Pre-publication draft of a paper which appeared in The Proceedings of ASCILITE '91, pp 367 - 376

Modelling Elementary Subtraction: Intelligent Warfare Against Bugs

Mark A. P. Kuzmycz and Geoffrey I. Webb

Department of Computing and Mathematics, Deakin University

Abstract

This paper discusses an intelligent system .that uses Input/ Output Cognitive Modelling (IOCM) techniques to form a model of the student. The paper describes FBM, an IOCM system that uses features to represent the inputs and outputs of the tasks being presented to the student and forms a relationship which describes in essence the knowledge the student has in the domain. Also presented is ASPMoRe, an intelligent tool that takes the model of the student and adapts the lesson to both refine the model and give the student practice in weak areas of his knowledge. Results have shown that the system can be an effective tool for educational purposes.

Keywords

Student Modelling, Subtraction, Intelligent Tutoring Systems, Diagnostic modelling,

1 Introduction

The Subtraction Modeler is an intelligent educational system that develops detailed models of student's elementary subtraction ability. This system is distinguished from previous systems for modelling subtraction skills by its low computational overheads, ability to generate individualized tests that home in on student's difficulties and by not using a library of possible student errors.

2 Feature Based Modelling.

Most previous approaches to cognitive modelling have employed structural modelling techniques. Structural modelling systems attempt to model the internal cognition of the student. Structural modelling systems include ACM (Langley. 1984), Buggy (Burton and Brown, 1978; Brown and VanLehn, 1980), ACT (Anderson, 1989, 1990) and Genetic Graphs (Goldstein, 1984) to name a few. In contrast to this approach, IOCM models the cognitive system in terms of the relationship between the inputs and the outputs of the cognitive system. Feature Based Modelling is an IOCM methodology that uses features to represent the inputs and outputs of a cognitive system. (Webb, 1989a, 1989b; Kuzmycz, 1990), In general, the model construction time of IOCM is lower than that of structural methods.

2.1 Some properties of FBM

A major advantage of Feature Based Modelling is its ability to handle multiple errors, distinct and unrelated misconceptions that are held simultaneously and can interact, making it difficult to identify each individual error. Multiple errors may be considered as a series of buggy procedures. Each buggy procedure represents one error. Also, errors are mutually exclusive as two buggy procedures cannot be executed at the same time. Multiple errors occur when there is an impasse (Brown and VanLehn, 1980) in an erroneous procedure or there is an impasse that the buggy procedure does not cover and a student uses a buggy repair to bypass the impasse. Impasses are related to the attributes in a column (task features) (Webb, 1991). Therefore, it is expected that when multiple errors are observed in the students answers there are a set of attributes (task features) that distinguish the errors. It is these features that trigger the student to choose between several buggy procedures.

In the example below described by Burton and Brown (1978) a student is displaying multiple errors for subtraction and it can be considered as having at least two buggy procedures.

In the below example, the contextual reasons for the firing of the second buggy procedure are explained in its description and so it is self explanatory in its origins. However, in general, this will not be the case with multiple errors, as most errors' descriptions are context independent or ambiguous. For example, if the student exhibited the 2 errors of MOVE/OVER/ZERO/BORROW and STOPS/BORROW/AT/ZERO.

$$\begin{array}{r} 113 \\ -32 \\ \hline 121 \end{array} \qquad \begin{array}{r} 104 \\ -33 \\ \hline 101 \end{array}$$

Common multiple bug SMALLER/FROM/LARGER and 0-N=0.

First, if the minuend is smaller than the subtrahend then the student writes the value of subtracting the minuend from the subtrahend in the result.

Second, if the subtrahend zero then the student writes a zero in the subtrahend.

The observed reason for the student in the above example to choose method one over that of method two is that the minuend is zero. Identifying the context in which the errors occur is natural for Feature Based Modelling. In the above example Feature Based Modelling (FBM) would use the task feature "The Minuend is Zero" in the association (Webb, 1991) to identify a change in the procedure. In other words when there are multiple errors the system knows under what circumstances the error occurs. This enables the tutor (human or computer) to effectively correct the bug. This property is indicative of all errors found in modelling and so what might otherwise seem to be random noise can be detected in FBM as an error. No previous cognitive modelling has offered this feature to ITS.

