Student Modeling from Conventional Test Data: A Bayesian Approach without Priors

Kurt VanLehn, Zhendong Niu, Stephanie Siler and Abigail S. Gertner

Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260
VanLehn@cs.pitt.edu, niu+@pitt.edu, siler+@pitt.edu, gertner+@pitt.edu

Abstract. Although conventional tests are often used for determining a student's overall competence, they are seldom used for determining a fine-grained model. However, this problem does arise occasionally, such as when a conventional test is used to initialize the student model of an ITS. Existing psychometric techniques for solving this problem are intractable. Straightforward Bayesian techniques are also inapplicable because they depend too strongly on the priors, which are often not available. Our solution is to base the assessment on the *difference* between the prior and posterior probabilities. If the test data raise the posterior probability of mastery of a piece of knowledge even slightly above its prior probability, then that is interpreted as evidence that the student has mastered that piece of knowledge. Evaluation of this technique with artificial students indicates that it can deliver highly accurate assessments.

Introduction

Diagnostic testing uses conventional test formats (e.g., items with multiple choice or numerical answers) but assumes that a student's competence can be characterized by a set of several subskills or factors. For instance, competence in multi-digit addition might be characterized with the subskills of carrying, processing columns without carries, and other subskills as well.

In its simplest form, a diagnostic test scoring algorithm inputs binary answer data (i.e., 1 if the question was answered correctly, 0 if answered incorrectly) and outputs a level of mastery for each of the subskills. Existing algorithms (e.g., DiBello et al., 1995; Samejima, 1995; Tatsuoka, 1990, 1995) enumerate all possible subsets of subskills that have distinguishable patterns of answers. The algorithms examine each subset and pick the one that best fits the answer data.

An unusual application of diagnostic testing arose during the development of the Andes physics tutoring system (VanLehn, 1996; Conati, Gertner, VanLehn & Druzdzel, 1997). Andes has a student modeler that uses Bayesian techniques to calculate the probability of mastery of each of about 350 rules. Because it is a Bayesian, it requires for each rule a prior probability, that is, the probability that a randomly drawn student from the population will have already mastered that rule

B.P. Goettl et al. (Eds.): ITS '98, LNCS 1452, pp. 434-443, 1998.

before using Andes. In order to find these prior probabilities, we planned to use diagnostic testing. That is, we planned to treat each of the rules as a distinct "subskill" or "factor," then use diagnostic testing to find out which "subskills" each student had mastered. By counting how many times each rule/subskill was mastered, we could estimate the prior probability of that rule in the population.

The test, which was developed by the physics instructors associated with the project, had 34 items, all of which had multiple choice or short-answer formats. We determined which of the Andes rules were required for correctly answering each problem. Overall, 66 rules were used at least once during the test.

Unfortunately, 66 "subskills"are much more than typically used in diagnostic testing. The existing scoring techniques would have to examine as many as 2^{66} subsets of rules in order to tell which subset best fit a given student's answers. This made existing scoring techniques inapplicable, so we had to develop a diagnostic test scoring algorithm that would infer, given a student's binary answer data, whether or not the student had mastered each of the 66 rules.

If such an algorithm can be found, then it could be used for other purposes besides determining priors for Andes student modeling. For instance, we could routinely give a student who was about to use Andes a short multiple-choice test. The results of the test could be used to initialize the student's model.

The algorithm needs to be told which rules (or subskills, etc.) are required for correctly solving each problem. If the problem can be solved with two or more correct strategies, then it may be necessary to use a disjunction in such a specification. For instance, if one strategy requires rules 1, 2 and 3, and the other strategy requires rules 3, 4 , 5 and 6, then one would have to specific that correctly solving the problem requires knowing rule 3 and either rules 1 and 2 or rules 4, 5 and 6.

However, items are usually written to have just one correct solution, although students may always invent ones that the authors did not anticipate. Psychometricians often assume that items have just one correct solution, which allows them to use a simplified representation, called a Q-matrix (Tatsuoka, 1990), for the relationship between knowledge and problems. In a Q-matrix, the columns are problems, the rows are pieces of knowledge, and cells are 1 if the piece of knowledge is required by the problem and 0 is if the knowledge is not required for solving that problem. Although a Q-matrix cannot accurately represent problems with multiple correct solution strategies, it has been found adequate for most tests, including the one we have used for testing.

The general problem can be stated as follows: Given

- a test with N items;
- a knowledge base of M rules (we will use "rule" to stand for any kind of unitization of knowledge);
- a M by N Q-matrix;
- an N-long binary vector indicating which test items the student answered correctly,

output a M-long real-numbered vector indicating the probability of mastery of each rule.

