

Evaluation of Feature Based Modelling in Subtraction

Mark Kuzmycz
Geoffrey I. Webb

*Department of Computing and Mathematics, Deakin University
Geelong, Australia 3217*

Abstract.

One aim of intelligent tutoring systems is to tailor lessons to each individual student's needs. To do this a tutoring system requires a model of the student's knowledge. Cognitive modelling aims to produce a detailed explanation of the student's progress. Feature Based Modelling forms a cognitive model of the student by creating aspects of problem descriptions and of students' responses. This paper will discuss Feature Based Modelling and show the results of an evaluation carried out in the domain of elemental subtraction.

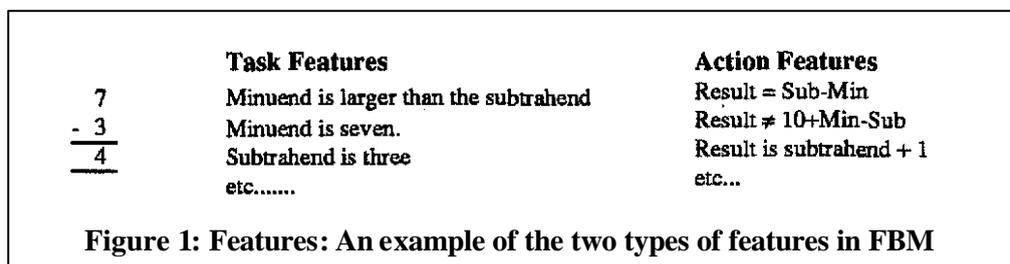
1 Introduction

Feature Based Modelling (FBM) is a student modelling approach that represents the knowledge of the student through the use of Task Features and Action Features to produce a set of relationships. The modelling system is able to handle multiple errors, concept migration and noise and does not require the use of a bug library.

This paper presents FBM with emphasis on the methods it employs to handle error migration and noise. Also presented are results of the evaluation of Feature Based Modelling in the domain of elementary subtraction. These results demonstrate credible performance in an environment that is rich in noise, multiple errors and concept migrations.

2 Feature Based Modelling

FBM is a machine learning approach to student modelling [13] that uses a set of relationships to capture the student's knowledge. The key element of FBM's formalism is the feature which represents a property of a task or of a student's response [11]. There are two types of features. Task Features represent the attributes of the task presented to the student (figure 1). Action features represent the attributes of the response.



The machine learning process creates a set of relationships between the Task Features and the Action Features [11, 12, 14]. Each of these relationships, known as associations, states that whenever all of the task features in the association are found in the task the student will act in a manner described by the single Action Feature. An association is represented in the form of TF1, TF2, TF3 ... and TFn \rightarrow AF. FBM distinguishes the associations into two classes - erroneous associations and ideal associations. Erroneous associations represent the student's misconceptions.

An erroneous association may give the correct answer when applied to some problems but there will be at least one problem (task) for which the association will give an incorrect answer. An ideal association when applied to any problem will always give the correct answer. This type of association represents the student's mastery in the domain.

2.1 Errors

To teach subject matter to a student it is useful to be able to address the misconceptions that a student holds in the domain being taught. A misconception is a consistent erroneous method or misunderstanding of a concept that, when applied to a problem, may cause the student to perform in an incorrect manner. By examining the actions of the student a cognitive modeler forms a model that is capable of explaining the observed behaviour of the student.

Important qualities of student modelling systems include the range of errors that the system can represent and how it handles such problems as multiple errors, error migration and noise. In FBM all relationships are precompiled and stored in memory. When a problem is evaluated the system searches through the relationships in the model and updates the relevant data in the relationships. FBM uses a statistical method that requires the following conditions to be satisfied. First, there must be sufficient evidence to be able to make the assertion that the relationship is an association. Second, the data must satisfy some association threshold. The sufficient evidence criteria used herein, $N + P > 3$, where N is the number of counter examples and P is the number of positive examples, is used to determine if there is sufficient evidence for the association to be accepted. This association threshold, $P / (P + N) \geq 0.80$, allows for 20% of the data to contain counter evidence (this allows for noise).