Noise is data that is incorrect to some degree. Such events are common in real-life applications and intelligent tutoring systems also fall foul to these problems. Noise in tutoring systems comes from several factors.

- **Copying** - students taking answers from other students.
- **Disturbances** - physical disturbances that effect concentration. For example, talking.
- **Motivation** - if the student's motivation level has dropped then the results will not necessarily reflect the student's ability
- **Transcription** - the student and/or data collector enters in the wrong values.

A useful feature of FBM is its ability to handle noise and hence still model effectively a student in a noisy environment.

3 Subtraction

Subtraction was selected as a test bed for Feature Based Modelling system, as a simple domain that contains several desirable qualities for a diagnostic system. First, although the domain is quite simple the errors that the student display are quite complex, numerous and varied. To obtain an accurate model of such a cognitive system is a challenge that would excite the best researches in the field of AI. Also, subtraction is a well researched domain and as such it makes a good testing ground for diagnostic systems as well. There are many other diagnostic systems for subtraction enabling comparison with previous systems.

In the current implementation in the subtraction domain we decided to limit ourselves to three column problems that result in a positive correct solution. (I.e. the top number is larger than the bottom number).

4 Management of Student Interactions

The system that is used to manage the interactions with students is called Automated Problem Selection for MOdel REfinement (ASPMoRe). As such it attempts to achieve two goals. First, the system will try to traverse (cover) all of the subtraction domain. Second, it will attempt to focus questions around the students' errors.

The domain of three column subtraction contains 554,400 distinct problems that may be presented to the student. In such a domain it is difficult to select a suitable set of questions that enables the student to reveal his error and assist in the models refinement. It is normal to select a set of problems for tutoring purposes but this invariably diminishes the diagnostic power of the system. Also, it can be quite difficult to select a reasonable set of problems. ASPMoRe is created to make a selection of the TASKS based on what the student had achieved previously and his needs (weaknesses).

ASPMoRe is an intelligent tool to present a student with a set of problems that is based on the student model derived from the interactions of the student. As stated previously, this system seeks to cover as much of the modelling domain as possible so as to allow the student to reveal his error(s). It is insufficient for the system to cover the domain once as the student may exhibit noise. To overcome this possibility, the system seeks to present problems that cover the domain and meet insufficient evidence requirements (Webb, 1989b) P + C >= 3. This means that there are about 192,000 problems to be presented before the modelling system considers the lesson complete. However, although this target will never be reached in practice, the system can produce accurate and useful models from very small numbers of problems. Further, this is only a current implementation limitation and may be changed dynamically to enable all problems to be presented before the next round commences. The second aspect of the system is to focus on the student's error. The system does this by detecting all the possible associations formed. Once these associations are found the least generalizations of the association are generated. A generalization is the relaxation on the number of constraints; hence the association in this case becomes more general. The least generalization is the most specific generalization. That is, there is no generalization of the association that is a specialization of the least generalized association. With the least generalizations the system then seeks to form problems that contain the least generalization of the association along with the sibling action feature (Webb, 1988. 1991). An intuitive view of the selection of the sibling action feature is that whenever the student sees a problem containing a group of features, say A&B&C, he/she responds with the action feature AF. To produce a problem where the correct solution is to answer AF would compound the association and not investigate the problem or give the student the chance to improve. Whereas if a problem is chosen for which the answer ~AF, where ~AF is any sibling feature of AF, is required to achieve the correct answer. If the student answers the problem with the incorrect feature, AF, then the erroneous association is supported. Conversely, if he answers with the correct feature, ~AF, then the association will be refined.