Research approach/method

Our approach was empirical: Try several diagnostic test scoring methods and evaluate them using artificial students. An artificial student is a function that, when given a set of mastery levels for the rules, produces a binary answer data vector. To use an artificial student for evaluation, the answer vector is fed into the test scoring method, which then predicts the mastery levels for each rule. If the predictions match the original mastery levels, then the test scoring method is accurate.

Although we would have preferred to evaluate the scoring methods with human students, that would require knowing with certainty what their levels of mastery were on each of the 66 rules.

In principle, artificial students should be based on real cognitive models. They should model forgetting, priming, guessing, cheating and all the other things students do during tests. However, we used a simpler framework that only modeled inadvertent mistakes (slips) and guessing. In our artificial students:

- P(the problem's answer is correct | all rules required for answering it are mastered) = 1 – slip.
- P(the problem's answer is correct | at least one rule required for answering it is not mastered) = guess / number of possible answers for this problem.

That is, the model has two global parameters: the probability of a slip and the probability of a guess. Once a person has decided to guess, whether they get the problem correct is inversely proportional to how many possible answers that multiple choice problem has. Since the equations above only give the probability of a problem's answer being correct, the probabilities are used as odds in a random number generator to generate the actual binary answer vector.

For evaluating the proposed assessments, we needed a collection of artificial students. We generated a collection of 201 artificial students with slip = .1 and guess = .3 by first randomly generating a level of competence C, then randomly generating rule masteries such that P(mastery of a rule) = C. That is, artificial students with high C had many rules mastered and student with low C had few rules mastered.

After the artificial students have been assessed and predictions about their mastery levels have been obtained, we need to measure the degree of match between the predicted and actual mastery levels. We evaluated the match separately for each rule, since we wanted to see if some rules were easier to assess than others. For each rule, we counted the number of artificial students where (A) the rule was actually mastered, (B) the rule was predicted to be mastered, and (C) the rule was both predicted to be mastered and actually mastered. Then:

- Accuracy is C/B. It should be as close to 1 as possible.
- Coverage is C/A. This should also be as close to 1 as possible.

Originally, we sought to maximize accuracy alone. However, when we were adjusting parameters (as described below), we found that accuracy reached a plateau rather than a peak as we varied the parameter values. We needed some way to select a parameter value among those which tied for producing the maximum accuracy. Thus, we adopted coverage as a desirable but secondary feature. To make parameter tuning easier, we define utility = coverage + 2*accuracy, which gives accuracy twice as much weight as coverage. We later considered the effects of replacing the "2" with

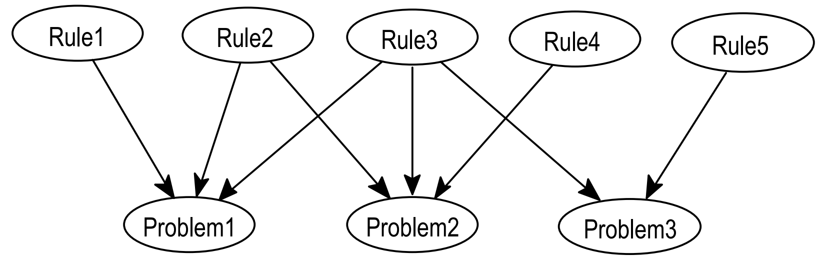

Fig. 1. A Bayesian network used for assessment.

other values in order to see if varying the relative importance of coverage and accuracy made any difference in our findings (see below).

Failures

We tried several obvious schemes but rejected them for various reasons. However, the scheme we ultimately accepted is based on what we learned from our failures, so we will discuss them first.

Noisy And

The first test scoring method we considered was based directly on the causality, as we perceived it. Mastery of rules *causes* problems to be answered correctly or not. Thus, we constructed a Bayesian network (Pearl, 1988) with nodes for problems and rules, and links from the rules to the problems (the causal direction). That is, a problem's node's parents were the nodes representing rules that were required for answering the problem correctly. For instance, in Figure 1, rules 1, 2 and 3 are required by problem 1, so they are the parents of problem 1 in the network. The conditional probability table of the problem's node is:

- P(answer correct | all rules required for answering it are mastered) = 1 – slip.
- P(answer correct | at least one rule required for answering it is not mastered) = guess / number of possible answers to that problem.

This is similar to a noisy And (Pearl, 1988), hence the name. It is also the same function as used in the artificial students, but the slip and guess parameters can be changed in order to optimize the performance of the assessment function.