2.2 Multiple Errors

Throughout our investigations we have found that very few students with errors have only one error. On the contrary, they have two, three, even as many as four multiple errors held simultaneously. This phenomenon has also been observed by numerous other researchers [4, 5, 9, and 10]. Since it is common for students to have several errors the modelling system must be capable of modelling multiple errors. This requires a representation that is capable of representing more than one error at a time. DEBUGGY [5] (figure 2) attempts to handle multiple procedures by ordering the errors. FBM achieves the ability to handle multiple errors by creating separate erroneous associations for each error (figure 2). This approach to multiple errors is similar to that of ACM [9]

2.3 Error Migration.

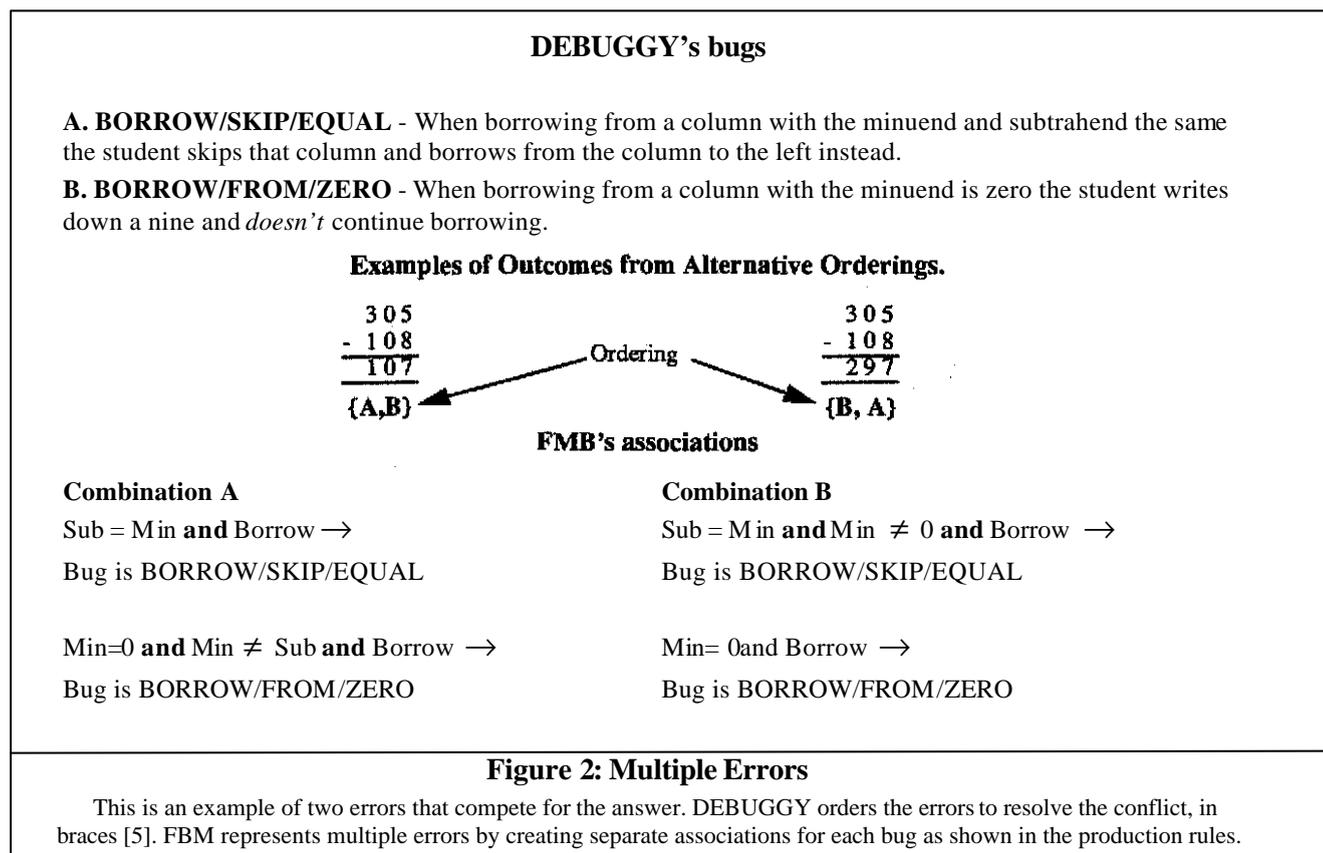
When a student is interacting with a tutor his ideas and concepts about the subject domain change with both practice and time. In order to be useful to the student, the student model must emulate (follow) the student's actions and adjust its model to reflect the student's new knowledge. Otherwise it will cease to be an accurate model of the students' understanding.

