters must be specified beforehand in order to create a Bayesian network that can be used for inference. The weights might be specified in any of several ways, for example using the observed frequencies of marginal and conditional probabilities in a data set, or derived from a theory, or using the

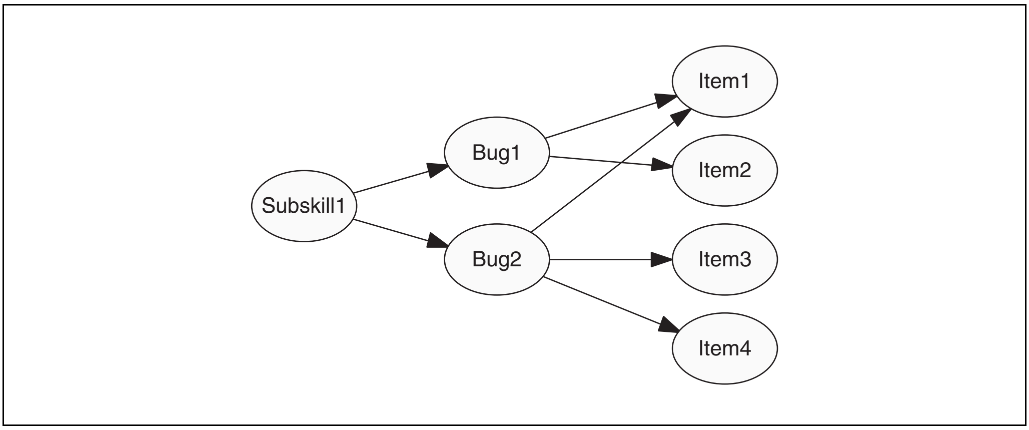

Figure 1. An example of a Bayesian network for bug diagnosis

subjective probability estimates of experts. A sample Bayesian network based on the problem of diagnosing subtraction bugs is shown in Figure 1.

The strengths of causal connections in the Bayesian network are indexed by marginal and conditional probabilities attached to the various nodes and arcs. For instance, consider just two of the nodes depicted in Figure 1, Bug 2 and test Item 4, and the depicted causal dependency between them. Assuming only two states for each node (0 and 1) and no missing values, the initial parameter values that need to be specified for this relationship are one simple or marginal probability and two conditional probabilities (the three probabilities of the complementary events are then determined), as follows:

a. P(Bug 2 = 1); P(Bug 2 = 0)
b. P(Item 4 = 1 | Bug 2 = 1); P(Item 4 = 0 | Bug 2 = 1)
c. P(Item 4 = 1 | Bug 2 = 0); P(Item 4 = 0 | Bug 2 = 0)

In general, up to $n2^k$ conditional probabilities must be specified for network inferences using any network topology, where n is the number of nodes, k is the maximum number of parent nodes in a network, and each node (variable) has only two possible values. Thus, for larger networks, learning algorithms may be useful, even critical, for filling in conditional probabilities that cannot be estimated from data or by experts.

Once the structure of a Bayesian network (BN) is specified and parameter values of the necessary causal association weights (conditional probabilities) are provided, the network can be used for inference or diagnosis. When a Bayesian network is used for diagnosis, evidence is entered by specifying the values of certain nodes (here corresponding to the test items), and the output consists of estimated posterior probabilities on the set of nodes of interest (the bugs or other latent variables). Formally, given a set of test items, A_m, which have been instantiated to values $A_i = a_{ik}$, the task is to compute the posterior probability distribution $P(L_j|A_i = a_{ik}, B, \theta)$ of a set of latent bug or subskill variables L_j, where a_{ik} is the kth state of the variable $A_{i,}$, B is the network structure (including both the network topology and the causal weights) and θ denotes the network parameters.

In calculating posterior probabilities on a variable, the network goes through a process called *network propagation* (also called belief updating, or belief inference). This refers to updating the probability distribution of the possible states of a node when it receives evidence from one or more of its neighboring nodes. A network is assumed to update the causal weights in the network each time a new piece of evidence comes into the network. Updated posterior probabilities can be obtained as

Bayesian Network Posterior Probability = α Likelihood × Prior Probability; that is,

$$P(H_i|e) = \frac{\mathrm{P(e|H_i)P(H_i)}}{\mathrm{P(e)}} = \alpha L(H_i|e_1, e_2, e_3, ..., e_n)P(H_i), \tag{1}$$

where e is a set of evidence $e_1, e_2, e_3, \ldots, e_n$, H_i is a hypothesis of interest, and α is a normalizing constant.

Assume X_i is a node representing a bug or misconception. When information about X_i is provided by one of its neighboring nodes (denoted as e), the updated posterior probability on X_i is calculated by the following equation with the message-passing form of network propagation (Pearl, 1988):

$$P(X_i = x_i|e) = P(X_i = x_i|e) = \alpha \sum_u P(x|u)\pi_x(u) \prod_{j=1}^{n} \mathrm{P}(Y_j(X_i) = y_j|W(Y_j) = w_j), \tag{2}$$

where $\pi_X(u) = P(u|e)$, α is a normalizing constant, and u is the set of X_i's parent nodes (subskills with direct causal links to bug X_i). $Y(X_i)$ is the set of X_i's child nodes (item performance variables caused by bug X_i), and $W(Y_j)$ is the set of parents of Y_j.

Other algorithms besides message passing (Pearl, 1988) have been proposed to perform the belief updating, including trees of cliques (Jensen, 1996; Lauritzen & Spiegelhalter, 1988), qualitative propagation (Henrion & Druzdzel, 1990), cutset conditioning with the clique-tree method (Suermondt, Cooper, & Heckerman, 1990), and symbolic manipulations of sums and products (D'Ambrosio, 1991). Pearl's message-passing algorithm has been one of the most widely used propagation algorithms, perhaps because of its efficient formula for calculating posterior probabilities when a set of evidence is introduced into a network. However, even with message passing, in a tree having m children per parent and n values per node, $n^2 + mn + 2n$ real numbers and $2n^2 + mn + 2n$ multiplications are needed for each update (Pearl, 1988). Thus, the computation of updated posterior probabilities can be very complex, even in simple cases involving only one or two pieces of evidence instantiation affecting no more than a few nodes. Efficient computer programs have been developed for the message propagation calculations, including First Bayes (www.tonyohagan.co.uk/1b/), HUGIN (www.hugin.com), MSBNx (research.microsoft.com/adapt/MSBNx/), WinBUGS (www.mrc-bsu.cam.ac.uk/bugs/), GeNIe (genie.sis.pitt.edu/about.html), and Netica (www.norsys.com). In the present study, HUGIN (Andersen, Olesen, Jensen, & Jensen, 1989) was used to instantiate the Bayesian networks and to perform student diagnosis by calculating posterior probabilities.

The Data: VanLehn's (1981) Study of Subtraction Bugs

VanLehn (1981) identified a variety of bugs and subskills involved in multicolumn subtraction. In an empirical study, students from the third to fifth grades from two school districts in the southern San Francisco Bay area took a short subtraction test. Each form of the tests presented two-, three-, or four-digit integer subtraction problems in an open-ended format. All actual answers of the students were analyzed to build the taxonomy of subtraction bugs. Data from 520 examinees who were given Form 1 were analyzed. The 14 (of 17) test items that were analyzed in the present article are: Item 1 (43 – 7), Item 2 (80 – 24), Item 3 (127 – 83), Item 4 (183 – 95), Item 5 (106 – 38), Item 6 (800 – 168), Item 7 (513 – 268), Item 8 (411 – 215), Item 9 (5391 – 2697), Item 10 (3005 – 28), Item 11 (854 – 247), Item 12 (700 – 5), Item 13 (608 – 209), and Item 14 (3014 – 206).

