User Modeling and User-Adapted Interaction **8:** 131–152, 1998.

Using Decision Trees for Agent Modeling: Improving Prediction Performance

BARK CHEUNG CHIU and GEOFFREY I. WEBB
School of Computing and Mathematics, Deakin University, Australia.
e-mail: chiu@deakin.edu.au

(Received 6 October 1997; accepted in revised form 20 January 1997)

Abstract. A modeling system may be required to predict an agent's future actions under constraints of inadequate or contradictory relevant historical evidence. This can result in low prediction accuracy, or otherwise, low prediction rates, leaving a set of cases for which no predictions are made. A previous study that explored techniques for improving prediction rates in the context of modeling students' subtraction skills using Feature Based Modeling showed a tradeoff between prediction rate and predication accuracy. This paper presents research that aims to improve prediction rates without affecting prediction accuracy. The FBM-C4.5 agent modeling system was used in this research. However, the techniques explored are applicable to any Feature Based Modeling system, and the most effective technique developed is applicable to most agent modeling systems. The default FBM-C4.5 system models agents' competencies with a set of decision trees, trained on all historical data. Each tree predicts one particular aspect of the agent's action. Predictions from multiple trees are compared for consensus. FBM-C4.5 makes no prediction when predictions from different trees contradict one another. This strategy trades off reduced prediction rates for increased accuracy. To make predictions in the absence of consensus, three techniques have been evaluated. They include using voting, using a tree quality measure and using a leaf quality measure. An alternative technique that merges multiple decision trees into a single tree provides an advantage of producing models that are more comprehensible. However, all of these techniques demonstrated the previous encountered trade-off between rate of prediction and accuracy of prediction, albeit less pronounced. It was hypothesized that models built on more current observations would outperform models built on earlier observations. Experimental results support this hypothesis. A Dual-model system, which takes this temporal factor into account, has been evaluated. This fifth approach achieved a significant improvement in prediction rate without significantly affecting prediction accuracy.

Key words: Agent modeling, Student modeling, Inductive learning, Decision tree.

1. Introduction

A modeling system may be required to predict an agent's future actions under constraints of inadequate or contradictory relevant historical evidence. In such a situation, it faces a dilemma: to make an unreliable prediction or to make no prediction at all. This can result in low prediction accuracy, or otherwise, low prediction rates, leaving a set of cases for which no predictions are made. Contradicting or ambiguous predictions occur when a modeling system triggers two or more hypotheses that lead to incompatible results. For some applications it may

be acceptable for the modeling system to return such predictions, stating explicitly that any one of a set of possible outcomes are considered plausible. Narrowing the possibilities to a small selection may be quite satisfactory. For example, a web page recommender (such as the one described by Balabanovic, 1998) seeking to predict the page that a user will want to access next, will perform very acceptably if it is able to identify a small selection of pages that includes the desired target. In other applications, however, a specific single prediction is more valuable. For example, if an educational system is to provide feedback based on a model of a student, that feedback might be more helpful if it is based on a single hypothesis about the student, rather than on a range of credible hypotheses. This research concerns techniques for improving the quality of prediction in such contexts.

Kuzmycz (1997) studied some strategies employed in the original Feature Based Modeling system (FBM) (Webb and Kuzmycz, 1996) for resolving contradicting predictions, in the context of modeling elementary subtraction skills. He found that the strategies for reducing contradicting predictions do so at the expense of also reducing prediction accuracy. Techniques for reducing contradicting predictions without significantly degrading prediction accuracy remain an important goal.