ACM handles error migration by creating a new description that explains the new error observed. It does not remove the old error from the model when that error no longer applies to the student [9]. This means that the modelling system places unwanted restriction on how the model may be explained all of the answers that the system has observed [5]. This means the new error will only be explained when and if the number of examples of the new error exceeds that of the old error. Initially FBM was similar to this as it too required the new error to outweigh the old error. This method of handling bug migration is not satisfactory for tutoring systems as the tutors require knowledge about a student's error as soon as possible so that remedial actions may be taken. ACT [1, 2] has few problems with bug migration as the model tracing theory does not attempt to find any common/deep misconceptions.

FBM solves the problems of error migration by adopting a technique that is time sensitive to the data. In consequence, if the data is shuffled into a different order then, in the presence of a concept change, a different model may be produced.

The method, known as data ageing, enables the system to place greater emphasis on newer data

than on older data. In the implementation of FBM used herein the existing data is aged by the equation $\text{Data} = \text{Data} \times \frac{9}{10}$ each time a related datum is added to the system.



2.4 Noise

Educational interactions rarely occur in ideal environments and consequently students often make seemingly random errors (noise). This noise can be due to a lack of concentration, copying other student's results, distractions or transcription errors. Noise affects student models in a number of ways. A student may seem to possess multiple errors or the student's errors may seem to be migrating.

A variety of techniques can be used to handle noise. A modelling system can settle for explaining a subset of the student's errors [5, 7, 8] or use statistical methods to identify the noisy data [9]. Although these methods handle noise to varying degrees they neglect the fact that student's strategies change with time (concept migration). What is needed is a technique that successfully integrates noise and concept migration into student modelling. FBM achieves this through the use of the model's inertia. Inertia is the model's resistance to change. It limits the change wrought by each input of new data while allowing change as more evidence is found. The inertia of the modelling system is a side-effect of the system's ability to age the data, ability to ignore small levels of counter-evidence and the association threshold.

3 Evaluation of Subtraction

Subtraction is being used to test FBM in a domain that is both simple and yet poses a challenge for modelling errors. The domain also enables the modelling system to be compared with other student modelling systems. The model of the subtraction domain consists of 28 Task Features, distinguishing 9216 problems from a maximum of 495,450 subtraction problems, and 24 Action Features which can describe numerous bugs previously found in subtraction [3, 4, 5, 10 and 16].

Each problem presented to the student contains three columns in the minuend and a minimum of 1 column and a maximum of three columns in the subtrahend.

The FBM system uses an automated problem generator (ASPMoRe) [6] to select a set of problems for the student to answer. ASPMoRe produces a different set of problems for each student depending on answers from previous questions. The problems generated by ASPMoRe target in on the students' errors to enable the modelling system to model more effectively.

A problem with evaluating/comparing the results of modelling systems is that there is no standard testing procedure. VanLehn (30-85%) and Langley (80%) [8, 9, 10] make claims that their system is capable of modelling some percentage of the observed errors in the subtraction domain.