Unfortunately, because the rule nodes are roots of the network, this approach requires assigning a prior for each rule node. In our application, the population priors are not known. Bayesians often use 0.5 for a prior when they don't know the actual prior. However, we discovered that for most rules, the posterior depended strongly on the choice of prior. For 46 of the 66 rules, when we tried the priors of .3, .5 and .8, the posterior was within ±0.05 of the prior. For the remaining 20 rules, the posteriors were relatively insensitive to the priors.

What seems to be happening with the 46 rules that are overly sensitive to the prior is the following:

- If a problem is incorrect, then *any* rule required by it could be unmastered. Thus, the network passes out blame rather thinly.
- If a problem is correct, then either the student guessed or all the rules must be mastered. Thus, even a little evidence of nonmastery of one of the rules will cause the network to infer that the student is guessing, which attenuates the credit the network passes out.

Although the above is just our interpretation of what happens, it is a fact that the network passes out credit and blame rather thinly. For most rules, it seems to take a lot of consistent evidence before the noisy-And network is willing to change the prior probabilities much.

Noisy And with second order probabilities

Since the influence of the priors was so strong, we sought to unbias the network using second order probabilities (Neopolitan, 1990), which are thought to be a way of representing ignorance of priors. Each rule node was given a new parent with 5 values, 0 through 4, which were given a uniform prior probability distribution (each value had a prior of 0.2). The rule's conditional probability table was $P(\text{rule}|\text{parent}=N) = 0.25*N$, where N is one of the parent's values. In other words, this network is saying that it is equally likely that the rule node could have a "prior" of either 0, 0.25, 0.5, 0.75 or 1.0

This made no difference. Assessments given to students were exactly the same as those given by 0.5 priors in the preceding, simpler noisy-And network.

A noisy Or network.

To obviate the need for priors, we tried reversing the links in the Bayesian network. Each rule became a noisy Or (Pearl, 1988), with problems as the parents. That is, if any of the problems that required a rule was answered correctly, then the rule was probably mastered.

However, this meant that the rules were conditionally independent given the evidence, which just isn't true. For example, suppose rules R1 and R2 are required by problem P, and P is incorrect. Finding out that R1 is mastered should convince you that R2 is not mastered. But because R1 is d-separated from R2 by P, they are conditionally independent in the network. Thus, evidence about R1 will not influence the probability of mastery of R2 the way it should.

Counting

In desperation, we tried an extremely simple solution. We kept a score for each rule. We incremented the score by 1 when a problem requiring the rule is correct. We decremented the score by 1/N when a problem was incorrect and required N rules to solve it correctly. The intuition is that if the problem is correct, then the rule must be mastered. If it is incorrect, then at least one of the N rules required by it is incorrect, but since we don't know if it is this one, we blame each rule equally.

Because rules participate in differing numbers of problems, the ranges of the scores can vary widely. Each rule needs a threshold that is unique to that rule. If the rule's score is above the threshold, the rule is predicted to be mastered.

In order to find values for all 66 thresholds, we used the artificial students to generate scores on rules. For each rule, we adjusted its threshold to maximize the rule's utility.

We discovered that the best threshold was almost always the minimal possible score for that rule. That is, a rule was considered mastered if any of the problems it participated in were correct. This means that the counting technique is working exactly like the noisy Or, without the noise. Therefore, it also misrepresents the dependencies.

A success

Of all the methods we tried, the noisy-And seemed the most plausible since it approximated the causality and it represented the conditional dependencies correctly. The only problem was that most of the rules were overly sensitive to their prior probabilities. In fact, they had posterior probabilities that were within .05 of their prior probabilities.

We examined the 20 rules where priors were not affecting the posteriors. In all cases, there was at least one problem that required either only that rule, or that rule and one other. That is, there was at least one problem where the rule was used in isolation (or nearly in isolation). Conversely, all rules that were used in isolation (that is, one of the problems they appeared in had only one or two required rules) were relatively unaffected by the priors. In Figure 1, rules 3 and 5 are "used in isolation" because they are used in problem 3, which has only two rules required for correctly solving it.

Finding that rules used in isolation were insensitive to the priors suggested that it was the test that was responsible for lack of sensitivity, and not the assessment method itself. This renewed our faith in the noisy And method.

When the noisy And apportions credit and blame, it usually spreads them thinly among the rules, so that it hardly moves the rules' probabilities from their priors. This suggests looking at the difference (change) in probability as a way to compensate for the small amount of credit/blame being handed out.