Subtraction Bugs

The four most common bugs identified in these data by VanLehn (1981) were selected for analysis. These bugs are Smaller-From-Larger (Bug 1), Stop-Borrow-At-Zero (Bug 2), Borrow-Across-Zero (Bug 3), and Borrow-No-Decrement (Bug 4). Descriptions of these subtraction bugs are as follows:

Bug 1. Smaller-From-Larger: When borrowing is needed from the left column, a student switches the top and bottom numbers and subtracts the smaller from the larger number. The column that is supposed to be borrowed from remains unchanged.

Bug 2. Stop-Borrow-At-Zero: Borrowing from zero is carried out in the ones column and the subtraction is correct for that column, but zero in the tens column remains unchanged. The subtraction in the hundreds column, however, is correctly done again with a decrement for the tens column.

Bug 3. Borrow-Across-Zero: Borrowing for the ones column is carried out from the hundreds column. Zero in the tens column remains unchanged.

Bug 4. Borrow-No-Decrement: Borrowing for the ones column is carried out correctly, except that there is no decrement in the tens column which should have been decremented.

These four bugs generate different characteristic wrong answers. For example, consider the subtraction item 306 – 187. The correct answer to this problem is 119. If a student has only Bug 1, Smaller-From-Larger, his or her wrong answer will be 281. The characteristic wrong answer generated by Bug 2, Stop-Borrow-At-Zero, is 129. For Bug 3, Borrow-Across-Zero, the characteristic wrong answer is 29, whereas for Bug 4, Borrow-No-Decrement, it is 229. Thus, particular wrong answers are diagnostic of the corresponding bugs. This assumption, that the bugs (and combinations of bugs) generate specific characteristic wrong answers, is crucial to the present investigation. A student was operationally defined as exhibiting a bug if he or she showed a specific characteristic answer of the bug at least once across the set of test items.

Subtraction Subskills

The present study also hypothesized a set of latent procedural subskills that are presumed to play a role in students' subtraction performance. In terms of the Bayesian network, these subskills are assumed to be causally related to the subtraction bugs. Four basic multicolumn subtraction subskills identified by VanLehn (1981) were used for the current study, as follows:

Subskill 1. Borrow: When the smaller number appears on the top and the larger number on the bottom in the ones (or the tens) column, a decrement should be made in the tens (or the hundreds) column.

Subskill 2. Double-Borrow: When the smaller number appears on the top and the larger number on the bottom in both the ones and the tens columns, decrements should be made in both the tens and the hundreds column.

Subskill 3. Subtract-From-Zero: When zero appears on the top either in the ones or the tens column, a ten or a hundred should be borrowed for the ones and the tens columns, respectively, and the corresponding decrement should be made.

Subskill 4. Borrow-From-Zero: When the smaller number appears on the top in the ones column and the number in the tens column is zero, a hundred should be borrowed for the tens column and then a ten should be borrowed for the ones column. The corresponding decrements should be made as well.

Rationale and Design of the Present Study

Diagnosis of procedural subskills, or bugs in those subskills, uses information about the examinee's performance in some specific test context. Clearly, item format can play a role in how effectively and successfully cognitive or educational diagnostic assessment can be carried out for an individual student. Research studies such as VanLehn (1981) may use items in open-ended format in order to obtain information diagnostic not only regarding a student's mastered and nonmastered subskills but also regarding exactly why a student failed to answer an item correctly. On the other hand, large-scale high-stakes tests tend to incorporate multiple-choice items, for ease in test management including scoring. One possible way to realize both these advantages would be to design multiple-choice test items in which the incorrect answers are constructed to be diagnostic of specific common errors or bugs. Use of incorrect answers as evidence has been tried in diagnostic testing extensions of the graded response model (Samejima, 1995) in the context of latent trait modeling (e.g., Hemker, Sijtsma, Molenaar, & Junker, 1997; Sadler, 1998), and has recently been explored using other cognitive models (de la Torre, 2009; Lee & Corter, 2003).

A simulation was conducted exploring this idea of constructing multiple-choice test items where the incorrect alternatives are diagnostic of specific misconceptions or bugs. To do this, two different item scoring schemes were incorporated into the building and testing (BM) model: first, binary scoring of the answer to each item as either correct or incorrect (the scoring scheme typically used in large-scale tests) and, second, a diagnostic scoring scheme simulating a multiple-choice test format constructed specifically to identify specific bugs. The type of diagnostic multiple-choice test we aimed to simulate is one in which the three incorrect answers for each item correspond to the characteristic answers generated by the specific bugs that the diagnostics were attempting to identify. In the present study, a more data-driven approach was adopted, using as the three simulated ''distractors'' for each test item the three most common incorrect answers given by examinees in the VanLehn (1981) study. In the studies reported below, the performance of Bayesian networks in these two testing situations is compared.

In designing an effective Bayesian network, a structure is sought that is parsimonious but at the same time represents the appropriate causal relationships of the domain variables. In the present investigations, several assumptions were incorporated to minimize the networks' structural and computational complexity. These assumptions were as follows: (a) certain bugs were implemented as direct causes of item performance; (b) item performance variables were treated as mutually independent or as conditionally independent given bug states; (c) in certain of the networks, subskills were incorporated as direct causes of the bugs; (d) because each bug is believed to represent a unique form of incorrect procedure, bugs were assumed to be either independent or conditionally independent given specific subskills; and (e) certain interdependencies among the subskills were assumed and implemented in the network structure.

Based on these assumptions, four Bayesian networks were constructed by varying the specificity of information available from test items (binary correct–incorrect information only vs. diagnostic specific-answer information) and whether subskill nodes were assumed as causes of the bugs (bug only vs. bug and subskills). The resulting four networks were a binary-answer bug network (Network 1, see Figure 2), a binary-answer bug-plus-subskill network (Network 2,

