Pre-publication draft of paper accepted for publication in the Proceedings of the International Conference on Advanced Research on Computers in Education, Tokyo, Japan, 1990, pp 81 - 88

EDUCATIONAL EVALUATION OF FEATURE-BASED MODELLING IN A PROBLEM SOLVING DOMAIN

Geoffrey I. Webb, Department of Computing and Mathematics, Deakin University, Geelong, Australia 3217
Geoff Cumming, Department of Psychology, La Trobe University, Bundoora, Australia 3083
Thomas J. Richards and Kwok-Keung Yum, Department of Computer Science, La Trobe University, Bundoora, Australia 3083

This research has been supported by the Australian Research Council

Abstract

Feature-Based Modelling is a machine learning based cognitive modelling methodology. An intelligent educational system has been implemented, for the purpose of evaluating the methodology, which helps students learn about the unification of terms from the Prolog programming language. The system has been used by Third Year Computer Science students at La Trobe University during September 1989. Students were randomly allocated to an Experimental condition, in which FBM modelling was used to select tasks, and give extra comments, or to a Control condition in which similar tasks and comments were given, but without FBM tailoring to the individual. Ratings of task appropriateness, and comment usefulness, were collected on-line as the students worked with the tutor; overall ratings were obtained by questionnaire at the end; and semester exam results were examined. Despite the fact that only a minority of students showed sufficient misunderstanding for FBM to have potential value, of the ten comparisons chat relate most directly to the aims of the Tutor, while in no case reaching significance, seven were in favour of the Tutor, and only two against. These preliminary results are very encouraging for the FBM principles of the Tutor.

1 INTRODUCTION

Feature Based Modelling (FBM) is an approach to cognitive modelling that does not require prior identification of possible cognitive bugs, and which allows students to adopt their own approaches to problem solving in a domain. FBM has been successfully implemented in the context of the DABIS knowledge-based educational system [1] and a piano scale tutoring system [2]. However, it has not previously been implemented in a problem solving domain. Nor has it previously received extensive formal evaluation.

The Unification Tutor has been developed with the aim of rectifying both of these deficiencies. It operates in the domain of the unification of Prolog terms, a simple, yet not trivial, problem solving domain.

Unification is a key procedure in the implementation of Prolog [3]. Two Prolog terms *unify* if there is a single *substitution* that, when applied to each, provides the same result. For example, x(X, y) unifies with x(Y, Y) because, when applied to either term, the substitution {X=y, Y=y} produces the term x(y, y). (We adopt the usual Prolog convention of using upper case letters to signify variables.) There will typically be many unifiers for any two terms that unify. The unifiers of interest are the *most general unifiers—those* that assign values to the least number of variables.

FBM applies attribute-value machine learning to cognitive modelling. Most previous cognitive modelling systems have sought to model the internal knowledge structures, cognitive operators and strategies of a student [4, 5, 6, 7]. By contrast, FBM models the student's cognitive system as a whole, seeking to establish the relationships that hold between the inputs and the outputs of the student's cognitive system. The features of the tasks on which a student is engaged are related to the actions that the student performs while engaged on those tasks. Each task is described by a set of *task features*. Each action is described by a set of *action features*.

FBM seeks to develop a model chat can predict, for any set of task features, the action features that will describe the students actions. This model is recorded as a set of *associations.* Each association relates a set of task features to a single action feature. It predicts that, when engaged in a task to which all of the task features apply, the student's actions will exhibit the given action feature. Means for detecting associations have been described previously [8].

The associations of greatest interest are *erroneous associations* - associations that are not appropriate. For example, an association between the task feature **Terms_different** and the action feature **Do_not_unify** (predicting that the student when confronted with. two terms that are not identical will state that they do not unify) is inappropriate because some terms that are different do unify. By contrast, an association between the task features **Terms_different** and **Terms_do_not_contain_variables** and the action feature **Do_not_unify** predicting that the student when confronted with two terms that are different and do not contain variables will state that they do not unify) is appropriate because, in fact, all such terms do not unify.

2 THE UNIFICATION TUTOR

The Unification Tutor is an Intelligent Educational System, implemented in C, for teaching the unification of terms from the Prolog programming language. Through its use of FBM the Unificatio n Tutor is able to perform detailed evaluation of each individual's comprehension of the domain without needing to enforce a particular viewpoint of the domain [9].

The Unification Tutor makes use of 22 task features and 14 action features. These are described elsewhere [10]. Table 1 shows some tasks each with a small selection from its task features. Table 2 shows some tasks, a student's response to each of those tasks and selected action features from the attributions that each response represents.