This research attempts to solve this problem. While we have used C4.5 (Quinlan, 1993) as the induction engine within FBM, the techniques that we have developed are applicable to any FBM system, and the most effective technique, temporally divided Dual models, is applicable to many agent modeling systems. The FBM-C4.5 (Webb, Chiu and Kuzmycz, 1997) modeling system uses a set of decision trees to model an agent's knowledge. We explore alternative techniques for improving the prediction performance of a subtraction skill modeller. The first three techniques attempt direct resolution of conflicting predictions. Like Kuzmycz (1997), we found that such techniques achieve an increase in the number of predictions made at the expense of a reduction in prediction accuracy. An explanation for this effect is advanced. This provides an insight into the problem's nature and a reason for us to revise our objective: to develop techniques that directly improve a system's prediction rate without affecting prediction accuracy. Techniques that successfully achieve this objective are presented.

This paper is structured as follows. First, the FBM framework and its derived modeling systems are described. One of these, FBM-C4.5, which has been used as the agent modeller for this study, is described in more detail. Section 2 presents three techniques for resolving conflicting predictions from multiple trees, and the experimental results. An evaluative study of the Single-tree approach, which gives a unique prediction for each unseen case, is presented in Section 3. A final technique that takes models' temporal characteristics into account is reported in Section 4. The resulting Dual-model approach is shown to achieve our objective.

1.1. FEATURE BASED MODELING AND INDUCTIVE LEARNING

Input-Output Agent Modeling (IOAM) models an agent in terms of relationships between the inputs and outputs of the cognitive system. Feature Based Modeling (FBM) is a technique for IOAM. For FBM, the context in which an action is performed is characterized by a set of attribute values called context features. An agent's action is characterized by a set of attribute values called action features. For each attribute with action features as values, a model is inferred that predicts a specific action feature for any given combination of context features. For example, when modeling subtraction skills, one attribute might relate to whether the result equals the minuend minus the subtrahend, with action features $R = M - S$ and $R \neq M - S$ as values. Another might relate to whether the result equals the subtrahend minus the minuend, with action features $R = S - M$ and $R \neq S - M$ as values. These disparate models can be considered in isolation, to examine different aspects of the agent being modeled. Alternatively, the predictions of each model can be aggregated to make detailed predictions about specific behavior. In the latter mode of use, an FBM system can be seen as an ensemble of classifiers, an approach to inductive learning that has recently enjoyed considerable success in the machine learning community (Ali et al., 1994; Breiman, 1996; Chan and Stolfo, 1995; Dietterich and Bakiri, 1994; Heath et al., 1993; Kwok and Carter, 1990; Nock and Gascuel, 1995; Oliver and Hand, 1995; Schapire, 1990; Wolpert, 1992).

The first implementation of FBM used a novel induction system to form the models (Webb, 1989). FBM-C4.5 replaces this idiosyncratic induction system with the well-known C4.5 (Quinlan, 1993) machine learning system. Comparative evaluation of the original FBM and FBM-C4.5 in the domain of subtraction showed the FBM-C4.5 provided the same level of accuracy in prediction while making substantially more predictions (Webb, Chiu and Kuzmycz, 1997).

The use of inductive learning for agent modeling has been studied previously (for examples, Desmoulins and Van Labeke, 1996; Gilmore and Self, 1988). An induction based modeling system may require prohibitive resources for implementation and operation if it seeks to develop a high fidelity model of the internal operation of an agent's cognitive system. The IOAM approach allows a system to treat the operation of the cognitive system as a black box and models the operation in terms of the relationships between the inputs and outputs of the system. Once the specifications of inputs and outputs are derived, a general-purpose classifier learning algorithm can be employed as the induction engine. This contrasts with approaches that seek to model the internal cognition of an agent's cognitive system (for example, Anderson et al., 1985; Anderson et al., 1990; Baffes and Mooney, 1996; Brown and VanLehn, 1980; Brown and Burton, 1978; Corbett and Anderson, 1992; Giangrandi and Tasso, 1995; Goldstein, 1979; Hoppe, 1994; Ikeda et al., 1993; Langley and Ohlsson, 1984; Langley et al., 1990; Martin and VanLehn, 1995; Ohlsson and Langley, 1985; Sleeman, 1982; Sleeman et al., 1991; Young

and O'Shea, 1981). Alternative IOAM approaches include RMB (Kuzmycz, 1995) and FFOIL-IOAM (Chiu, et al., 1997).