However, this claim must be viewed with scepticism given that it is derived by determining how many errors can be generated by their system and, in view of the fact that multiple procedures can be used to generate a single result. In other words, a post hoc attempt to generate observed errors may do so using different bugs to those acquired by the student. With the evaluation of the FBM system we have tried to use a method of evaluation that can be used by any modeler as a comparison. The evaluation we used is to see how accurately a model based on the student's previous answers can predict the student's future answers.

The testing procedure used involved 31 nine to ten year old students. These students were given six rounds of tests, each round containing 32 problems. A round of tests was given to the student each week, except for round 5 which was given two weeks after round 4. Once the student answered his 32 questions the modelling system evaluated his responses. ASPMoRe, using the model, then produced a new set of 32 subtraction problems that focussed on the student's errors and covered any unseen portions of the domain in order to search for more/new errors. When the new problems were generated the student model was once again consulted to determine the student's response. In the next round of testing statistics were compiled to determine the accuracy of the system's predictions.

Predictions			ROUNDS				
			Two	Three	Four	Five	Six
Model Predicts	Columns predicted	Predicted %	98.0	99.9	99.6	99.3	99.0
		Correct %	90.0	88.1	92.4	92.2	92.1
		Incorrect %	10.0	11.9	7.6	7.8	7.94
		Multiple %	10.7 (92)	12.5 (87)	15.6 (94)	14.3 (93)	10.0 (93)
		No-prediction	2.0	0.1	0.4	0.7	1.0
	Erroneous	Number %	12.3	16.7	18.3	18.2	12.8
		Correct %	53.5	54.8	59.7	34.8	58.0
		Incorrect %	46.5	45.2	40.3	65.2	42.0
Student's Wrong	Number %	17.7	20.5	18.5	12.7	15.4	
	Correctly Predicted %	37.1	44.6	58.9	50.2	48.2	
	Incorrectly Predicted %	62.9	55.4	41.1	49.8	51.8	

Table 1: Statistics of the evaluation

Table 1 summarizes the results of these experiments. The table is divided into two sections. The first section examines the ability of the modelling system to predict the precise digit that a student

would provide for a column of a problem. In the rows entitled **columns predicted** information is contained about the percentage of columns for which the system made predictions (Predicted %), information on how accurate the predictions were (Incorrect %) and the percentage of columns for which the modeler had made more than one prediction (Multiple %) with the number in brackets representing the percentage of the multiple errors that were correct. The erroneous rows show the results columns that the system predicted the student would get wrong. This consists of the percentage of predictions that the system made (Number %) and the percentage of these predictions that were right/wrong (Correct % and Incorrect % respectively). The second section containing the rows entitled **Student's wrong** gives information on the percentage of the answers that the students got wrong (Number %), the percentage of the columns that the student got wrong and the modeler correctly predicted the students' response (Correctly Predicted %) and the percentage of columns that the student got wrong but the modelling system either didn't predict the answer or made a wrong prediction (Incorrectly Predicted %).

These tests were performed in a particularly harsh environment for any modelling system to operate. With a week between each round, a student can undergo several concept changes through interaction with other students, interactions with teachers and forgetting learned skills. It should also be noted that between rounds 4 and 5 the students' teachers tutored the students on the errors observed on round four. Under these conditions it is difficult for any modelling system to perform let alone to excel.

3.1 Analysis

The students in the domain have made few errors (Number % in Student's wrong rows) in comparison with the number of problems that the student has seen. This means that if a student makes a random error the impact will be quite large on the results obtained from the evaluation. If we assume that an average of one column for every student is noisy (1% of answers) and that the average error made by a student is 17% over the 6 rounds then the percentage of errors due to noise would be 6%. This implies that the noise would produce a worsening of results, in relation to error predictions, of about 6%.

On average (over all six rounds) 52.16% of the errors predicted by the modelling system were accurately predicted (Correct % in the Erroneous Rows). That is, of every incorrect digit that the system predicted the student entered that precise digit.