2.1 Task Selection

Tasks are selected by choosing a combination of task features (a feature set) that is suitable for presentation to the student. In doing so the system attempts to balance three objectives –

1. to keep the tasks challenging enough to maintain student motivation;
2. to keep the tasks simple enough that the student can make sense of them; and
3. to select tasks for which the system will be able to analyse the students actions.

To achieve these ends three sets of feature sets are maintained—*Mastered*, *Current* and *Unavailable.* The first of these is the set of all feature sets that the system believes the student will be able to tackle correctly. The second is the set of all feature sets that the system believes are suitable for presentation to the student. The third set contains all remaining feature sets.

A difficulty faced by the system when evaluating the student's actions with respect to a pair of compound terms is to assign credit for those actions between the pair of compound terms as a whole and pairs of subterms within those compound terms. For example, if a student responds to the task *Unify x(X) and x(Y)* by indicating that they do not unify, does this indicate that the student thinks that the subterms *X* and *Y* do not unify, or that some other principle prevents the two compound terms from unifying even though their arguments unify?

Term 1	Term 2	Task Features
x(X, X)	x(a, Z)	Terms different, Variables_duplicated, Bindings_are_consistent.
x(X, X)	x(a, b)	Terms_different, Variables_duplicated, Bindings_are_inconsistent.

Table 1: Some tasks, and selected features from their feature sets

Term 1	Term 2	Answer	Action Features
x(X, X)	x(a, Y)	{X=a, Y=a}	Correct, Unify, Do_not_allow_multiple_bindings
x(X, X)	x(a, Z)	none	Incorrect, Do_not_unify
x(X, X)	x(a, b)	{X=a, X=b}	Incorrect, Unify, Allow_multiple_bindings
x(X, X)	x(a, b)	none	Correct, Do_not_unify

Table 2: Some tasks, responses and selected action features that describe those responses

The Unification Tutor overcomes this problem by presenting to the student only pairs of compound terms for which it believes the student can independently correctly tackle each pair of subterms. As a result, it is possible to assign credit for any inappropriate actions when tackling a pair of compound terms to aspects of the compound terms as a whole, rather than to the subterms. To this end, every pair of subterms for a pair of compound terms must have a set of features from Mastered. Thus, feature sets for pairs of compound terms are only advanced from Unavailable to Current if feature sets of the appropriate types to form the necessary subterms are in Mastered.

Between each task the student model is used to update Mastered, Current, and Unavailable. By this means the student model is used to control the sequence of tasks that the student encounters. A feature set is placed in Mastered so long as (i) it is (appropriately) associated with the action feature **Correct** and (ii) so long as none of its subsets participate in an erroneous association. Clause (i) ensures that the system has sufficient evidence to believe that the student can solve correctly tasks with the given set of features. Clause (ii) ensures that the system does not have sufficient reason to believe that faulty processes underlie the student's correct solutions.

Initially, all feature sets describing tasks involving two compound terms are placed in Unavailable and all other feature sets are placed in Current.

2.2 The Domain Model Based Adviser

The Domain Model Based Advisor (DMBA) is a sub-system that examines the appropriate action features for the current task and the features of the student's action, and provides suitable comments. If the student's answer is correct, a simple message to this effect is provided. If the student answer is inappropriate, the mismatch between the appropriate and actual action features is used to generate a comment. This comment is domain model driven. That is, it is based on domain-based assumptions about the likely cause of the type of mismatch detected; no student modelling information contributes. The principles used by the DMBA to generate comments are those used by the earlier DABIS system [11].

When the student's action is appropriate, the DMBA randomly selects and displays one of a small number of short comments indicating this fact. When the student's action is inappropriate the comments are longer and more detailed.

2.3 The Student Model Based Adviser

The Student Model Based Advisor (SMBA) is a sub-system that provides comments to the student based on the FBM model. A suitable erroneous association is selected and described to the student. The student is then exhorted to reconsider her or his approach to the type of problem described by the task features of the association. It must be emphasised that the current SMBA gives only simple, rudimentary comments. The focus at present is on the FBM-based strategy for selecting when to comment, and on what topic, rather than the sophistication of individual comments. There is considerable scope for improving the SMBA.

Two conditions must be satisfied before an erroneous association is considered suitable for comment:

1. The task features of the association must apply to the task that has just been completed and the action feature must apply to the student's action for that task. This ensures that an association will only be commenced upon when the comment will relate to the students most recent action and thus will be pertinent and salient.