1.2. AN OVERVIEW OF THE FBM-C4.5 SUBTRACTION MODELLER

The FBM-C4.5 subtraction modeller models an n-digit subtraction problem by treating it as n related tasks, each involving specification of a digit for a single column. Without losing generality, we used three-digit subtraction as the problem domain. In this subtraction modeller, context features and action features, adopted from the FBM modeller (Kuzmycz and Webb, 1992), are used to represent inputs and outputs. Context features describe the problems with which a student is faced. Action features describe aspects of a student's actions for a particular problem. There are eleven action features: Result $= M - S$, Result $= M - S - 1$, Result $= 10 + M - S$, Result $= 10 + M - S - 1$, Result $= M$, Result $= S$, Result = zero, Result $= M - S - 2$, Result $= 10 + M - S - 2$, Result $= S - M$ and Result $=$ correct, where M and S stand for minuend and subtrahend digits respectively. Action features are not mutually exclusive. That is, a student's action may correspond to more than one action feature. However, C4.5 requires mutually exclusive classes. Thus, in keeping with the FBM method, FBM-C4.5 uses eleven decision trees to model different aspects of a student's actions (behavior).

The context features of a unit problem are described by 12 attributes. They are listed below with their meanings and possible attribute values, where T, F and N stand for *true*, *false* and *not applicable* respectively.

- M_is_0: {T, F}, the Minuend digit, M, is zero.
- S_is_0: {T, F}, the Subtrahend digit, S, is zero.
- S_is_9: {T, F}, S is nine.
- S_is_BK: {T, F}, S is left blank.
- M_vs_S: {G, L, E}, M is greater (G) or less than (L), or equal (E) to S.
- $M\_L\_is\_0$: {T, F, N}, M in the column to the left is zero.
- $M\_L\_is\_1$: {T, F, N}, M in the column to the left is one.
- $M\_R\_is\_0$: {T, F, N}, M in the column to the right is zero.
- $S\_R\_is\_9$: {T, F, N}, S in the column to the right is nine.
- M_S_R: {G, L, E, N}, similar to M_vs_S, but it describes the column to the right.
- M_S_2R: {G, L, E, N}, similar to M_vs_S, but it describes two columns to the right.
- Column: {L, I, R}, the current column is left-most (L), inner (I), or right-most (R).

Figure 1 illustrates how 11 training examples, one for each decision tree, are formed from the inner column of a 3-digit problem. The context features, described by 12 attributes, are extracted based on the column's environment and applied to

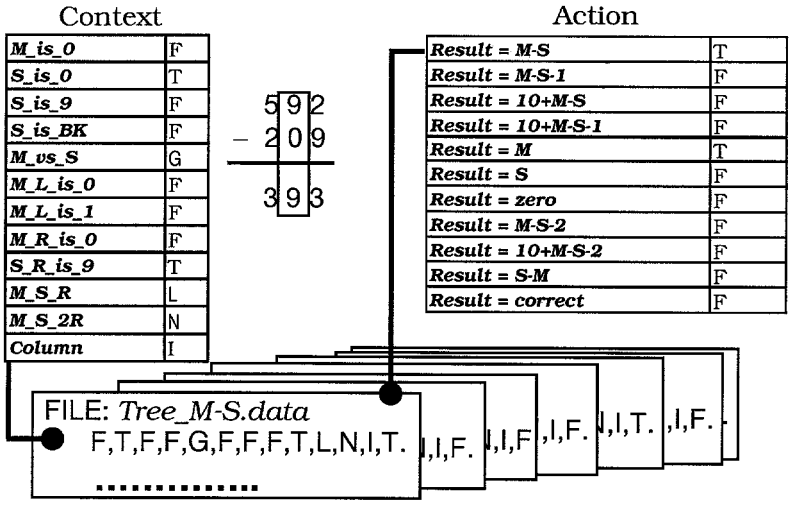

Figure 1. Formation of a column's training examples for decision trees.

each example. At this inner column, M (minuend digit) is nine, S (subtrahend digit) is zero, and the student's answer is nine. Two action features, Result $= M - S$, and Result $= M$, correspond to the student's action. These two action feature attributes are, therefore, set to T. The other action feature attributes are set to F. One 3-digit subtraction problem will generate three training examples for each decision tree. After all examples of a student's subtraction performance are processed, C4.5 is used to infer a decision tree from each training set for the corresponding action feature attribute.