For Each Association, As, in the Student Model

Push all least generalizations of As onto the stack

End For

Push a maximum of 200 most general insufficient information subsets onto stack

WHILE Subsets in stack

If association **THEN**

FORM a problem containing the Task Features and the sibling Action Feature

ELSE

FoRM several problems containing the Task Features and each Action Feature

End While

The formation of a problem can be found in previous work (Kuzmycz, 1990) and uses a K-least specialized machine learning algorithm to create a set of compact problems that cover the least general associations with the fewest number of problems possible. The current system uses the suggested improvements therein to improve performance significantly (60 fold speed up).

5 Evaluation techniques

Several test of the system's performance have been applied.

First, the system is tested on how well it can predict the student's answer. We have applied the student models formed after viewing a sequence of te sts to the student's performance on the next test. Each test contains 32 subtraction problems. Thus, we have taken a rather game approach of predicting as many as 32 problems in advance. This in itself places large demands on a modelling system as it has been found that students change their buggy procedures, and since there is one week between tests this is a most likely event. However, it does enable us to see how well the system can make long range forecasts (predictions) and also enable us to make a study on the stability of the students' bugs.

Feature Based modelling uses associations to represent the students' knowledge in the domain and hence we can use these associations to make predictions about each student. In the system there are two types of associations. Erroneous associations are associations that may either give a correct answer or incorrect answer. The other associations, appropriate associations (Webb, 1991) tell the system two things.

- The student is expected to answer with the action feature when the subset is contained in the problem.
- The student is expected to get the feature right in the context of the problem when it is selected.

The reason for making such distinctions is that for tutoring we are often interested primarily in erroneous associations (errors). However, to predict the students' answers we need both associations. To make predictions about a student we need to use heuristics to ensure that the predictions are both correct and honest. In general, the heuristic is needed for domains which have not been completely explored, as it is often the case in the subtraction domain. In such domains there may exist an association that has never been used incorrectly and under some circumstances will not give a correct answer when applied to a problem. Using the knowledge about the association we can make the predictor ignore any action feature that does not give rise to a correct solution, hence, making an acceptable prediction of the student. Intuitively it makes little sense to predic t an action feature that gives an incorrect answer if we know that the student never makes an error when the subset is found in the problem. After all, a tutoring system does not use an appropriate association as its goal for teaching, so why should the predictor use it if it will not give the correct answer. An example is that if a student has no 'buggy' procedures and accidentally enters the wrong number (noise) and the system states, ‘yes I predicted this’. Such a prediction is unacceptable when the system can make distinctions from its’ expectations (if it expects the student to get the problem right or wrong).

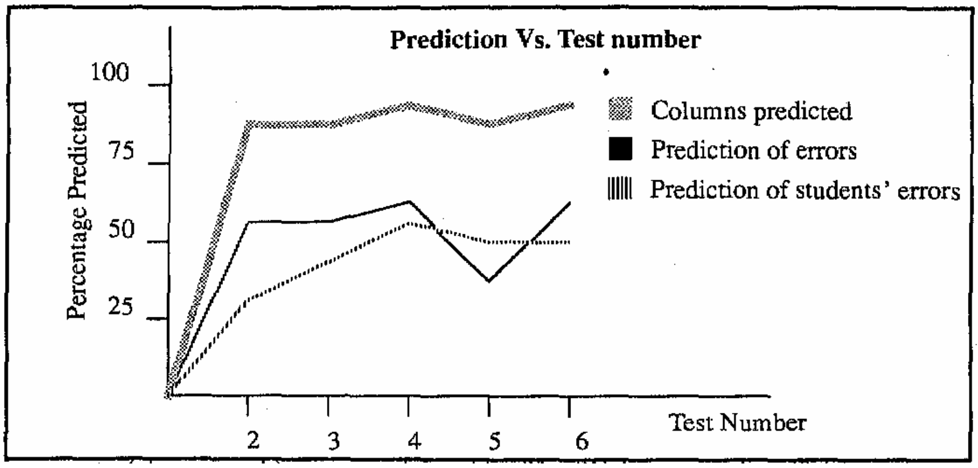
```
Init Model
FOR I = First problem to Last Problem
        FOR subset = Subset(i)(First) to Subset(i)(Last)
          IF Subset(I) = = Association THEN
            IF Association = =Erroneous THEN
                ADD Action feature to predicted set
          ELSE
                IF Action = = CORRECT THEN
                   ADD Action feature to predicted set
                END IF
            END IF
          END IF
   END FOR
   Calculate expected values from features in predicted set.
END FOR
```