2. The association must not previously have been commented upon more than once. If an association has been discussed twice and the student is persisting with the same misunderstanding then it is assumed that the comments provided are ineffectual.

```
Consider the following two terms
data(sub list, x(term), x(elt))
data(Second, Estimate, Estimate)
Enter the most general unifier for these terms or type none, help or
exit.
=>{Second=sub_list, Estimate=x(term), Estimate=x(elt)}
A substitution should never contain the same variable on the left of
more than one pair. Your answer has Estimate on the left of more than
one pair.
It appears to me that whenever you examine two terms that have a
variable appearing more than once opposite arguments that do not unify
you create two substitution pairs with the same variable on the left of
each.
Perhaps you should reconsider how you tackle such problems.
Press space to continue.
```

Figure 1: An interaction with the Unification Tutor.

2.4 Interacting with the Unification Tutor

Figure 1 shows an example interaction with a student. Each interaction starts with the display of a pair of terms to be unified. Next, the student enters an answer. The answer is analysed and a comment is provided by the DMBA. Finally, if a suitable association can be found in the student model, the SMBA provides an additional comment. In Figure 1, the student's response has been underlined. All other text is output from the Unification Tutor. The comment provided by the SMBA has been set in a box. The immediately preceding comment has been provided by the DMBA.

3 EVALUATION OF THE UNIFICATION TUTOR

3.1 Procedure

A third-year undergraduate Computer Science class of 68 students taking a course entitled Artificial Intelligence received normal lecture presentations, which included material in two successive lectures one week apart about unification in Prolog. Several days before the second of these lectures, the Unification Tutor was made available to students. It remained available for four weeks - the middle two weeks being non-lecture vacation weeks. Students were encouraged to use the Tutor, but doing so was an optional extra, to be carried out in the student's own time. It was explained to students that the Tutor is a research prototype and that they would be participating in an experimental evaluation of various ways that a tutoring system might select, organise and comment on unification tasks.

For this evaluation the Tutor was modified by the addition of rating scales designed to tap student opinion frequently during use of the system. Immediately after entering a response to a task, the student was asked:

> 'How confident are you that your answer is correct? (1 = completely unsure . . . 6 = very confident)'

After the student had typed a digit, the Tutor gave its comment on the student's response. If the response was correct, the student was asked:

> 'Did this problem help your understanding? (1 = not at all . . . 6 = very helpful)'.

If the response had been incorrect, the student was asked:

> 'How helpful was this comment? (1 = not at all . . . 6 = very helpful)'

After the student had typed a digit, the next task was presented. (Or, if an SMBA comment was to be made, it was presented here and followed by the latter question once again.)

These rating questions were designed to give fine-grained evaluation of problem selection, and the Tutor's responses to student errors, with minimum disruption to normal use of the tutoring system. Students readily accepted the rating questions, and responded promptly to them.

At the lecture given at the end of the four weeks during which the Tutor was available, all students were asked to complete a brief questionnaire asking for ratings of the tutoring system, and for open-ended comments. Four questions on unification were included on the semester exam and results of these have also been analysed.

The evaluation of the Unification Tutor consisted of a comparison between two versions of the Tutor, identical except that in the Modelling condition (MOD) the full Tutor was used, whereas in the No-Modelling condition (NOMOD) the FBM student model was not used. In NOMOD, therefore, feature sets were labelled as Mastered on the usual simple performance criterion (which is, in fact: answered correctly on at least 80% of occasions, minimum 4 trials), but without reference to the requirement that no subset of the feature set participate in an erroneous association. This latter requirement is the essence of how FBM modelling is used in the Tutor to attempt to give a student an individually tailored sequence of tasks.

At initial logon to the Tutor, students were allocated randomly by the system to the MOD or NOMOD condition, and remained in that condition for all their use of the Tutor. The lecturer, the researchers, and other staff having contact with the students were blind as to which condition any student was in until after completion of the experiment. Students were not aware of the precise nature of the experimental manipulation. It must be stressed that the difference between the MOD and NOMOD conditions is extremely subtle: the tasks set, the comments given on correct or incorrect response (with the exception that SMBA comments were never seen in the NOMOD condition), the general difficulty level, and the general way progress was made from easy to hard tasks, were all identical or very similar for the two conditions. In NOMOD the Tutor computed the FBM model, even though it was not used, to ensure that any time delays experienced by the students also did not differ between conditions.