When FBM-C4.5 predicts a student's answer for a problem, the problem context is extracted and used to consult the eleven decision trees. The decision trees being consulted are then confined to those that make a specific prediction for a digit. If these predictions lead to the same digit, the system adopts the digit as the final prediction. Otherwise, the system makes no prediction about the student's answer. Figure 4 shows a sample theory inferred by FBM-C4.5. Decision trees with only one leaf labeled F predict the student will not exhibit the corresponding actions. *Tree_M* predicts that if the subtrahend digit is zero, the student will assign the minuend as the answer.

2. Resolving Conflicting Predictions

All the FBM systems make no prediction, by default, when there exists ambiguity, or conflicts, among the individual predictions from different classification rules or decision trees. This leaves room for improving the systems' performance. A first consideration may be increasing the number of predictions by resolving conflicting predictions. FBM-C4.5 can be augmented by different mechanisms for this purpose. The following methods aim to make more predictions, through con-

flict resolution, without affecting the prediction accuracy. These rely on selecting the more reliable prediction, through the use of voting, and quality measures for decision trees and decision leaves, from a set of competing predictions.

2.1. USING VOTING TO MAKE A FINAL PREDICTION

Among alternatives for resolving conflicting predictions, a simple method is to select a prediction with a majority of votes from the competing predictions. The technique of majority voting is widely used in machine learning. For example, it provides a means for determining a class label at a leaf node of a decision tree where the local training subset contains objects with different class labels while there is no valid attribute for further partitioning (Quinlan, 1986). Similar methods have also been applied with ensembles of classifiers (Heath et al., 1993; Chan and Stolfo, 1995). For FBM-C4.5, the immediate predictions from decision trees predict different aspects of students' actions. However, these predictions can determine specific digits in the range from 0 to 9. Hence, the mechanism of majority voting can be embedded within FBM-C4.5 to resolve conflicting predictions. The predictions of all trees that predict a specific action in the current context are considered. The specific prediction that occurs most frequently is adopted. If two or more specific predictions tie for the first place, no prediction is made. The system will also fail to make a prediction if all trees fail to make a specific prediction.

2.2. USING A MEASURE OF DECISION TREE QUALITY

An alternative approach to improving the prediction rate is to adopt the predictions from the most reliable decision trees. There are a number of ways to infer the quality of decision trees making conflicting predictions, including prediction accuracy on the data from which they were learned. We employed stratified ten-fold cross-validation (Kohavi, 1995) for estimating the error rate of each decision tree. For each action feature, the training examples are randomly divided into ten equal-sized partitions. Each partition, which preserves the original class distribution, is used in turn as test data for the decision tree trained on the remaining nine partitions. The total numbers of correct and incorrect predictions of these tests are then used to estimate the error rate of the decision tree trained on the whole training set. An FBM-C4.5 system can improve its prediction rate by associating decision trees with estimated error rates, and consulting the trees in a ranked order. Only trees associated with error rates that are less than 0.5 will be consulted for the sake of accuracy. With this consulting order, the first tree that gives a positive prediction is used to make the system's prediction. While this method contrasts to the initial system, which consults the trees in parallel, the system may still fail to make a prediction in cases that none of the trees gives a positive prediction.