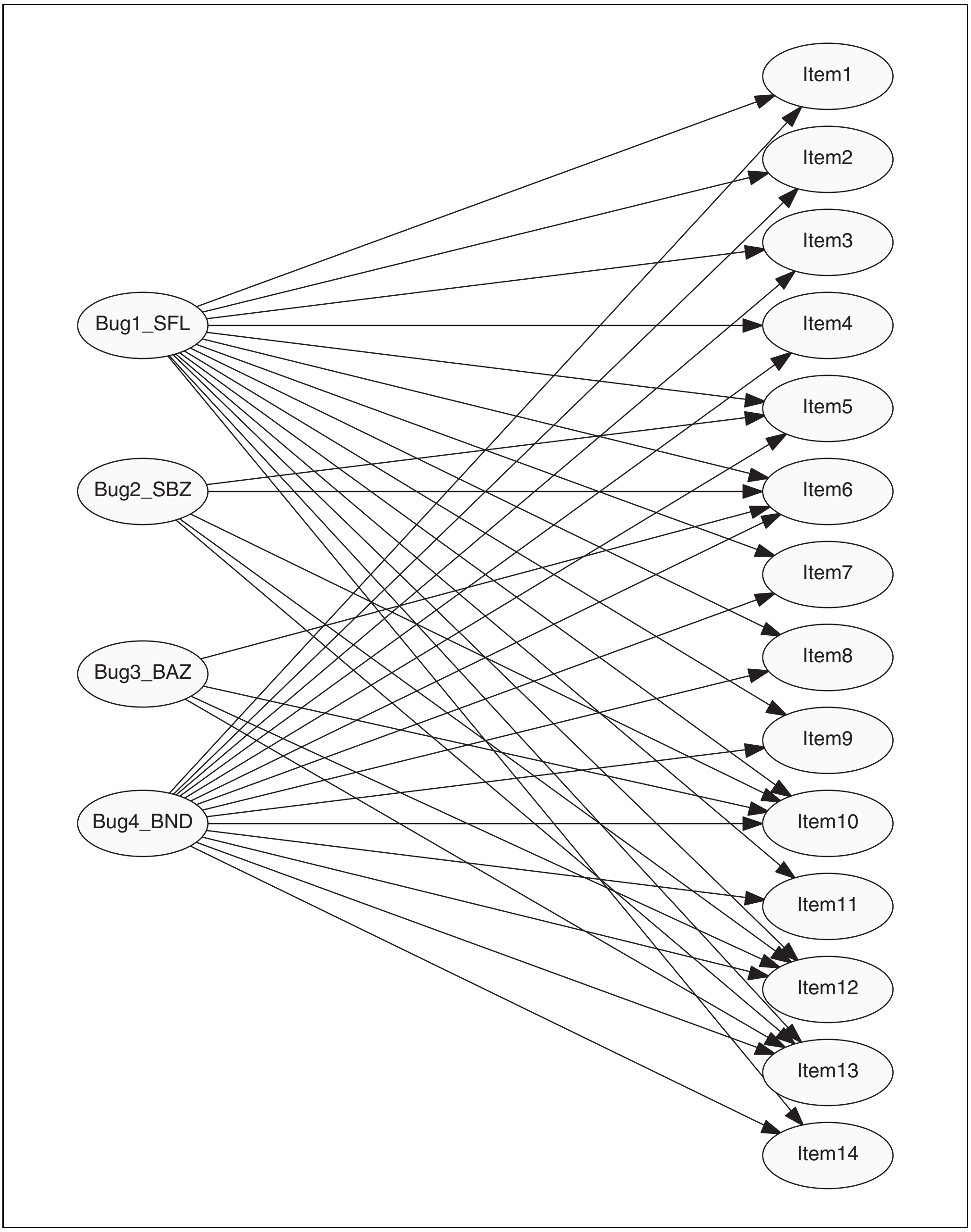

Figure 2. Network 1: Binary-answer bug Bayesian network

Note: The subtraction bugs are direct causes of item correct–incorrect performance. In the diagram, the bugs are labeled SFL = Smaller-from-Larger, SBZ = Stops-Borrow-at-Zero, BAZ = Borrow-Across-Zero, and BND = Borrow-No-Decrement.

see Figure 3), a specific-answer bug network (Network 3), and a specific-answer bug-plus-subskill network (Network 4).

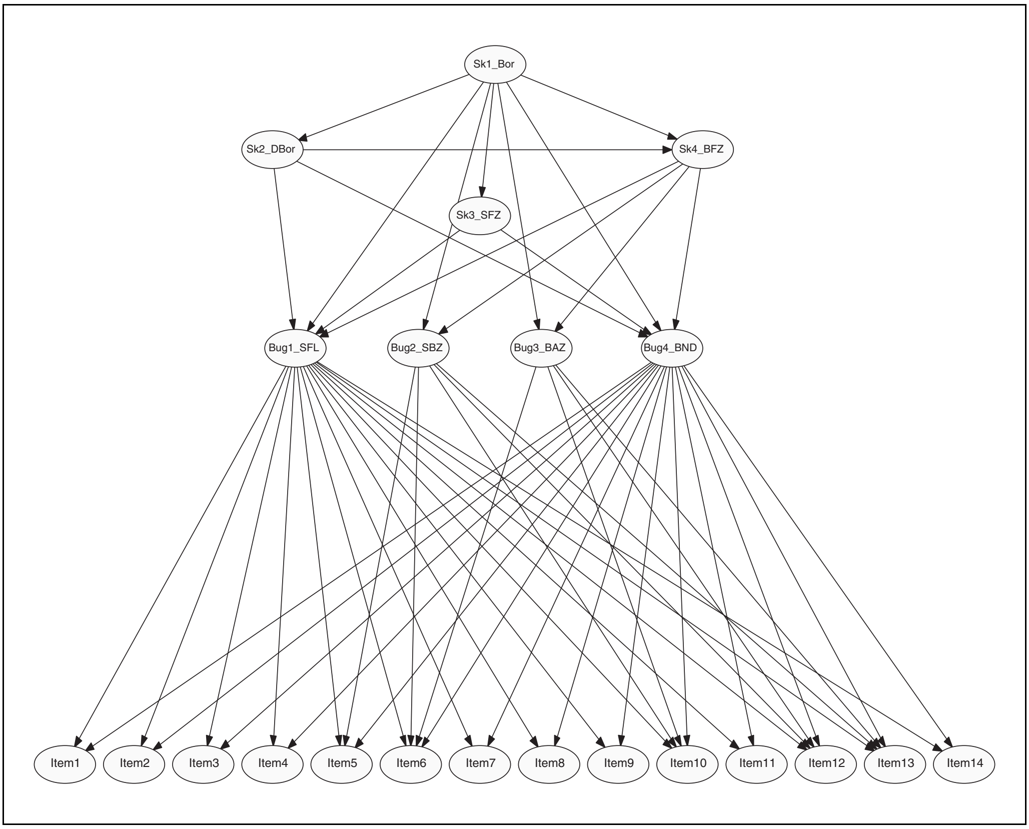

Figure 3. Network 2: Binary-answer bug-plus-subskill Bayesian network

Note: The subtraction bugs are direct causes of item correct–incorrect performance. Lack of a subskill is a direct cause of the corresponding subtraction bugs. Note that Subskills 2, 3, and 4 are specializations of Subskill 1, and Subskill 4 is also a specialization of Subskill 2.

Network 1 (Figure 2) was constructed so that the bugs were the direct and only causes of item performance. The network's bug predictions were made based on the evidence provided by a vector of binary-valued results from test items (i.e., correct or incorrect for each item). Network 2 (Figure 3) was constructed so that the bugs were the direct causes of item responses and the subskills were the direct causes of the appropriate bug manifestations. Subskills themselves are assumed to have a hierarchical structure representing that some subskills are specializations of others. This network was used to investigate the question of whether the addition of subskill nodes as direct causes of the bugs would enhance bug diagnosis. Network 3 (not shown) used only four latent nodes representing bugs, like Network 1, but was constructed to evaluate the effect of using specific bug-diagnostic answers as input to the network as opposed to simply using correct or incorrect item performance. Network 3 differs from the one shown in Figure 2 only in that each item node in Figure 2 is replaced with four item nodes (each having the identical parent node set) representing the four most frequent answers for that item. For each item, one correct and three incorrect potential answers were incorporated in the network, mimicking multiple-choice test items with four response alternatives. For instance, the four nodes related to Item 4 ($183 - 95$) represent student answers of either 88 (the correct answer) or 112, 98, and 128 as common incorrect answers. Note that each of these incorrect answers is diagnostic of a particular bug: The answer 112 is indicative of Bug 1 (occurring in the tens and hundreds columns), 98 is diagnostic of Bug 4, and 128 is indicative of Bug 1 (occurring only in the hundreds column). Similarly, Network 4 (not shown) expanded Network 2 (the bug-plus-subskill network) by employing four specific-answer nodes to summarize the performance on each item.