3.2 Results

In all, 46 students used the Tutor, 21 in MOD and 25 in NOMOD. Because of an unfortunate bug in the routine setting up the experimental conditions, in fact only 12 of the NOMOD students received the full NOMOD condition as described above. The other 13, who commenced work before the bug was noticed, received a partial NOMOD condition in which feature sets were not advanced from Current to Understood if a subset participated in an association. This should have only applied to the MOD condition. However, unlike the MOD condition, the FBM model did not influence whether feature sets in Mastered were moved back to Current, and the Student Model Based Advisor was not active. The two subgroups together will be referred to as the NOMOD group. (In fact, there was no sign of any difference between these two subgroups.)

The focus of the evaluation is on two questions: (i) whether the sequence of problems presented to a student is especially effective for that student, and (ii) the usefulness of the SMBA comments. The sequence of problems will only reflect FBM modelling if and when at least one erroneous association has been formed; this is also the condition for SMBA comments possibly to be made.

It turned out that for only 11 of the 21 students in MOD was at least one erroneous association formed, so the principal data comparisons should include just these students as the 'Experimental' condition. These students were, however, rather lower-performing than average, as would be expected given the conditions for the formation of erroneous associations, The 'Comparison' group should not, therefore, be all other students, but those students in NOMOD for whom at least one erroneous association was formed; there were 13 such students.

Table 3 reports some basic comparative data, for students with at least one erroneous association, for the two main experimental conditions. Note that the differences between the group means are also given in the form of Effect Sizes. The Effect Size (ES) is simply the difference between the two group means, expressed in z-score units. That is, the difference divided by the pooled standard deviation. Cohen [12] makes the arbitrary but useful suggestion that ES values of .2, .5, and .8 should be considered as 'small', 'medium', and 'large' respectively.

Of the on-line ratings, the most central comparison for evaluation of the Tutor is that of rated value of the tasks. The data show an ES of .23 in favour of the Tutor, although the difference did not reach significance. There is also a clear suggestion that MOD students were more confident, whether or not they were in fact correct.

Turning to the questionnaire data, collected after the Tutor had been available for four weeks, of the four key questions, listed in Table 3, no significant difference was found. Two differences favoured the tutor and two did not.

It is also worth noting the mean ratings for the three questionnaire statements tapping general attitudes towards the Tutor (not listed in Table 3.) On none of these items was there any sign of a difference between the groups. Including all students, for 'using the Tutor improved my understanding of unification' the mean was 4.88; for 'the Tutor was easy to use' the mean was 5.00, and for 'computer tutors will become more and more useful in education' it was 4.97. So, despite the fact that the students were working with a research prototype, they expressed positive evaluations of the Tutor and of such computer tutors in general.

Of particular interest are the SMBA comments. These were given only to some MOD students who had an erroneous association, and only in quite specialised circumstances. In fact, only 9 students received SMBA comments, and a total of 29 such comments were made. The mean on-line rating (scale 1 - 6) of these comments was 3.88, which can be compared with the mean rating of 3.45, given by the same students, of the usual comments given to incorrect student responses. This difference favours the SMBA comments, but does not however reach significance (t(df=8)=.44, *P*= .27*)*. (This comparison gives a more specific assessment of the SMBA than does the questionnaire rating of 'Useful feedback . . .' which gave a non-significant difference in the opposite direction.)

Five questions on the Semester exam for the subject were of relevance to the study. The first of these was on logic. This did not relate directly to the subject matter covered by the Unification Tutor, but serves as a general indication of aptitude. While not reaching significance, there is a clear suggestion that the aptitude of the NOMOD group was greater. The remaining questions (2 - 5) each required solution of unification problems. The mean performance on each of these was greater for the MOD group, although in no case does it reach significance. Especially in the light of the results for Question 1, this again provides a hint that the MOD condition compares favourably to the NOMOD condition.

	MOD mean	**NOMOD mean**	**Effect Size***	**t-value**	***P*** **(two-tail)**
On-line Ratings					
(n)	11	13		(df=22)	
Mean no. items completed	66.8	83.3	-.34	-.82	.42
Mean percent correct	71.2	73.3	-1.7	-.28	.78
Mean student response time (sec)	18.3	20.1	-.25	-.60	.56
Mean confidence, correct responses (scale 1- 6)	5.43	4.18	.96	2.18	.04
Mean confidence, error responses (scale 1- 6)	4.82	3.60	.90	2.07	.05
Mean rated value of error comment (scale 1-6)	3.51	3.51	.00	.00	.99
Mean rated value of task (scale 1 - 6)	4.56	4.19	.23	.56	.58
Questionnaire Ratings (scale 1 = strongly disagree … 6 = strongly agree)					
(n)	9	10		(df=17)	
Useful feedback was provided when I made mistakes	2.89	3.60	-.44	-.95	.36
Too much background knowledge was assumed by the Tutor	3.00	2.30	.74	1.57	.14
The Tutor only presented problems when I had learnt enough to examine them	3.67	3.40	.21	.43	.67
The choice of problems matched my needs	3.56	3.20	.28	.60	.55
Exam Performance (percentages)					
(n)	8	12		(df=18)	
Question 1 (Logic)	68.8	79.3	-.50	-1.11	.28
Question 2 (Unification)	76.2	62.5	.43	.93	.36
Question 3 (Unification)	46.2	45.8	.01	.02	.98
Question 4 (Unification)	35.0	34.2	.03	.06	.95
Question 5 (Unification)	33.7	25.8	.18	.40	.69
**Note: Effect size is the difference between the means, in standard deviation units*					