The second test is an attempt to determine the percentage of errors that the system can handle. The method that we have chosen is first run the data through the modelling system then pass it through the algorithm below. This is equivalent to the evaluation method used by Burton & Brown (1978) for evaluation of the BUGGY system, and thus, enables direct comparison of results. Of course there is always the possibility that we encounter bugs that other researchers haven’t and vice versa.

```
Init Model
        FOR I = First problem to Last Problem
          IF Answer(I) = INCORRECT THEN wrong := wrong + 1;
            FOR each Action feature attributed by the student
                IF Action Feature = ASSOCIATION of Subset(I) THEN cover := cover +1
                END IF
            END FOR
          END IF
        END FOR
```

Both of these evaluation techniques are subject to noise. This causes problems since the modelling system can handle noise but the evaluators cannot. So it should be taken into account when evaluating the results presented.

6 Results

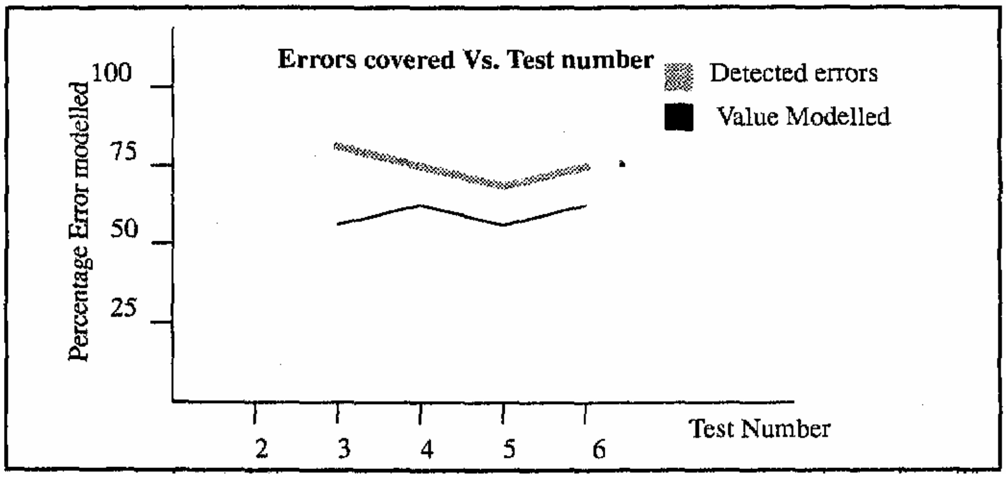


The results obtained from predicting the students' one week before the next round of tests (above graph) are very favourable., The "Columns Predicted" line gives an indication of the system's ability to predict the value of the column regardless of whether the column is wrong or right. The "Prediction of Errors" line is the percentage of errors that the system predicted and corresponded to the student's answer. The "Prediction of Students' Error" tells us the percentage of the student's errors the system predicted. Note, that the graph represents the average of all the students that participated. Also, between test 4 and test 5 the students were given some instruction in subtraction by their teachers based on the teacher assessment of the errors in test 4. Thus, students may be expected to have remedied some of their bugs before test 5. The results of the predictions indicate that 52% of the system's predicted column errors are actually errors exhibited by the students (Prediction of errors) and that the system successfully predicts 46% of the students' errors (Prediction of students' errors).

When interpreting these results the reader should remember that

- We are making long term predictions.
- The problems being predicted were chosen to focus on the areas that he system had identified as needing improvement.
- Relevant features were not included in the domain model.
- Noise such as slips and lapses of concentration can be expected in the students' performances.
- Previous studies in this domain have found that student's errors change over time (Brown and VanLehn, 1980; VanLehn, 1986).

