Pre-Publication draft of a paper which appeared in the Proceedings of the 1991 IJCAI Workshop: W.4 Agent Modelling for Intelligent Interaction pp 128-136

An attribute-value machine learning approach to student modelling

Geoffrey I Webb

Department of Computing and Mathematics

Deakin University

Geelong 3217

Abstract

This paper describes an application of machine learning to student modelling. Unlike previous machine learning approaches to student modelling, the new approach is based on attribute-value machine learning. In contrast to many previous approaches it is not necessary for the lesson author to identify all forms of error that may be detected or to identify the possible approaches to problem solving in the domain that may be adopted. Rather, the lesson author need only identify the relevant attributes both of the tasks to be performed by the student and of the student's actions. The values of these attributes are automatically processed by the student modeler to produce the student model.

1. Introduction

It is possible to describe the cognitive system at many levels of detail. A description in terms of neural interaction provides a very low level description. Various levels of description of symbolic processing provide intermediate levels of description. The highest level of description provides a function mapping cognitive inputs to cognitive outputs without attempting to describe the precise internal mechanisms that cause the particular mapping. Feature Based Modelling (FBM) uses attribute-value machine learning to produce a model at this latter, highest possible, level of detail.

Cognitive modelling at the level of input and output contrasts with most previous approaches to cognitive modelling which have attempted to produce a model describing the internal operation of the cognitive system (Brown & Burton, 1978; Clancey, 1987; Goldstein. 1979 Reiser, Anderson & Farrell, 1985: Sleeman, 1984; Stevens, Collins & Goldin, 1982; VanLehn, 1982.)

To appreciate the details of FBM, it is necessary first to review some of the basic principles of attribute-value machine learning.

2. Attribute-value machine learning

Attribute-value machine learning involves developing procedures for classifying objects. Those objects are described by vectors of attribute-values. A classification procedure maps vectors of attribute values onto discrete classes.

Research into attribute-value machine learning has been conducted in two contexts - the induction of decision trees (Quinlan, 1986*b*); and the induction of *class descriptions*, expressions that denote a class of objects. Michalski's (1984) Aq algorithm is an example of a machine learning algorithm that learns class descriptions- In the context of attribute-value machine learning, a class description is a partial description of a vector of attribute values that is associated with a class. Any object represented by a vector of attribute values to which the description applies is *covered* by that description. Any object covered by a class description is deemed to belong to the class with which the description is associated.

Most machine learning systems examine a set of examples, called the *training set*, in order to develop a classification procedure that correctly classifies the entire set. Some systems, however, make allowance for the possible presence of *noise* (inaccuracies) in the training set and allow the classification procedure to mis-classify some examples from the training set if there is evidence that the details of that example are inaccurate (see, for example, Quinlan, 1986*a*)

Where more than one classification procedure will adequately classify all examples, alternative classification procedures are usually evaluated on a criteria that measures the simplicity of the procedure. A more simple classification procedure is usually preferred to a more complex classification procedure.

An important observation about class descriptions is that they can be partially ordered in terms of generality (Mitchell, 1977.) Class description *A* is a *generalization* of class description *B* if *A* necessarily covers every case that *B* covers and *A* may cover cases that *B* does not cover. If *A* is a generalization of *B*, then *B* is a *specialization* of *A*.

3. Feature-Based Modelling

FBM is an attribute-value machine learning approach to cognitive modelling. It describes the inputs to the cognitive system in terms of attribute values and develops class descriptions that map those inputs onto symbolic descriptions of the cognitive system's outputs.

The attribute values necessary to describe the cognitive system's inputs are selected by the designer of the modelling system. These generally consist of a description of the task on which the subject is engaged and salient aspects of the context in which that task is being tackled. Each such attribute value is called a *task feature*. An attribute of which a task feature is a value is called a *task feature choice.*

The outputs of the cognitive system are also described as a vector of attribute values. Each of these attribute values is called an *action feature*. The attribute of which an action feature is a value is called an *action feature choice*.

The relationship between features and feature choices is described using a knowledge representation formalism called the *feature network* (Webb, 1988.)