Table 3. Basic comparisons for the two groups of students with erroneous associations.

4 DISCUSSION

All educational evaluation is fraught with difficulties of control, Hawthorne effects, and measurement. We elected to compare two versions of the Tutor that differed only in subtle, but crucially important ways, so that the Experimental and Control conditions would be very similar in most respects, and any differences could be assigned with some confidence to the influence of the FBM student model that constitutes the heart of the Tutor.

The general response of students to the Tutor was positive, despite it being only a research prototype, its use being an optional extra, and coming at a very busy time of the academic year, and finally that it was running on an extremely heavily-loaded computer system. The open-ended comments recorded by students on the questionnaire reflected this favourable tone. Beyond this, the most frequent specific comments were that the tasks were, on the whole, too easy, and did not progress in difficulty sufficiently quickly. (Even so, it is worth noting the overall error rate of 20.5 %.)

It was unfortunate that a bug prevented a clean MOD vs NOMOD comparison as intended. One puzzling difference emerged: the MOD students were more confident of their responses, although there appears to be little justification for this confidence, given that their answers were no more likely to be correct.

This is only a preliminary evaluation of the Tutor, with unfortunately small student numbers. Even so, on the three comparisons most directly related to the effectiveness of the tailoring of task selection to individual needs, all three favour the MOD students, i.e. they favour the Tutor. (These are the bottom on-line rating item and the two bottom questionnaire rating items in Table 3.) The one comparison that can speak directly to the value of the SMBA comments also gave a difference in favour of the Tutor. Finally, the subjects' performance on the semester exam also favoured the Tutor. These preliminary results must be regarded as very encouraging support for the FBM methodology.

REFERENCES

[1] Webb, G.I., *International Journal of Man-Machine Studies* (1988) 257.

[2] Amato, N. H. & Tsang, C. P., Student Modelling in a Scale Tutoring System, *Tech. Rep. 88/5, The University of Western Australia Department of Computer Science,* Nedlands, W.A, (1988)

[3] Sterling, L. & Shapiro, E. *The Art of Prolog,* MIT Press, Cambridge, Mass, (1986)

[4] Brown, J. S. & Burton, R. R., *Cognitive Science* (1978) 155.

[5] Sleeman, D. H., Assessing Aspects of Competence in Basic Algebra, in: Sleeman, D. H. and Brown, J. S., (eds.), *Intelligent Tutoring Systems*, Academic Press, London, (1982) pp. 185-199.

[6] Langley, P., Ohlsson, S. & Sage, S., A Machine Learning Approach to Student Modeling *Tech. Rep. CMU-RI-TR-84-7, The Robotics Institute*, Carnegie-Mellon University, (1984)

[7] Reiser, B.J., Anderson, J. R. & Farrell, R.G., Dynamic Student Modelling in an Intelligent Tutor for LISP Programming, in: *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, (1985) pp. 8-14.

[8] Webb, G.I., A Machine Learning Approach to Cognitive Modelling, in *Proceedings of the 1989 Australian Joint Conference on Artificial Intelligence,* Melbourne (1989) pp. 195-205.

[9] Wenger, E., *Artificial Intelligence and Tutoring Systems,* Morgan Kaufmann, Los Altos, CA, (1987)

[10] Webb, G. I., Cumming, C., Richards, T. J. & Yum, K-K., The Unification Tutor: An Intelligent Educational System in the Classroom, in *Proceedings of the 1989 ASCILITE Conference*, Gold Coast, (1989) pp. 408-420.

[11] Webb, G. I., The Domain Analysis Based Instruction System, in *Proceedings of the Fourth CALITE Conference*, Adelaide, (1986) pp. 295-302.

[12] Cohen, J., *Statistical Power Analysis for the Behavioral Sciences* , Academic Press, New York, (1977)