Graph two (below) represents the percentage (cover) of students' errors that the system is able to represent through associations. The value modelled states that if an error is detected the system is able to give the same answer as the student. The Detected Error is the ability that the system has of detecting the error.



From these results two conclusions can be drawn.

- Detected errors - Valued Modelled = percent loss due to insufficient action features
- 100- Detected errors = percent loss due to insufficient action features.

For the first equation the value is about 10%. The second is 25%. We have identified some Action Features that are needed. These are to do with column skipping and four digits in the result (Brown and VanLehn, 1980). An important Task Feature has also been discovered to be missing - the feature is "The minuend is one".

It is not only the missing features that is represented in these two graphs; there are also three other factors that contribute.

- Noise - As stated previously the system is resistant to noise however the testing methods are not. Each student on average is expected to have at least I column per round of tests as noise. This amounts to 8% of the students errors.
- Inertia - FBM was created to be noise resistant this resistance has been classified as inertia. Too little inertia and the system acts too aggressively. Too much and the system is sluggish. However, through recent testing it has been found that currently the system has a skewed inertia. That is from an erroneous association to no association takes on average 3 examples. From going to no association to an association can take from 3 examples to as many as 16 with an average of about 10. We are current undergoing research to handle the problem of the inertia.
- Test intervals - Unfortunately the time between tests is one week. During such time the students may be tutored, change their errors and soon.

7 Conclusion

The results of this preliminary research suggests that the Subtraction Modeller will prove itself as an educational tool that can be used in intelligent tutoring systems to assist students in learning subtraction skills. It does this by homing in on the student's difficulties without the need for an. error library and is still reasonably effective in long term predictions enabling intricate planning strategies (AI techniques) for which intelligent tutoring systems can take advantage.

References

[1] Anderson, J. R., Boyle, C. F., Corbett, A. T., Lewis, M. W. (1990) Cognitive Modeling and Intelligent Tutoring. *Artificial Intelligence*, Vol 42, pp. 7-49.

[2] Anderson, J. R., Conrad, F. G., Corbett, A. T. (1989) Skill Acquisition and the LISP Tutor. *Cognitive Science*, Vol 13, pp. 467-505.

[3] Brown, J.S., Burton, R. R. (1978) Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. *Cognitive Science*, Vol 2. pp. 155-192.

[4] Brown, J.S., VanLehn, K. (1980) Repair Theory: A Generative Theory of Bugs in Procedural Skills. *Cognitive Science*, Vol 4, pp. 379-426.

[5] Goldstein, I. P. (1979) The genetic graph: a representation for the evolution of procedural knowledge. *International J, Man-Machine Studies*, Vol 11(1979), pp.51-77.

[6] Kuzmycz, M. A. P. (1990) Sage: An Automated Approach to the Refinement of Student Models. *Honours Thesis*, Department of Computing and Mathematics, Deakin University.

[7] Langley, P., Ohlsson, S., Sage, S. (1984) A Machine Learning Approach to Student Modelling. *Technical Report*, The Robotics Institute, Carnegie-Mellon University.

[8] VanLehn, K. (1986) Arithmetic Procedures are Induced from Examples. In *Conceptual and Procedural Knowledge: The Case of Mathematics*, Hillsdale, NJ: Erlbaum. pp. 133-179.

[9] Webb, G. I. (1988) A knowledge based approach to computer-aided learning. *International J. Man-Machine Studies*, Vol 29. pp. 257-285.

[10] Webb, G. I. (1989a) Feature-Based Cognitive Diagnosis. *Technical Report*, Department of Computing and Mathematics, Deakin University.

[11] Webb, G. I. (1989b) A machine learning approach to student modelling. *Proceedings of the Third Australian Joint Conference on Artificial Intelligence*, Melbourne, pp. 195-205.

[12] Webb, G. I. (1991) Inside the Unification Tutor: The Architecture of an Intelligent Educational System. In The *Proceedings of ASCILITE 1991*.