To summarize, input to the binary-answer networks (Networks 1 and 2) was test item performance information limited to item correctness or incorrectness, as would occur with correct–incorrect scoring. On the other hand, in the specific-answer networks (Networks 3 and 4) four possible answers for a given item were implemented, simulating a type of *diagnostic* multiple-choice test with a form of scoring that goes beyond partial credit scoring in that it is designed to identify the presence of particular bugs or mastery of particular subskills.

Network Evaluation

The primary interest in the network evaluation was to ask the specific question: How accurately and reliably can these Bayesian network models diagnose a target set of subtraction bugs? To answer this question, a cross-validation holdout sample method was used, with a random split of the data into two sets: modeling ($n = 512$, 80% of the entire sample) and testing ($n = 129$, 20% of the entire sample). For a discussion of other potential methods of assessing model fit for Bayesian networks in CDM applications, see Sinharay (2006).

To implement the present method and validate the results, relatively independent and objective evidence is required as to whether a particular student possesses a particular bug. To this end, the raw test data were coded by objective rules to add information about the mastery of subskills and the presence of bugs. In both the modeling subsample and the validation subsample, a student was operationally defined as having a particular bug if at least one of the student's specific wrong answers (in the original open-ended item test) showed the characteristic answer produced by the bug. A student was defined as possessing a subskill if he or she successfully answered at least 50% of the test items requiring it. Note that use of this validation approach requires test data that provides students' specific erroneous (and therefore diagnostic) answers, and thus is possible only with a data set consisting of open-ended or constructed-response items, or using multiple-choice items with distractor alternative answers designed to elicit information about specific misunderstandings.

The modeling sample was used to estimate each network's parameters. In this estimation stage, the values of the parameters (the simple and conditional probability distributions) were directly estimated using the empirical data (including the inferred bugs and subskills for each examinee) from the modeling sample. Where data were incomplete (e.g., when a particular combination of the bugs directly affecting an item node was not observed in the modeling sample), reasonable values were supplied (see Lee, 2003, Appendix A, for the complete conditional probability distribution tables used as network parameters). Note that in the modeling phase, the data provided to the network consists of each student's vector of test item performance data (in either correct–incorrect or specific-answer form) plus the student's observed criterion values on the higher order variables of interest: the four bugs, plus possibly four subskills (in Networks 2 and 4). These data determine the network parameters (the conditional posterior probabilities). For example, in Network 1 (Figure 2) Item 9 is assumed to be causally affected by Bug 1 (Smaller-From-Larger) and Bug 4 (Borrow-No-Decrement). Parameter estimation yields an estimated conditional probability of getting the item correct given that *both* bugs have been exhibited by the individual as .22. If both bugs are not present, the probability of getting Item 9 correct is .73. If only Bug 1 is present, the probability of getting the item correct is estimated to be .49, and for Bug 4 alone, it is estimated as .52.

Once the network parameters were estimated, the HUGIN program was used to perform the posterior probability calculations as data from the validation (testing) sample was entered. In this phase, the vector of item performance data was entered for each individual student in the testing sample, and the network produced posterior probabilities for the bugs and subskills of each student. The four network models were validated based on their capabilities to correctly diagnose the presence of the subtraction bugs in each individual student in the holdout sample. In the present network evaluations, the percentage of students correctly classified on each bug was used as the network performance criterion. To predict the presence or absence of specific bugs, cutoff points were applied to the posterior probabilities of each bug calculated for an individual student, in order to obtain deterministic categorical predictions for presence or absence of the bugs. Then, a comparison of the networks' predicted bugs with the observed bug manifestations was made. Several schemes to set the cutoff points for the predictions using these posterior probabilities were employed. Specifically, fixed cutoff points of .25 and .50 were tried, along with two schemes using variable-specific cutoff points. The first such variable cutoff scheme used the observed base rates of each specific bug (estimated using the modeling sample) as the cutoff point for the posterior probability, whereas the second variable cutoff point scheme was based on a procedure using percentile ranks. The percentile rank cutoff points were obtained by finding the percent cutoff points in the predicted distribution of posterior probabilities for each specific bug (based on the modeling sample) that correspond to the base rate of each bug. For Network 1 (binary answers, bug nodes only), these cut points were .45, .51, .27, and .53 for Bugs 1, 2, 3, and 4, respectively. For Network 2 (binary answers, bug + subskill nodes), the cut points were .34, .68, .27, and .43. For Network 3 (specific answers, bug nodes only) the cut points were .35, .54, .07, and .82, whereas for Network 4 (specific answers, bug + subskill nodes) the cut-points were .39, .62, .08, and .87. Using these various types of cutoff points, the ‘‘predicted bug manifestations'' were obtained for each student. Then, comparisons between the observed and predicted bug manifestations were made in order to evaluate network performance.

Results

The observed ‘‘true'' bug manifestation base rates in the testing sample for Bugs 1 through 4 were .39, .31, .07, and .43, respectively. The predicted bug manifestation rates for the testing sample produced by the four networks are presented in Table 1.

Table 1 indicates that the students' observed bug manifestations were predicted reasonably well across all four networks. However, as predicted, the network using specific answers and including subskill nodes (Network 4) performed best. That is, the bug manifestations generated by Network 4 showed the closest fit to the observed bug manifestations. In fact, both networks using the student's specific answers as input data (Networks 3 and 4) performed better than the ones using only binary item information (Networks 1 and 2), regardless of the cutoff points used. The use of subskill nodes somewhat increased the networks' prediction performance within each type of network, especially comparing Network 1 versus 2. It can be seen that the effect of using specific-answer information is large relative to the effect of adding subskill nodes. This finding leads to the conclusion (perhaps not surprising) that bug diagnosis is more effective when the student's specific wrong answers are used as evidence.

The modest improvement in network prediction attained by including subskill nodes suggests that the additional causal relationship posited between the subskills and the bugs helped the networks distinguish between erroneous answers caused by the bugs and errors generated by other factors. It also suggests that there is more stability in assessing the possession of a subskill than in predicting the presence of a bug.

Table 1. The Averages of Observed Bug Base Rates (in the Modeling Sample) and Predicted Bug Manifestation Rates in the Validation Sample by the Four Network Models With Various Cutoff Points

	Observed base rate of bug	Network 1: Binary info, bug only	Network 2: Binary info, bug + subskill	Network 3: Specific answers, bug only	Network 4: Specific answers, bug + subskill
Fixed cut point = .25					
Bug 1	.39	.44	.43	.42	.40[a]
Bug 2	.31	.49	.44	.36[a]	.36[a]
Bug 3	.07	.09	.12	.07[a]	.07[a]
Bug 4	.43	.62	.50	.58	.51[a]
Fixed cut point = .50					
Bug 1	.39	.39[a]	.39[a]	.40	.40
Bug 2	.31	.34	.39	.33[a]	.33[a]
Bug 3	.07	.01	.04	.07[a]	.07[a]
Bug 4	.43	.47	.44[a]	.50	.47
Variable cut points using base rates					
Bug 1	.39	.41	.39[a]	.40	.40
Bug 2	.31	.41	.41	.35[a]	.35[a]
Bug 3	.07	.41	.41	.07[a]	.07[a]
Bug 4	.43	.47	.44[a]	.50	.47
Variable cut points using percentile ranks					
Bug 1	.39	.40[a]	.40[a]	.40[a]	.40[a]
Bug 2	.31	.33	.36	.33	.32[a]
Bug 3	.07	.09	.09	.08[a]	.08[a]
Bug 4	.43	.46	.45	.44[a]	.44[a]

[a] Indicates the best-fitting network for that bug, given a cut point scheme.