2.3. USING A MEASURE OF LEAF QUALITY

The error rate of a tree reflects the overall quality of the tree. We know that different leaves have different predictive power because the evidence on which they make predictions is different. A leaf with less support on a high quality tree may make a poorer prediction than a leaf with more support on a tree with a relatively lower quality. A closer look at how C4.5 builds a decision tree may help to explain this. C4.5 builds decision trees using a divide and conquer strategy. It recursively selects the best attribute on which to test and branch. Examples of the training set are partitioned based on the selected attribute. This process continues until the training set at a node cannot be further divided, for example, because no significant split exists. If a terminal node (leaf) contains examples of different classes, it takes the majority class as the leaf label. When the tree is used to predict the class of an unseen (test) example, the example is passed down the branches of the tree corresponding to tests that it satisfies. The label of the leaf at the end of the decision path that is traversed is used as the prediction. The reliability of this prediction can be estimated by examining the distribution of the classes of training examples at the leaf node. With reference to a leaf labeled with T for predicting that a student will exhibit a particular action, let t denote the total number of examples, and e denote the number of examples with alternative labels at the leaf node, then the value, e/t, is the error rate on the training data at the leaf. This could serve as an estimated error rate. However, a Laplacian error estimate, $(e + 1)/(t + 2)$, is used in preference to the actual error rate on that local training set. It favors lower error rates derived from many predictions over similar rates derived from fewer predictions. This is desirable as the former are likely to be more reliable indicators of low predictive error than the latter. The system will make a prediction when one or more individual trees make predictions. If more than one tree makes a prediction, the system can adopt the prediction of the decision node (leaf) associated with the lowest error estimate.

2.4. EXPERIMENTS

The data set used to evaluate FBM-C4.5 (Webb and Kuzmycz, 1996) was used to evaluate the proposed techniques. This data was collected as follows. 73 nine- to ten-year-old primary school students were divided into two treatments, Random group (36 subjects) and Error Repeat group (37 subjects). These subjects were administered five rounds of tests. Each test consisted of 40 three-column subtraction problems. Successive tests were all administrated at weekly intervals. In both Tests 1 and 5, a set of subtraction problems was randomly generated and presented to all subjects. For Tests 2 to 4, the experimental data was collected as follows:

- for the Random group, a new set of randomly generated problems was presented to the subjects;

- for the Error Repeat group, all problems from the last problem sheet for which the subject made an error were copied to the new problem set. Mixing new randomly generated problems to make a total of 40, the new set of problems was presented to the subjects.

For evaluation, a modeling system used all data from prior tests to build a student model and used the current test data to evaluate the current student model. That means the system started at Test 2 where Test 2 data was used as evaluation data against a student model which was built based on the data from Test 1. The subjects contributed 264 test sheets. In these 264 model training-testing processes, there were a total of 30,474 student answers, of which 3,630 were incorrect answers.

The performance of the original version of FBM-C4.5 was used as a baseline against which the new conflict resolution mechanisms were evaluated. It made 28,700 predictions, of which 26,507 (92.4%) were correct. Of the system's 1,999 predictions that a subject would provide an incorrect digit for a column (error predictions), 1,347 (67.4%) were accurate, predicting the exact digit provided. Because all versions are tested on the same data, we used matched-pairs t-tests (Salvia, 1990) to evaluate the statistical significance of observed differences in prediction and accuracy rates. The t-test is commonly used to compare machine learning algorithms. To determine if there is a significant difference between prediction performance, we use a t-test with the alpha level at 0.05. A computed p value can be used as a measure of how sure we can be that a difference observed in the samples is also true for the whole population. If the computed p value is below the alpha level, 0.05, we can be confident enough to conclude there is a performance difference between the corresponding modeling methods. Where we have not predicted in advance which technique will outperform the other, a two-tailed test is used. Where we have made such a prediction, a one-tailed test is used.