To implement this key insight, we modified the assessment function by giving each rule a threshold that was just above the prior. If the rule's posterior is above the threshold, then the rule is predicted to be mastered. The intuition is that in such cases, the rule is receiving credit (albeit not much) for the student's correct answers, and is

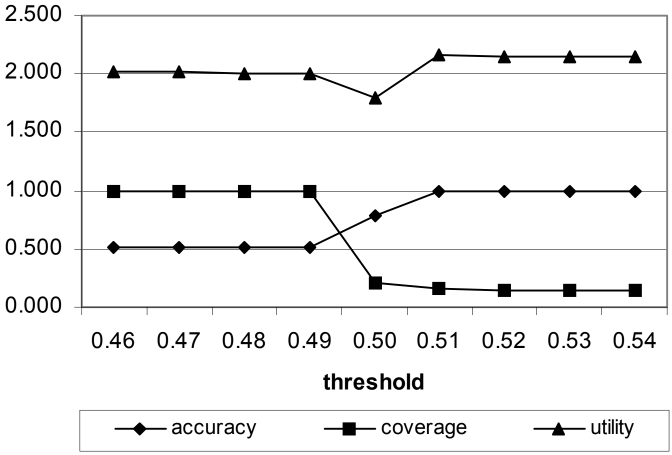

Fig. 2. Choosing a threshold to maximize utility

thus probably known. Because the thresholds are functions of the priors, it shouldn't make any difference what priors are used. Thus, the predictions should be independent of the priors.

Setting the thresholds

For each rule, we tried varying the threshold slightly around the prior. Figure 2 shows the result of varying the threshold on a typical rule (not one of the ones that appears in isolation) from –0.04 to +0.04 in steps of 0.01 about the prior, which was 0.5 in this case. As can be seen, the coverage varies from 1 to 0 while the accuracy varies from 0.5 to 1. Utility, which was defined as coverage + 2*accuracy, peaks at 0.51, so that is the threshold chosen.

Notice that plateau in accuracy. If we tried to use accuracy alone in order to select a threshold, all values for the threshold above 0.51 would be tied. However, by adding in a little bit of coverage to the utility function, we can make it peak. For this particular rule, if utility is defined as coverage + M*accuracy and M>1.8, then there is a peak at 0.51. The peak is at 0.50 if 1.6<M<1.8. There is no peak in utility if M<1.6. Instead, there is a plateau from 0 to 0.49. Since every rule's curve was similar to this one, values for M of 2 or more (i.e., accuracy is twice as important as coverage) caused the peak in utility to be slightly above the prior. Values of M between 1 and 2 would cause utility to peak at the prior itself, whereas values of M less than 1 (i.e., coverage is more important than accuracy) yielded plateaus rather than peaks. We defined utility with M=2 since that tended to maximize accuracy without affecting coverage very much (see Figure 2).

The rules that appeared in isolation had flat curves for accuracy and coverage in the vicinity of the prior, so we gave them thresholds that were the priors themselves. Their accuracy and coverage were both close to 1.

Setting the prior, slip and guess parameters

The new noisy And has three global parameters, namely the prior, the slip and the guess. To find values for these parameters, we searched the parameter space for values that would maximize utility given an initial set of 201 artificial students.

The procedure was to select values for the 3 global parameters, run the noisy And, collect posteriors on each rule for each student, select rule thresholds that maximized utility for that rule, then sum the overall utility.

We found that prior = 0.5, guess = 0.3 and slip = 0.01 yield maximal values. However, we also found that utility was relatively insensitive to all three values.

Varying the set of artificial students

However, we worried that these results could be idiosyncratic to the particular set of 201 artificial students used to produce them. Therefore, we generated new sets of artificial students, and compared the utilities obtained by assessing them to the utilities obtained from the original 201 students.

The first set was 400 artificial students generated with the same values for the guess and slip parameters. Predicted levels of mastery were generated for each of the 400 student's rules, and the prediction's accuracy, coverage and utility were calculated. The mean utility for the 400-student set was 2.465, and the mean utility for the original 201-student set was 2.372. Although statistically reliable (T-test, $p<.0001$) due to the large number of data points, the difference in means was only 4%. The next set was 201 artificial students generated by raising the value of the slip parameter to 0.1. Again, there was a small (1%) but reliable difference ($p < .01$). Similarly, lowering guess to 0.1 and keeping slip at 0.1 generated a small (1%) but reliable difference ($p < .001$).

Thus, the results seem relatively insensitive to the specifics of the set of artificial students used for evaluation. Regardless of how the artificial students use to evaluate the test scoring method were generated, the test scoring method yielded high accuracy and moderate coverage.

Discussion

The problem discussed here is to infer the posterior probability of mastery of a set of N rules given a test with M items, where each item has been scored as correct or incorrect. We are given a Q-matrix, which indicates which rules are required for correct answers to each problem.