Interestingly, the use of a .50 cut point resulted in the best recovery of the base rates across all the bugs. One possible reason for this finding is that this cut point corresponds to the point at which the odds change from favoring the presence of the bug to favoring the absence of the bug. Thus, this result can be seen as offering validation for the probabilistic interpretation of parameters in the Bayesian network. Overall, the findings concerning the relative performance of the four networks did not vary across use of different cutoff points.

Classification Rates

The classification rates of each network using the various cutoff points are reported in Table 2. The classification rates shown are the percentage agreement between the predicted and observed bug manifestations. That is, these numbers are based on the total number of positive hits (the network predicted presence of a bug in a student and he or she actually had it) and negative hits (the network predicted absence of a bug in a student and he or she actually did not have it).

These classification rates shown in Table 2 seem impressive. However, it is well known that when a phenomenon is rare (such as the presence of Bug 3 here), the percentage agreement can be large simply because of the high prevalence of negative hits. Therefore, an arguably more meaningful statistic is κ (Cohen, 1960). Cohen's κ adjusts the classification rates (percentage agreement) for the degree of chance agreement caused by the observed marginal frequencies of the events in question. Values of κ for the network model predictions are shown in Table 2, in parentheses.

Table 2. Correct Classification Rates in the Validation Sample for the Four Networks

Cut points	Fixed at .25	Fixed at .50	Using base rate	Using percentile rank
Network 1: binary info, bug only				
Bug 1	83 (.65)	86 (.71)	84 (.66)	85 (.68)
Bug 2	82 (.64)	89 (.74)	90 (.78)	90 (.77)
Bug 3	90 (.29)	92 (.00)	64 (.14)	91 (.31)
Bug 4	77 (.55)	85 (.69)	85 (.69)	85 (.69)
Overall	83 (.53)	88[a] (.54)	81 (.57)	88[a] (.61[a])
Network 2: binary info, bug + subskill				
Bug 1	85 (.69)	86 (.71)	86 (.71)	87 (.72)
Bug 2	87 (.72)	92 (.83)	90 (.78)	91 (.80)
Bug 3	89 (.35)	91 (.18)	66 (.19)	91 (.31)
Bug 4	85 (.69)	89 (.78)	89 (.77)	88 (.76)
Overall	87 (.61)	90[a] (.63)	83 (.61)	89 (.65[a])
Network 3: specific answers, bug only				
Bug 1	97 (.94)	100 (1.00)	99 (.98)	99 (.98)
Bug 2	95 (.89)	98 (.95)	96 (.90)	98 (.95)
Bug 3	100 (1.00)	100 (1.00)	100 (1.00)	99 (.92)
Bug 4	85 (.71)	92 (.84)	92 (.84)	97 (.94)
Overall	94 (.89)	98[a] (.95[a])	97 (.93)	98[a] (.95[a])
Network 4: Specific answers, bug + subskill				
Bug 1	99 (.98)	99 (.98)	100 (1.00)	99 (.98)
Bug 2	95 (.89)	98 (.95)	96 (.91)	99 (.98)
Bug 3	100 (1.00)	100 (1.00)	100 (1.00)	99 (.92)
Bug 4	93 (.86)	97 (.94)	97 (.94)	99 (.98)
Overall	97 (.93)	99[a] (.97[a])	98 (.96)	99[a] (.97[a])

Note: For each cell, the first entry shows the raw percentage agreement of how students are classified on each bug; the entry in parentheses gives Cohen's kappa.

[a]Indicates cut point scheme resulting in the best overall classification rate for that network.

Network 1 (binary data, bug nodes only). As shown in Table 2, the raw classification rates produced by Network 1 are greater than 80% across the four bugs. However, the values of κ for Network 1's prediction across the four bugs were about 50% to 60%. This moderate level of κ resulted mainly from Bug 3, whose κ was less than 31% using the different cutoff points. This is evidently due to the fact that 93% of the students in the testing sample did not have Bug 3.

Network 2 (binary data, bug and subskill nodes). The classification results generated by Network 2 showed a slight improvement from Network 1 in the prediction rates across the bugs (Table 2). The relative advantage of Network 2 was more noticeable in the values of κ. About a 9% increase in the κ values was observed when using a .50 cut point. Network 2 diagnosed Bug 2 and Bug 4 particularly well, showing κ values of 83% and 78%, respectively, using a .50 cut point. For Bug 3, Network 2 did slightly better but still poorly (based on the κ).

Network 3 (specific answer data, bug nodes only). Network 3 showed great improvements in classification rates compared to the previous two networks. Excellent classification performances were obtained ranging from 95% to 100% on the prediction of the three bugs (i.e., Bugs 1, 2, and 3) regardless of the cutoff points used. The Bug 4 prediction was slightly worse than the other three bugs, but still in the range of ''good'' prediction. Overall, 94% to 98% classification rates averaging across the four bugs were obtained using the various cutoff points (Table 2).

Table 3. Prediction of Bug Patterns in the Validation Sample by the Four Networks, Using Four Cut Point Schemes

Cut points:	Network 1: Binary info, bug only	Network 2: Binary info, bug + subskill	Network 3: Specific answers, bug only	Network 4: Specific answers, bug + subskill
Fixed at .25	42	72	66	85[a]
Fixed at .50	75	81	85	92[a]
Base rates	20	24	85	91[a]
Percentiles	83	88	91	92[a]

Note: Each entry shows the raw percentage agreement of how well students are classified by a network in terms of their "true" pattern on all four bugs.

[a] The best-fitting network for predicting bug patterns, for a given cut point scheme.

These prediction rate increases are not surprising when the type of information provided for Network 3 instantiations is considered: students' actual answers among four possible specific answer choices. However, given the fact that only the four most likely answers for each item were included (i.e., in fact, not all students' answers were incorporated within the network structure and instantiations), the prediction rates obtained by Network 3 still seem impressive. These substantial increases in prediction rates of Network 3 appeared also in the values of κ: Cohen's κ exceeded 90% for all four bugs using the percentile-rank cutoff point scheme. The effects of providing the network with specific answers were especially remarkable in the prediction of Bug 3. Network 3 was able to predict the presence of Bug 3 extremely accurately when the student information was specific (i.e., meaning they chose one of the four answer options), with a median κ value of 100% across the four cutoff point schemes.

Network 4 (specific answer data, bug and subskill nodes). Network 4 demonstrated the best prediction results for all four bugs. It also showed very stable performance across the different cutoff points. The classification rates ranged from 93% to 100% across the four bugs, with 99% classification rates in all four bugs using the percentile-rank cutoff point scheme (Table 2). The improvement of Network 4's prediction rate (by adding the subskill nodes) compared to that of Network 3 is noticeable in the values of κ. The mean κ values, averaged across the four bugs, were greater than 95% for all but the first cutoff point scheme.