2.5. RESULTS

The labels +Vote, +Tree_qty and +Leaf_qty are used to represent new versions that were implemented by introducing voting and ranking by quality measures on trees and leaves respectively. Table I summarizes the systems' overall performance against the baseline, where percentages in bold type highlight values that are significantly better than those of the original system while underlined values indicate results that are significantly worse. The p value and t value with df (degree of freedom) in column $+X$ row Y represent the statistical result of comparing version $+X$ against FBM-C4.5 for a performance category Y.

All new versions achieved significant improvements in prediction rate. While they made more correct predictions than FBM-C4.5, lower proportions of the additional predictions that the systems made were correct than of the baseline. That is, their overall prediction accuracy dropped slightly but significantly. The introduc-

Table I. Performance of new versions with conflict resolution against the baseline (two-tailed t-test).

	FBM-C4.5	+Vote	+Tree_qty	+Leaf_qty
Total predictions made	28,700	29,729	29,783	30,093
Prediction rate (%)	94.2	**97.6**	**97.7**	**98.7**
p value		<0.0001	<0.0001	<0.0001
t value (df)		14.28 (263)	9.55 (263)	15.16 (263)
Total predictions that were correct	26,507	27,321	27,308	27,543
	92.4	91.9	91.7	91.5
Prediction accuracy (%)		<0.0001	0.0001	<0.0001
p value		−4.21 (263)	−3.88 (263)	−6.31 (263)
t value (df)				
Error predictions made	1,999	2,169	2,346	2,173
Total prediction rate (%)	55.1	**59.8**	**64.6**	**59.9**
p value		< 0.0001	<0.0001	<0.0001
t value (df)		6.28 (263)	6.71 (263)	4.23 (263)
Error predictions that were correct	1,347	1,407	1,485	1,426
	67.4	64.9	63.3	65.6
Error prediction accuracy (%)		0.0361	0.0056	0.0272
p value		−2.13 (90)	−2.84 (90)	−2.25 (90)
t value (df)*				

*For error prediction accuracy, only pairs in which error predictions have been made can be compared.

tion of conflict resolving mechanisms increased the number of error predictions at the expense of error prediction accuracy. However, it should be noted that the number of correct predictions in both cases rises. Neither system alters any of the predictions made by the original FBM-C4.5. Rather, each makes additional predictions in some contexts for which the original system was unable to make predictions, but the accuracy of these additional predictions is lower than the accuracy of the original predictions.

2.6. DISCUSSION

Kuzmycz (1997) studied four strategies for resolving conflicting predictions for the FBM system.These strategies include adopting the most general (*General*), adopting the most specific (*Specific*), adopting the one with highest probability

Table II. Performance of four versions for conflict resolution in Kuzmycz's study.

		Default	Specific	General	Preference	Confidence
Overall	Rate (%)	82.28	**94.76**	**98.49**	**98.65**	**97.46**
Prediction	Accuracy (%)	92.27	83.65	90.60	89.47	88.68
Error predictions accuracy (%)		70.49	33.54	68.58	67.65	50.77

(*Confidence*), and referring to a ranked list of a priori likelihood (*Preference*). Our tree and leaf quality approaches can be viewed as variants of the confidence strategy, differing primarily in how the level of confidence is measured. Kuzmycz shows that all of the strategies increase the prediction rate significantly, at a cost of reducing prediction accuracy. Table II depicts part of the results from Kuzmycz's (1997) study on resolving conflicting predictions. Only percentages are listed because that study used a different set of subjects. The table shows the performance differences of Kuzmycz's four different methods for conflict resolution against their baseline (making no prediction by default, when conflicts exist). The values in bold font indicate the performance is improved significantly, while the underlined values imply the performance is degraded significantly.