The objective of FBM is to form for each action feature the set of all most general class descriptions that describe tasks for which the action feature will apply to the subject's actions. Each such class description is called an *association*.

An association relates a set of task features to a single action feature. It indicates that when every one of the task features is present for a task and the subject is able to produce the action feature, s/he will do so. An association of a set c task features *T* with action feature *a* is written as *T* ® *a*.

Association *A* ® *a* is more general than association *B* ® *b* if and only if a = b and there are potential tasks to which *A* applies to which *B* does not apply and no potential tasks to which *B* applies and *A* does not apply. This will always be the case when *A* is a subset of *B*.

Unlike most machine learning algorithms, FBM does not seek to develop the simplest model that correctly classifies all examples. If this were the case, it would seek to uncover the smallest set of associations such that for every example of an action feature being present during execution of a task, an association existed for that action feature whose task features were a subset of the task's features. Instead, FBM develops a model of all most general associations that are supported by the evidence. That is, every association supported by the evidence is included in the model unless there is a generalization of that association also supported by the evidence.

The model includes all most general associations so as to provide consistency over time. This is because the use of a minimal set of associations frequently results in a single example causing the formation of a new model that has little in common with the model held before consideration of that example. In the face of such frequent dramatic revisions to the cognitive model, it is extremely difficult for a system to provide consistent interactions with the subject of the model. By contrast, under the FBM approach, most revisions to the model involve generalizing or specializing a small number of associations. This leads to gradual changes in the cognitive model. Each successive state of the model relates to the previous state in a manner that directly reflects the content of the most recent example made available to the system.

As FBM uses induction to detect the associations, it is not necessary to identify in advance the bugs that a student may adopt. The student's bugs are identified at run time through the use of machine learning without reference to a library of possible bugs.

4. Noise

In developing an FBM model it is necessary to allow for the possibility of noise in the examples available to the system. Noise can be introduced by a number or mechanisms. Inattentiveness may cause a subject to perform in a manner that is not representative of her/his underlying approach to a type of task. A simple slip, such as pressing the wrong key on a keyboard, may also introduce inaccuracies.

FBM accommodates this possibility by allowing an association to be formed despite the existence of a small number of counter-examples and by requiring a minimum number of positive examples before an association can be formed. The criterion for accepting an association $T \to a$ refers to two quantities -

1. P, the number of times that the task features in T have been present and the subject could have produced action feature a; and

S, the number of times that the task features in T have been present and the subject has produced action feature a.

Current implementations of FBM accept that there is sufficient evidence for an association if $P \geq 3$ and $\frac{S}{P} \geq .8$. The first condition seeks to ensure that sufficient examples have been encountered to demonstrate that the apparent association is not the result of noise. The second condition seeks to establish that sufficient of the examples support the association while allowing some counter-examples as a result of noise. These conditions have been developed by trial and error and provide good performance in practice.

On the face of it, allowing up to 20% of the evidence relating to an association to be contra-indicative might appear likely to lead to accepting associations that are overly general. However, a number of additional measures guard against this possibility.

5. Contra-indicative associations

Assuming that there are regularities in the operation of the cognitive system being modelled, if an association $T \to a$ is overly general then there will be regularities in counter-examples that are covered by $T \to a$. If the relevant task features are available to the modelling system, these regularities will be represented in the student model by an association between a specialization of T and a feature other than a that belongs to the same feature choice as a. If such a contra-indicative association exists then $T \to a$ is rejected. As a result, a specialization of the rejected association will be accepted that does not cover the contra-indicative examples.

6. Appropriate and inappropriate associations

It is important to realize that many associations will be *appropriate*. That is, they will be associations that will be adopted by the ideal subject. For example, when modelling the cognitive system of a learner driver it would be appropriate for there to be an association between the task features that represent approaching a red traffic signal and an action feature that represents applying the brake.

By contrast, some associations will be *inappropriate*. It should be noted that an association $T \to a$ may be inappropriate even if a is appropriate to many of the tasks covered by T. For example, when modelling the cognitive system of a learner driver it would be inappropriate for there to be an association between the task features that represent approaching a traffic signal and an action feature that represents applying the brake, even though it is sometimes appropriate to apply the brake when approaching a traffic signal.