On average (over all six rounds) the modelling system correctly predicted the precise digit for 47.8% of all students' errors (Correctly Predicted %).

These results should be contrasted with a accuracy of 10% from random prediction between the ten possible values and considered in the light of the concept changes that occurred over the week between the forming of each model and the test to which it was applied; the noise that might be expected. It is particularly interesting to note the drop in accuracy that occurs between rounds four and five, presumably as a result of concept changes introduced by the teachers' tuition.

There are two other factors that affect the results obtained from the evaluation apart from the concept migration and noise. First, missing features which affect the system's ability to predict the students' errors (Correct % in the Erroneous rows) and the number of students' errors predicted (Correctly Predicted %). Second, the inertia needs tuning to allow concept migration to flow more smoothly. This factor affects the system's ability to predict all of the student's errors (Correctly Predicted %).

4 Conclusion

Feature Based Modelling uses features to represent the students' knowledge in the subject domain. In this way the system is able to represent and distinguish between multiple errors. FBM uses data ageing to enable the modelling system to cope with concept migration and as a side-effect of data ageing and the association threshold the model is able to tolerate noise and still cater for concept migration. To test these abilities FBM was subjected to a harsh testing environment that contained within it concept migrations and noise. According to the data compiled, the results of predicting the student's answers show that FBM is competent at this task and hence is an effective student

modelling system. Also, we believe that we will be able to achieve better performance by adding further features to our domain model and by altering the system inertia to optimise its ability to handle bug migration and noise.

References

- [1] Anderson, J. R., Boyle, C. F., Corbett, A. T., Lewis, M. W. **Cognitive Modeling and Intelligent Tutoring**. *Artificial Intelligence*, 1990, Vol 42, pp. 7-49.
- [2] Anderson, J. R., Conrad, F. G., Corbett, A. T. **Skill Acquisition and the LISP Tutor**. *Cognitive Science*, 1989. Vol 13. pp. 467-505.
- [3] Brown, J. S. and Burton, R. R. **Diagnostic Models for Procedural Bugs in Basic Mathematical Skills**. *Cognitive Science*, 1978, 2, 155-192.
- [4] Brown, J. S. and VanLehn, K. **Repair Theory: A Generative Theory of Bugs in Procedural Skills**. *Cognitive Science*, 1980, 4, 427-433.
- [5] Burton R. **Diagnosing bugs in a simple procedural skill**. In D. Sleeman and J. S. Brown (Eds.), *Intelligent Tutoring Systems*. Academic Press, London, 1982.
- [6] Kuzmycz, M. A. and Webb, G.I. **Modelling Elementary Subtraction: Intelligent warfare against bugs**. In *Proceedings of the Fourth Australian Society for Computers in Learning in Tertiary Education Conference*, Launceston, pp. 367-376.
- [7] Langley, P. and Ohlsson, S. **Automated Cognitive Modelling**. In *Proceedings of the National Conference on Artificial Intelligence*, 1984.
- [8] Langley, P., Ohlsson, S. and Sage, S. **A Machine Learning Approach to Student Modelling**. *Technical Report CMJJ-RI-TR-84-7*, The Robotics Institute, Carnegie Mellon University, 1984.
- [9] Langley, P., Wogulis, J. and Ohlsson, S. **Rules and Principles in Cognitive Diagnosis**. In N. Fredericksen, R. Glaser, A. LesGold, M. G. Shafto (Eds.) *Diagnostic Monitoring of Skill and Knowledge Acquisition*, Hillsdale, NJ: Hove and London, 1990.
- [10] VanLehn, K. **Arithmetic Procedures are induced from Examples**. In J. Hiebert (Ed.) *Conceptual and Procedural Knowledge: The Case of Mathematics*, Hillsdale, NJ: Erlbaum, 1986, 133-179.
- [11] Webb, G. I. **A knowledge based approach to computer-aided learning**. In *International Journal of Man-Machine Studies*, 1988, Vol 29, pp. 257-285