The results reported in Table 2 provide evidence for excellent classification rates by the proposed Bayesian networks. However, the networks vary considerably in their relative performance. The strongest effect apparent from these tables is the effect of using specific wrong answers as diagnostic information for the bug prediction. Use of this information results in considerable improvement in the network classification performance for all four bugs. Positive effects of using subskill nodes were also observed, but they were relatively small.

Bug Pattern Analysis

Analyses were also conducted to see how well the networks could achieve simultaneous prediction of all four bugs. Each student was assigned to one of the bug groups, according to his or her bug pattern, that is, a vector of the observed presence or absence of each of the four bugs, based on analysis of the student's answers to the open-ended items. This resulted in assignment of each student to 1 of 16 possible observed bug patterns. Then, these observed bug patterns were compared with the networks' predicted bug pattern for each student. The raw percentage agreement (in terms of exact matches) of the observed and predicted bug patterns using the various cutoff points is shown in Table 3.

Table 4. Model Fit Statistics for the Four Network Models ($N = 520$)

	Binary data (correctness)	Specific answer data
Bug-only network	AIC: –3793.38 BIC: –4031.59 LL: –3681.38	AIC: –9701.68 BIC: –10156.84 LL: –9487.68
Bug-plus-subskill network	AIC: –3716.13 BIC: –4050.05 LL: –3559.13	AIC: –9587.06 BIC: –10137.93 LL: –9328.06

Note: Smaller absolute values of each statistic indicate better fit. AIC = Akaike's information criterion; BIC = Bayesian information criterion; LL = log-likelihood.

When the networks were compared on their capabilities to predict the bug patterns, substantial effects of both specific wrong answers and subskills were observed. As was the case for predicting a single bug, the specific-answer networks (Network 3 and 4) performed substantially better than the binary-answer networks (Network 1 and 2). Also noteworthy were the effects of using subskill nodes, as reflected in the difference between Network 1 versus Network 2, and between Network 3 versus Network 4. As expected, the best fit with the observed bug pattern manifestations was found in Network 4 across all four cutoff points. All in all, the networks' overall capability of predicting the four bugs simultaneously was only slightly worse than the networks' capability of predicting one bug at a time. The differences resulting from using different cutoff points were more noticeable in bug pattern classification results than in the single bug classification results.

Model Fit Statistics for the Network Models

It could be argued that the finding that use of a more complex network with additional latent nodes achieves a better fit to the data is not surprising; models with more parameters tend to fit better. However, this result is not logically necessary in a cross-validation study; if overfitting has occurred, a more highly parameterized model may show worse fit to the data from the testing sample, either in terms of model fit statistics or in terms of prediction performance. Thus, the results above can be taken as serious evidence that addition of subskill nodes can stabilize and improve the diagnosis of unstable bugs.

However, use of model fit statistics such as Akaike's information criterion (AIC) or the Bayesian information criterion (BIC) can provide another perspective regarding the value of adding subskill nodes to the networks. Thus, the four network models (with the same parameter estimates as were used in the prediction analyses above) were applied to the complete data set ($N = 520$). The resulting fit statistics—AIC, BIC, and the model log-likelihood—are shown in Table 4. The values are reported in the form reported by HUGIN, without the multiplying constant of –2 used by Akaike; a lower absolute value represents better fit. Because the networks predicting binary correctness and the networks predicting specific answers are applied to different (though related) data sets, fit measures should be compared only within a column. The absolute value of the log-likelihood decreases when subskill nodes are added, indicating better model fit, as expected. AIC and BIC measure fit adjusted for model complexity. For the binary correctness data, results are unclear: AIC indicates that the bug-plus-subskill architecture is slightly better than the bug-only network, whereas BIC indicates the reverse. This is not surprising, because BIC is known to favor more parsimonious models to a greater degree than AIC (Li, Cohen, Kim,

& Cho, 2009). However, for the specific-answer data, both AIC and BIC suggest that the networks with both bug and subskill nodes are superior to the bug-only networks.

Discussion

The results of the present study suggest that Bayesian networks can indeed be used effectively to diagnose bugs in multicolumn subtraction skills, even though such bugs are extremely unstable. Bug patterns were also identified reasonably well. A practical lesson from these results is that presence or absence of bugs can be effectively predicted using a fixed cut point of .50 on the posterior probabilities, a simple rule that can be implemented without using a modeling sample to estimate the cut points. Overall, the high successful diagnosis rates demonstrate that Bayesian networks are useful as tools for the cognitive diagnosis of bugs in cognitive skills.

A second major finding was that the bug diagnosis rates are improved by incorporating explicit subskill nodes in the network. This result suggests that the problem of bug instability that has stymied development of diagnostic measures might be improved by simultaneous diagnosis of subskills. A criticism that might be advanced here is that there was no real measure of external validity of the bug and subskill diagnoses. But some simple consistency checks on the network predictions indicated that bug and subskill diagnoses were reasonable. For example, in a new random split of the data into modeling ($N = 346$) and validation ($N = 174$) samples, the bug-plus subskill network applied to the binary-answer data gave dichotomized subskill predictions that were all positively correlated with total test score and bug predictions that were all negatively correlated with total test score (and the correlations of predicted bugs with predicted subskills were all either negative or 0). Furthermore, when subtest scores were defined for only those subtraction items involving subtraction from zero, and for only those items involving borrowing from zero, the differential pattern of correlations showed that each subskill diagnosis variable correlated most highly with the appropriate test or subtest score, for example, Skill 1 (borrow) with total test score, Skill 4 (borrow across zero) with the borrow-from-zero subtest score, etc. These findings demonstrate overall and divergent validity of the obtained bug and subskill diagnoses, and incidentally suggest that the present methods may be effective methods for cognitive diagnosis of subskills as well as for bugs.

Finally, the present study has shown that diagnosis of bugs (and subskills) in procedural skills is most successful when students' specific wrong answers are used as evidence. This type of information is available with open-ended or constructed-response test items, but it could also be provided by multiple-choice tests constructed for this purpose, as simulated here. Using such specific-answer information from the student, the bug prediction rates reached 90% in the bugs-only network. The best diagnosis was attained when both the subskill nodes and specific answers were incorporated in the network structure, in which case the prediction rate reached 97% or higher across the different cutoff points.

Bug Identification and Cognitive Diagnosis

The proposed Bayesian network framework shown in this present study can be classified as a special case of latent class models, and more narrowly as latent cognitive diagnostic models. Some fundamental features of cognitive diagnostic models are shared by the models considered in the present study. At the core of the measurement problem are latent variables (here, bugs). The latent variables are believed to be manifested through responses to a set of items (Sijtsma, 1998). Independence among items is assumed to be conditional on the attributes (or the bugs). The pattern of network connectivity between the bugs and the items serves as the basis for model building, just as the Q-matrix does in typical cognitive diagnostic models (e.g., Tatsuoka, 1985).

However, the models used in the present study also have some unique features compared to existing cognitive diagnosis models. Although typical cognitive diagnostic models have focused on modeling item performance using latent subskills (cognitive attributes), in the present study item performance was modeled using bugs, systematic errors that are independent of each other (in Networks 1 and 3) or in turn explained by higher order cognitive attributes or skills (in Networks 2 and 4).