In most circumstances, inappropriate associations will be of greater interest than appropriate associations.

An association $T \rightarrow a$ is flagged as inappropriate only if the student has exhibited *a* or a task covered by *T* for which *a* was inappropriate.

Aside from allowing the system to determine which associations are appropriate and inappropriate, this measure also prevents the system from acting upon associations that are overly general as a result of having not yet encountered a counter-example. For instance, if the subject has only approached red traffic signals and has acted appropriately by applying the brake each time then the evidence to hand would support an association between approaching a traffic signal and applying the brake. However, this association would not be flagged as inappropriate as it had not been demonstrated in an inappropriate context.

7. Viewpoint independence

Most domains can be tackled from multiple *viewpoints* (Wenger. 1987.). Substantially different sets of operators and strategies can provide equally valid solutions for tasks from a single domain. Even for such a simple domain as elementary subtraction substantially different solution methods are widespread (Fawcett & Cummins, 1970.)

This has profound implications for approaches to cognitive modelling that seek to develop accurate models of the internal operation of the cognitive system. In order to determine which operators a subject applies incorrectly it is first necessary to determine which viewpoint is being applied. Otherwise, the operators that it is assumed are being applied incorrectly may bear no relationship whatsoever to those utilized by the subject. As it is not possible to directly observe the cognitive operators that a subject applies to a task, this is a problem of enormous complexity that is yet to be successfully tackled.

By contrast, FBM does not need to place the subject within a viewpoint. Nor does it need to deal with unobservable internal cognitive entities. It deals solely with observables, the input and output to the cognitive system. The effect on the relationship between the cognitive input and output of the viewpoint that the subject adopts can be evaluated without the need to identify that viewpoint.

Closely related to the need to locate a subject within a viewpoint is the requirement of many approaches to cognitive modelling for a pre-specified library of possible correct and incorrect cognitive operators and strategies. A typical example of such an approach is the bug library of the BUGGY system (Brown & Burton 1978.)

FBM has no such requirement. It is not necessary to identify the possible bugs that a subject may adopt. Rather, it is only necessary to identify in advance the aspects of cognitive input and output that are required to identify the bugs. The precise description of each bug, in the form of an association, will be automatically generated when and if required.

8. Existing implementations

FBM has been implemented in four intelligent educational systems. The first implementation is an off-line student modelling sub-system for the DABIS knowledge-based tutoring system (Webb, 1988.) Being off-line, the student model is not available to the tutoring system and so cannot be used to manage interactions with the student. The primary value of the modelling sub-system has been to demonstrate that FBM produces credible models and to provide models for use by the teacher managing the system. The major lesson to have utilized this system examines English word classes for Linguistics students.

Amato & Tsang (1988) have incorporated FBM in a piano scale tutor. Each task involves playing a scale on an electronic keyboard. Task features include the appropriate tonic, hand motion, number of octaves, touch and tone of scale. Action features describe the tonic, hand motion, number of octaves, touch and tone of the student’s attempt to play the scale. The model is used to generate advice and to select appropriate scales for the student to practice.

The English word classes lesson and the piano tutor have demonstrated the application of FBM to classification tasks and to complex skills. The third and fourth systems to be developed demonstrate the application of FBM to problem solving domains. The Unification Tutor (Webb, Cumming, Richards & Yum, 1989) examines the unification of terms from the Prolog programming language. The student is presented with a pair of Prolog terms and asked to provide a most general unifier for those terms. Task features describe the pairs of terms. The student model is used both to select tasks for the student to examine and to generate advice. In the initial implementation of the system, this advice takes the form of describing an association and exhorting the student to revise her/his approach to tasks to which the task features apply.

Enter a most general unifier for the following terms or type none, ? or exit.

```
second(value(z), u)
second(E, E)
```

=>{E=value(z) ,E=u}

It appears to me that when two terms have a variable appearing more than once opposite terms that are different you create two substitution pairs with the same variable on the left of each.

```
You should never create two substitution pairs with the same variable on
the left of each.
```

Perhaps you should reconsider how you tackle such problems.