Since we do not have prior probabilities of mastery, it seemed initially that we could not use the Noisy-and approach. However, we were unable to find a method that both represented the conditional dependencies that exist and did not require priors until we noticed that the posterior probabilities of the noisy And did vary in response to the data, but just did not move far from the priors. The network was passing out credit and blame rather thinly. Thus, we used priors, but looked for small changes away from the prior. If a rule's posterior is above the prior by an amount specified by a rule-specific threshold, then the rule was interpreted as being mastered. Artificial students were used to select thresholds that maximized accuracy and coverage.

We discovered that there were essentially two kinds of rules on the test. The rules used in isolation were accurately assessed by virtually any combination of parameters. For the other rules, the coverage and accuracy traded off, as shown in Figure 2. Thresholds were picked so that the accuracy was always over 90% and often close to 100%, but the coverage was around 25%.

According to this evaluation, if the assessment says that the student has mastered a rule, then it is probably right. On the other hand, if it says that the student has not mastered a rule, then it will often be wrong. The latter should thus be interpreted as "insufficient evidence to conclude mastery exists" instead of "non mastery." We experimented with Bayesian network-based methods for discriminating between lack of evidence and evidence of non-mastery, but failed to find one that worked. This would be a good topic for future work.

Bayesian methods have always had difficulty working with applications where priors are not available. As textbook authors (e.g., Neopolitan, 1990) are fond of pointing out, a prior of 0.5 on a true/false variable does not represent ignorance, but instead represents certain knowledge that the variable in is true 50% of the time in the population. Second order probabilities (Neopolitan, 1990), Dempster-Shafer (Pearl, 1988; Neopolitan, 1990) and other techniques have been used to represent ignorance, but they have drawbacks of their own. As far as we know, using the difference between priors and posteriors has never been tried before as a means of making predictions insensitive to the priors. It only applies when the outcome nodes (the ones whose posteriors are important) are the same as the nodes whose priors are unknown. However, such networks are common in diagnostic applications such as student modeling. Empirical testing with artificial data suggests that our technique is successful for our application, but more work is needed to understand the properties of this solution.

We embarked on this research in order to use a diagnostic test to provide priors for Andes' Bayesian network. Having evaluated the proposed diagnostic testing method and finding that it produces accurate results with artificial students, we have analyzed the test data from the 178 human students, found out which rules were mastered by which students, and thus obtained the priors that Andes' needs. Ironically, the method that we proposed here could be used with Andes itself, thus removing the need for priors. Our next step may be to try this and evaluate it with either artificial students.

Acknowledgements

This research is supported by ARPA's Computer Aided Education and Training Initiative under grant N660001-95-C-8367, by ONR's Cognitive Science Division under grant N00014-96-1-0260 and by AFOSR's Artificial Intelligence Division under grant F49620-96-1-0180. We would like to thank the rest of the Andes group for their help.

References

Conati, C., Gertner, A., VanLehn, K. & Druzdzel, M. (1997). On-line student modeling for coached problem solving using Bayesian networks. In A. Jameson, C. Paris & C. Tasso, *User Modeling: Proceedings of the Sixth International conference, UM97*. New York: Springer Wein.

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*. Hillsdale, NJ: Erlbaum.

Neopolitan, R. E. (1990). *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. New York: Wiley.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan-Kaufman.

Samejima, F. (1995). A cognitive diagnosis method using latent trait models: Competency space approach and its relationship with DiBello and Stout's unified cognitive-psychometric diagnosis model. In P. D. Nichols, S. F. Chipman & R. L. Brennan (Eds.) *Cognitively Diagnostic Assessment*. Hillsdale, NJ: Erlbaum.

Tatsuoka, K. K. (1990). Toward an integration of item-response theory and cognitive error diagnoses. In N. Fredericksen, R. L. Glaser, A. M. Lesgold & M. G. Shafto (Eds), *Diagnostic Monitoring of Skill and Knowledge Acquisition*. Hillsdale, NJ: Erlbaum.

Tatsuoka, K. K. (1995). Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition and classification approach. In P. D. Nichols, S. F. Chipman & R. L. Brennan (Eds.) *Cognitively Diagnostic Assessment*. Hillsdale, NJ: Erlbaum.

VanLehn, K. (1996). Conceptual and meta learning during coached problem solving. In C. Frasson, G. Gauthier and A. Lesgold (Eds.) *ITS96: Proceeding of the Third International Conference on Intelligent Tutoring Systems*. New York: Springer-Verlag.