Simultaneous Diagnosis of Bugs and Subskills

The present results suggest that diagnosis of bugs in procedural skills may be improved by a causal model that posits subskill knowledge or expertise as causal factors influencing the expression of bugs (cf. Templin et al., 2008). Thus, the findings support accounts suggesting that bugs may be flexible or improvised reactions to impasses in problem solving caused by insufficient mastery of a subskill (e.g., J. S. Brown & VanLehn, 1980).

For Bug 4 in particular, the use of subskill nodes resulted in a great improvement in prediction. One reason for this finding could be that the answers caused by Bug 4 (Borrow-No-Decrement) looked like computational slips, more so than other bugs investigated in this study. Adding the causal connection between Bug 4 and the corresponding subskills helped the networks to predict Bug 4 better. These subskill effects on Bug 4 predictions were observed in both binary- and specific-answer networks, which is in contrast with the other three bugs, for which the effects of adding subskills were found mainly in the binary-answer networks.

Several explanations might be advanced for the rather small subskill effects for the remaining bugs. But fundamentally, the relationships between the bugs and the subskills were not strong to begin with. This may be due to the great number of bugs that can arise when a single subskill is missing (VanLehn, 1990). It certainly seems possible that students' lack of a subskill could manifest itself in some other form than as one of these four bugs. Also, the use of only the four most frequent answers for each test item in the present study omits many possible answers characteristic of many other bugs, bug combination, or bugs combined with computational errors.

More than 200 different bugs were described in VanLehn (1990). Thus, one possible question to be addressed by future research is, How well will Bayesian networks work for diagnosing other bugs identified in the literature? How many bugs could the network structures incorporate without suffering problems with parameter estimation? However, it should be kept in mind that the rarest bug (Bug 3) occurred in less than 10% of the present sample. This suggests that many bugs, though they can be found in very large samples using open-ended test items, may not arise often enough to make it practical to devise diagnostic tests for them.

On the Benefits of Using Specific Wrong Answers as Diagnostic Evidence

An increase in prediction accuracy by using a student's specific incorrect answers was found for all bugs. Although diagnostic use of errors has been advocated by cognitive researchers for decades (e.g., Siegler, 1976), this idea has not yet been widely adopted in psychometrics (though see de la Torre, 2009; Lee, 2003; Lee & Corter, 2003; Sadler, 1998). Yet implications of this finding are clear, namely, that developing tests that are effective instruments for cognitive diagnosis may require a rethinking of item development and scoring procedures. Specifically, incorrect alternatives for each test item should be constructed so as to provide not only the information that the student does not know the correct answer, but also information as to *why* the student did not get the correct answer. Designing multiple-choice test items with incorrect alternative answers (distractors) that are characteristic of specific misconceptions or bugs will be a useful tool in developing large-scale tests that can be used for diagnosis of misconceptions and bugs in procedural skills.

The study results suggest that such information about specific wrong answers may be especially beneficial for the diagnosis of very infrequent bugs, because here the diagnosis of Bug 3 had the most benefit from adding subskill nodes. The networks were not able to predict Bug 3 accurately using binary item performance data. However, when the specific-answer information was provided, the networks predicted Bug 3 even better than the other bugs. In contrast, the effect of using specific-answer information was relatively small for Bug 4. Because Bug 4 is the most common bug (in the present sample), the networks attributed the causes of wrong answers to this bug more easily and did not need specific answers to indicate its presence.

One might ask why prediction was not better (or even perfect) when the diagnosis used students' actual wrong answers. One reason is that although the students were coded to have a bug when their wrong answers appeared to be symptoms of that particular bug, it is possible that those wrong answers that appeared to be ''buggy'' answers were in fact caused by other factors. Furthermore, not all of the students' wrong answers were incorporated into the network design. To simulate typical multiple-choice items, only the four most frequent answers were included in the network. Thus, some highly diagnostic information about students' incorrect responses for some items was not incorporated into the networks. This could be improved in future investigations and in testing applications. Finally, there is the possibility that bugs may interact, in the sense that a student with several bugs may show the single most characteristic answer for none of the bugs. Given these issues, the prediction rates achieved by the specific-answer networks in this study (i.e., greater than 90%) seem impressive.

Conclusions

It can be argued that the methods described here can improve diagnostic assessment of procedural skills, by helping to establish exactly why a student cannot execute a specific problem-solving operation. It could be that a procedural skill has not been learned, or has been learned incorrectly. Diagnosis of which subskills a student has mastered, and whether the student has a specific misunderstanding or bug, should enable more focused and informative assessments of student achievement and the effectiveness of instructional programs, perhaps leading to better individualized instruction (D. E. Brown & Clement, 1989; Chi, 2005; Chi, Siler, & Jeong, 2004).

It has been debated whether educational interventions can be designed or targeted more effectively if it is known that a student has a particular misconception versus merely knowing that the student does not have a required subskill (Chi, 2005). But strong posterior evidence that an individual has a particular bug (as opposed to making careless errors) may help to establish that a specific subskill needs remediation. Also, in some application domains a stable misconception may be quite resistant to remediation and thus may require special instructional effort (e.g., D. E. Brown & Clement, 1989; Chi et al., 2004). In such a domain, diagnosing bugs or other misconceptions would seem indispensable to creating appropriate targeted instruction. Thus, the methodology proposed here may help to trigger and shape the design of effective individualized instruction.

Acknowledgments

The main analyses reported here are based on results from the first author's doctoral thesis (Lee, 2003). Portions of this research were reported at the 2003 annual meeting of the Classification Society of North America, Tallahasee, Florida, June 2003. The authors thank Kurt VanLehn for making available the subtraction data from the Southbay study of multicolumn subtraction, Robert Mislevy for his helpful advice at an early stage of the project, and Charles Lewis and Matthias von Davier for providing useful feedback on an earlier draft of this article.

Authors' Note

Jihyun Lee now works at National Institute of Education, Nanyang Technological University, Singapore.

Declaration of Conflicting Interests

The author(s) declared no conflicts of interests with respect to the authorship and/or publication of this article.

Funding

The author(s) received no financial support for the research and/or authorship of this article.

References

Adams, R. J., Wilson, M., & Wang, W.-C. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement, 21*, 1-23.

Almond, R. G., & Mislevy, R. J. (1999). Graphical models and computerized adaptive testing. *Applied Psychological Measurement, 23*, 223-237.

Andersen, S. K., Olesen, K. G., Jensen, F. V., & Jensen, F. (1989). HUGIN—a shell for building Bayesian belief universes for expert systems. In N. S. Sridharan (Ed.), *Proceedings of the 11th International Joint Conference on Artificial Intelligence* (pp. 1080-1085). San Mateo, CA: Morgan Kaufmann.

Beland, A., & Mislevy, R. J. (1996). Probability-based inference in a domain of proportional reasoning tasks. *Journal of Educational Measurement, 33*, 3-27.

Brown, D. E., & Clement, J. (1989). Overcoming misconceptions via analogical reasoning: abstract transfer versus explanatory model construction. *Instructional Science, 18*, 237-261.

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

Brown, J. S., & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science, 4*, 379-426.

Carpenter, T. P., Franke, M. L., Jacobs, V. R., Fennema, E., & Empson, S. B. (1998). A longitudinal study of invention and understanding in children's multidigit addition and subtraction. *Journal for Research in Mathematics Education, 29*, 3-20.

Chi, M. T. H. (2005). Commonsense conceptions of emergent processes: Why some misconceptions are robust. *Journal of the Learning Sciences, 14*, 161-199.