My answer is none.

Press space to continue.

Figure 1: An interaction with the Unification Tutor

The Unification Tutor illustrates the viewpoint independence of FBM. The Unification Tutor does not incorporate assumptions about the subject's viewpoint. The only tie to a particular viewpoint is a description of an algorithm for unification that is available as part of a help facility. The modelling system does not assume that this algorithm is being applied and the feedback provided to the student in no way relates to the algorithm. Indeed, the student need not even consult the help facility and may remain quite unaware of the algorithm.

Figure 1 shows a typical interaction with the Unification Tutor. The Tutor presents two terms to be unified. The student's response is underlined. A range of associations are developed as a result of this interaction all those with a set of task features that is more general than {FUNCTORS_ARE_IDENTICAL, ALL_INDIV1DUAL_ARGUMENTS_UNIFY, ARGUMENTS_OPPOSITE_A_VARIABLE_ARE_DIFFERENT} and more specific than {ARGUMENTS_OPPOSITE_A_VARIABLE_ARE_DIFFERENT} associated with each of the task features PROVIDE_A_UNIFIER and PROVIDE_MULTIPLE_BINDINGS. The association with the most general set of task features and most specific action feature is chosen - {ARGUMENTS_OPPOSITE_A_VARIABLE_ARE_DIFFERENT} →PROVIDE_MULTIPLE_BINDINGS. Suitable student model based feedback is provided that relates to this association. Note the viewpoint independence of the interaction - no assumptions are made about the student's viewpoint of the domain.

The extremely crude current use of the student model will be upgraded in future versions of the system. One obvious measure to incorporate is to offer to demonstrate to the student methods for tackling the types of task that that an association predicts the student will tackle incorrectly.

The Unification Tutor has been used successfully in third year Computer Science courses at La Trobe University and Deakin University. Formal experimental comparisons of versions of the system that utilize the FBM model and those that do not have shown better performance for students using versions incorporating the model (Webb, Cumming. Richards & Yum, 1990.) However, the size of the groups involved and the differences in performance have been too small for the results to be statistically significant.

The fourth system to incorporate FBM is a modelling system for student's performing elementary subtraction problems (Kuzmycz, 1990.) This system is able to detect all of the most common bugs identified by Brown & Burton (1978.) The system was evaluated on a class of 23 Year Four (eight to nine year old) students. These students were given four tests at weekly intervals. After the first test, the student model was used to produce the subsequent tests with the aim of refining the model. After analysis of the first three tests, the model was used to generate predictions about the student's solutions on the fourth test. Despite this test concentrating on aspects of the student's subtraction ability for which the system could determine that its model required refinement, 97% of the system's predictions were correct.

More detailed evaluation of the Unification Tutor and the Subtraction Modeler is on going.

9. Scope and limitations

As demonstrated by the range of domains to which it has been applied, FBM has wide scope. It has been successfully applied to a classification task, a manual skill and problem solving domains.

However, a number of potential shortcomings need to be addressed.

The computational complexity of the methodology increases exponentially with the number of task features, although the increase with additional action features is only linear. While this may become a serious problem as FBM is applied to more complex domains, it has not yet proved to be a difficulty. The Unification Tutor employs 22 task features and 14 action features. During use at La Trobe University in 1989 on a heavily loaded Pyramid 90 mx the average CPU time spent on updating the student model after a task was 3.4 seconds (8.3 seconds real time.) As student modelling was performed while the student was reading domain model based comments provided by the tutor, the experienced delay as a result of modelling was minimal. Further, a re-implementation of the modelling system is in progress which is expected to improve dramatically on the computational performance of the existing system.

Another shortcoming of the current methodology is that it is slow to respond to a change in the subject's approach to a domain. Consider the situation where the student has adopted an erroneous approach to a domain and an association reflecting this approach has been formed on the basis of 100 positive examples. If the student now changes her/his approach to the domain, at least 21 negative examples will be required before the association will be rejected. More than 21 negative examples may be required as some examples of the new approach are likely to be positive examples of the association. Further, if the new approach is also erroneous and differs substantially from the previous approach, large numbers of positive associations may be required before an association reflecting the new approach can be developed.