Our results mirror Kuzmycz's findings: techniques that resolve conflicting predictions increase the number of predictions at the expense of decreasing prediction accuracy. While the decline in prediction accuracy from our techniques is less than those reported from Kuzmycz's techniques, our ultimate goal, to improve the system's prediction rates without affecting its prediction accuracy, is still not achieved. Should there exist ways to achieve this goal, the problem must be reviewed from other perspectives. We also need a theory to account for this trade-off effect found with conflict resolution. Ambiguous or conflicting predictions may reflect inconstancy in an agent's actions. This can be regarded as a special kind of information addressing the agent's behavior. This information will be lost once a set of competing hypotheses is resolved. It may explain why the prediction accuracy drops when conflicting predictions are resolved in favor of a single alternative. For this reason, techniques other than conflict resolution should be explored.

3. Using the Most Useful Action Features to Build a Single Tree

Motivated by this unsolved problem, how to reduce ambiguous predictions without significantly degrading prediction accuracy, we seek a solution with other approaches. The tree and leaf quality approaches, as well as those in Kuzmycz's (1997) study, attempt to resolve multiple predictions. One may circumvent the problem by producing only one prediction for each unseen case. We can achieve

this by developing a single tree that predicts the most *useful* action feature for predicting an agent's actions in a given context. Such a tree requires a training set labeled with the most useful action feature for each example.

3.1. TWO-PHASE IDENTIFICATION ALGORITHM

We propose a two-phase identification algorithm (see Figure 2) that can be employed at the training stage. For each training example that is accompanied with more than one action feature, each action feature is validated by a lazy Bayesian tree* (Zheng and Webb, 1997) trained from all other training examples. The lazy Bayesian tree is used for the sake of computational efficiency. This filtering process reduces the number of examples with multiple action features. At the second stage, those training examples with multiple action features form a temporary test set. A temporary decision tree, trained on examples that each has a unique action feature, predicts the most useful action feature for each example in the test set. The ultimate training set, in which each example is labeled by a most useful action feature, or as unknown if a most useful action feature cannot be identified, is used to infer a single tree for the system.

3.2. EVALUATION

The Single-tree approach was evaluated with the same data set used in the previous study. The overall performance of Single-tree against the baseline is shown in Table III. The Single-tree's improvement in prediction rate is significant (in bold font) while its decrease in prediction accuracy is also significant (underlined). In comparison with those techniques for resolving conflicting predictions, presented in Table I, Single-tree produces the greatest increase in the number of predictions but at the expense of having the lowest accuracy. For error prediction rate and error prediction accuracy, the differences between these two systems are not significant.

Regarding the inferred theories generated by these two systems, the difference between their knowledge representations is obvious. Figures 3 and 4 show the outputs of the Single-tree and multi-tree (FBM-C4.5) versions when a student's first test performance was captured. Both models exhibited identical performance in predicting the student's answers for the next test. The multi-tree representation analyses each action in detail. For the Single-tree version, a leaf labeled as correct covers those actions leading to correct answers, while a leaf with an other label indicates how an erroneous action is predicted. We suggest that this single-tree model is likely to be easier for teachers and students to understand.

* For each test example, the Lazy Bayesian Tree learning algorithm generates one relevant decision path. The leaf of the path uses a local naive-Bayesian classifier, instead of a majority class, to classify the test example.

```
Given: raw training set M with N examples, from a single student, in the form of
        (Att, undetermined_action) where Att is a set of problem context features;
        11 training sets, A_i , each has N examples in the form of (Att, a_i ), a_i ∈ {F,T} where
        F/T stand for the status, absence/presence, of a corresponding action feature.
Output: a training set of examples in the form of (Att, Action), Action ∈ { M-S, M-S-1,..,
        correct, unknown}.

FOR n := 1 to N DO
        obtain a status list (a_1 ,...,a_i ,.., a_11 ) from the n-th examples from training sets A_1 ,..,A_i
        ,..,A_11
        generate an index list, L, of competing actions, where L = (.,i,..) for any a_i = T
        IF L has more then one element THEN L = Reduce_ competing_actions (L, n)
        IF L has one element i THEN undetermined_action := Action_i
        ELSE IF L is empty THEN undetermined_action : = unknown
        ELSE
           append example_n to undetermined example list U; and
           append L to a list of competing action lists LL

FOR each undetermined example, example_u , in U DO
        retrieve the corresponding L from LL
        undetermined_action = Classify_within_competing_actions(example_u, L)

Process name: Reduce_competing_actions.          /* Phase 1: Internal identification */
Given: L, an index list of competing actions; n, an index of the current example.
Output: L, a revised index list of competing actions.

        FOR each index i in L DO
           build a lazy Bayesian tree LBT_i based on A_i excluding the example_n
           IF the predicted class of the test item, example_n , is F
           THEN remove i from L
        RETURN L

Process name: Classify_within_competing_actions.   /* Phase 2: Global identification */
Given: example_u , a test example ; L, a list of index of competing actions.
Output: an action, or unknown.

        prepare a temporary training set by copying all examples, for which class labels
        match the actions described by the index in L, from the raw training set
        IF the temporary training set is not empty
        THEN
           build a decision tree D to test example_u and RETURN the action predicted by D
        ELSE
          RETURN unknown
```