Chi, M. T. H., Siler, S. A., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and Instruction, 22*, 363-387.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*, 37-46.

Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interactions, 12*, 371-417.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spieglehalter, D. J. (1999). Probabilistic networks and expert systems. New York, NY: Springer.

Cox, L. S. (1975). Systematic errors in the four vertical algorithms in normal and handicapped populations. *Journal for Research in Mathematics Education, 6*, 202-220.

D'Ambrosio, B. (1991). Local expression languages for probabilistic dependence. *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 95-102). San Mateo, CA: Morgan Kaufmann.

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

de la Torre, J., & Douglas, J. (2004). Higher-order latent trait models for cognitive diagnosis. *Psychometrika, 69*, 333-353.

Desmarais, M. C., & Pu, X. (2005). Computer adaptive testing: Comparison of a probabilistic network approach with item response theory. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *User Modeling 2005: Proceedings of the 10th International Conference, UM 2005, Edinburgh, Scotland, UK, July 24-29, 2005* (pp. 392-396). Berlin, Germany: Springer.

DiBello, L. V., Stout, W. F., & Roussos, L. A. (1995). Unified cognitive/psychometric diagnostic assessment likelihood-based classification techniques. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 361-389). Mahwah, NJ: Erlbaum.

Embretson, S. E. (1991). A multicomponent latent trait model for measuring learning and change. *Psychometrika, 56*, 495-515.

Gitomer, D. H., Steinberg, L. S., & Mislevy, R. J. (1995). Diagnostic assessment of troubleshooting skill in an intelligent tutoring system. In P. Nichols, S. Chipman, & R. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 73-101). Mahwah, NJ: Erlbaum.

Heiner, C., Heffernan, N., & Barnes, T. (2007, July). *Educational data mining*. Supplementary Proceedings of the 13th International Conference of Artificial Intelligence in Education. Marina del Rey, CA, 2007. Retrieved from http://aied.inf.ed.ac.uk/AIED2007/AIED-EDMproceedingfull2.pdf

Hemker, B. T., Sijtsma, K., Molenaar, I. W., & Junker, B. W. (1997). Stochastic ordering using the latent trait and the sum score in polytomous IRT models. *Psychometrika, 62*, 331-347.

Henrion, M., & Druzdzel, M. J. (1990). Qualitative propagation and scenario-based scheme for exploiting probabilistic reasoning. In P. B. Bonissone, M. Henrion, L. N. Kanal, & J. F. Lemmer (Eds.), *UAI '90: Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence* (pp. 17-32). New York, NY: Elsevier.

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

Jensen, F. V. (1996). *An introduction to Bayesian networks*. New York, NY: Springer.

Jensen, F. V., Lauritzen, S. L., & Olesen, K. G. (1990). Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly, 5*, 269-282.

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.

Korb, K. B., & Nicholson, A. E. (2004). *Bayesian artificial intelligence*. London, England: Chapman & Hall/CRC.

Kouba, V. L., Zawojewski, J. S., & Strutchens, M. E. (1997). What do students know about numbers and operations? In P. A. Kenney & E. A. Silver (Eds.), *Results from the sixth mathematics assessment* (pp. 87-140). Reston, VA: The National Council of Teachers of Mathematics.

Lauritzen, S. L., & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B, 50*, 157-224.

Lee, J. (2003). *Diagnosis of subtraction bugs using Bayesian networks* (Unpublished doctoral dissertation). Columbia University.

Lee, J., & Corter, J. E. (2003, June). *Diagnosis of bugs in multi-column subtraction using Bayesian Networks*. Paper presented at the annual meeting of the Classification Society of North America, Tallahassee, FL.

Li, F., Cohen, A. S., Kim, S.-H., & Cho, S.-J. (2009). Model selection methods for mixture dichotomous IRT models. *Applied Psychological Measurement, 33*, 353-373.

Maris, E. (1995). Psychometric latent response models. *Psychometrika, 60*, 523-547.

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

Martin, J., & VanLehn, K. (1995). A Bayesian approach to cognitive assessment. In P. Nichols, S. Chipman, & R. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 141-165). Mahwah, NJ: Erlbaum.

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

Mislevy, R. J., Steinberg, L. S., Breyer, F. J., & Almond, R. G. (2002). Making sense of data from complex assessments. *Applied Measurement in Education, 15*, 363-389.

Pardos, Z. A., Heffernan, N. T., Anderson, B., & Heffernan, C. L. (2007). The effect of model granularity on student performance prediction using Bayesian Networks. In C. Conati, K. McCoy, & G. Paliouras (Eds.), *User Modeling 2007: Proceedings of the 11th International Conference, UM 2007, Corfu, Greece* (pp. 435-439). Berlin, Germany: Springer.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Mateo, CA: Morgan Kaufmann.

Reckase, M. D. (1997). The past and future of multidimensional item response theory. *Applied Psychological Measurement, 21*, 25-36.

Russell, S., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. Englewood Cliffs, NJ: Prentice Hall.

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. (1995). Acceleration model in the heterogeneous case of the general graded response model. *Psychometrika, 60*, 549-572.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology, 8*, 481-520.

Sijtsma, K. (1998). Methodology review: Nonparametric IRT approaches to the analysis of dichotomous item scores. *Applied Psychological Measurement, 22*, 3-31.

Sijtsma, K., & Verweij, A. (1999). Knowledge of solution strategies and IRT modeling of items for transitive reasoning. *Applied Psychological Measurement, 23*, 55-68.

Sinharay, S. (2006). Model diagnostics for Bayesian Networks. *Journal of Educational and Behavioral Statistics, 31*, 1-33.

Sleeman, D. (1984). An attempt to understand students' understanding of basic algebra. *Cognitive Science, 8*, 387-412.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., & Cowell, R. G. (1993). Bayesian analysis in expert systems. *Statistical Science, 8*, 219-247.

Suermondt, J., Cooper, G., & Heckerman, D. (1990). A combination of cutset conditioning with clique-tree propagation in the pathfinder system. *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence* (pp. 245-254). New York, NY: Elsevier.

Tatsuoka, K. (1985). A probabilistic model for diagnosing misconceptions in the pattern classification approach. *Journal of Educational Statistics, 12*, 55-73.

Templin, J. L., Henson, R. A., Templin, A. E., & Roussos, L. (2008). Robustness of hierarchical modeling of skill association in cognitive diagnosis models. *Applied Psychological Measurement, 32*, 559-574.

VanLehn, K. (1981). *Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills* (Technical Report). Palo Alto, CA: Xerox Palo Alto Research Center.

VanLehn, K. (1990). *Mind bugs: The origins of procedural misconceptions*. Cambridge, MA: MIT Press.

VanLehn, K., & Martin, J. (1998). Evaluation of an assessment system based on Bayesian student modeling. *International Journal of Artificial Intelligence in Education, 8*, 179-221.

VanLehn, K., & Niu, Z. (2001). Bayesian student modeling, user interfaces and feedback: A sensitivity analysis. *International Journal of Artificial Intelligence in Education, 12*, 154-184.

von Davier, M., & Yamamoto, K. (2004). Partially observed mixtures of IRT models: An extension of the generalized partial-credit model. *Applied Psychological Measurement, 28*, 389-406.

Young, R. M., & O'Shea, T. (1981). Errors in children's subtraction. *Cognitive Science, 5*, 153-177.