Consider a FBM model of a learner driver who has applied the brake every one of the 100 times s/he has approached a traffic signal. An association will have been formed between approaching a traffic signal and applying the brake. S/he now revises her/his approach to driving and only applies the brake when approaching a red or amber traffic signal. If 50% of traffic signals approached are green, it will take another 52 examples before the old association is rejected.

To overcome this problem, the next implementation FBM will discount the value of older examples so that greater weight will be placed on more recent examples. As a result, the model will more closely correspond to the subject's recent approach to the domain.

10. Conclusion

In contrast I most previous approaches to cognitive modelling, FBM does not attempt to produce a model of the internal operation of the cognitive system. Instead, through the application of attribute value machine learning, it is able to produce detailed models at the level of cognitive input and output.

It would clearly be preferable to have accurate models of the internal operation of the cognitive system. Such models could support more powerful educational interactions than can be provided by FBM. However, it is not always feasible to create such models.

FBM can create accurate high level cognitive models that are computationally inexpensive to develop without the need to anticipate the forms of bug that may be encountered or the student's approach to problem solving in a domain. These models have been successfully utilized in computer based courseware in four widely differing domains.

References

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

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

Clancey, W J. (1987) *Knowledge-Based Tutoring: The GUIDON Program*. MIT Press, Cambridge, Mass.

Fawcett, H. P & Cummins, K. B. (1970). *The teaching of mathematics from counting to calculus* Merrill, Columbus Ohio

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

Kuzmycz, M. (1990) SAGE*: An Automated Approach to the Refinement of Student Models*. Honours Thesis, Deakin University, Department of Computing and Mathematics, Geelong.

Michalski, R. S. (1984) *A theory and methodology at inductive learning*. In R.S. Michalski, J.G. Carbonell & T M. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach*. Springer-Verlag, Berlin, pp. 83-129.

Mitchell, T. M. (1977) Version spaces: A candidate elimination approach to rule learning. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, pp. 305-310.

Quinlan, J R, (1986*a*) *The effect of noise on concept learning*. In R.S. Michalski, J.G. Carbonell & T M. Mitchell (Eds.) *Machine Learning: An Artificial Intelligence Approach*. Vol. 11, Morgan Kaufmann, Los Almos pp. l49-166

Quinlan, J. R. (1986*b*) Induction *of decision trees*. Machine Learning, Vol. 1, pp. 81-106.

Reiser, B..J., Anderson, J R & Farrell. R. G. (1985*) Dynamic student modelling in an intelligent tutor for Lisp programming*. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 8-14.

Sleeman, D. H. (1984) *Inferring student models for intelligent computer-aided instruction*, in R. S Michalski, J. G Carbonell & T. M. Mitchell (Eds) Machine Learning: An Artificial Intelligence Approach, Springer-Verlag, Berlin, pp 483-510.

Stevens. A.L., Collins A. & Goldin, S. E. (1982) Misconceptions *in students' understanding*. In D H. Sleeman & J S Brown (Eds.) *Intelligent Tutoring Systems*, Academic Press, London, pp. 13-24.

VanLehn, K. *Bugs are not enough: Empirical studies of bugs, impasses, and repairs in procedural skills*. Journal of Mathematical Behaviour, Vol.3 (1982), pp. 3-7

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

Webb. G. I, Cumming, G., Richards, T. J. & Yum, K-K. (1989). *The Unification Tutor An intelligent educational system in the classroom.* Proceedings of the Seventh Annual Conference of the Australian Society for Computers in Learning in. Tertiary Education, Gold Coast. pp. 408-420.

Webb, G. I., Cumming, G, Richards, T. J. & Yum, K-K. (1990). Educational *evaluation of Feature-Based Modelling in a problem solving domain*. Proceedings- of the Conference on Advanced Research on Computers in Education, Tokyo.

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