Figure 2. Two-phase identification process.

Table III. Performance of Single-tree against the baseline (two-tailed t-test).

	FBM-C4.5	Single-tree
Number of predictions made	28,700	30,130
Prediction rate (%)	94.2	**98.9**
p value		<0.0001
t value (df)		11.93 (263)
Total predictions that were correct	26,507	27,495
Prediction accuracy (%)	92.4	91.3
p value		<0.0001
t value (df)		−5.53 (263)
Error predictions made	1,999	2,095
Error prediction rate (%)	55.1	57.7
p value		0.1328
t value (df)		1.51 (263)
Error predictions that were correct	1,347	1,373
Error prediction accuracy (%)	67.4	65.5
p value		0.1886
t value (df)		−1.33 (85)

Tree_actions

```
M_S_R = G: correct
M_S_R = N: correct
M_S_R = L:
|---M_vs_S = G: M-S
|---M_vs_S = L: 10+M-S
|---M_vs_S = E: S-M
M_S_R = E:
|---M_S_2R = G: correct
|---M_S_2R = L: M-S
|---M_S_2R = E: correct
|---M_S_2R = N: correct
```

Figure 3. Knowledge representation inferred by a single-tree modeller.

Tree_M-S-1

```
F
```

Tree_M-S-2

```
F
```

Tree_10+M-S-1

```
F
```

Tree_10+M-S-2

```
F
```

Tree_M

```
S_is_0 = F: F
S_is_0 = T: T
```

Tree_M-S

```
M_vs_S = G: T
M_vs_S = L: F
M_vs_S = E: T
```

Tree_10+M-S

```
M_vs_S = G: F
M_vs_S = L: T
M_vs_S = E: F
```

Tree_zero

```
M_vs_S = G: F
M_vs_S = L: F
M_vs_S = E: T
```

Tree_correct

```
M_S_R = G: T
M_S_R = L: F
M_S_R = N: T
M_S_R = E:
|---M_S_2R = G: T
|---M_S_2R = L: F
|---M_S_2R = E: F
|---M_S_2R = N: T
```

Tree_S

```
M_is_0 = T: F
M_is_0 = F:
|---M_L_is_1 = F: F
|---M_L_is_1 = N: F
|---M_L_is_1 = T:
    |---S_R_is_9 = F: F
    |---S_R_is_9 = T: T
    |---S_R_is_9 = N: F
```

Tree_S-M

```
M_vs_S = G: F
M_vs_S = E: T
M_vs_S = L:
|---M_L_is_1 = T: F
|---M_L_is_1 = N: F
|---M_L_is_1 = F:
    |---S_is_9 = T: T
    |---S_is_9 = F:
    |---M_R_is_0 = T: F
    |---M_R_is_0 = N: F
    |---M_R_is_0 = F:
        |---M_S_R = G: T
        |---M_S_R = L: F
        |---M_S_R = E: F
        |---M_S_R = N: F
```

Figure 4. Knowledge representation inferred by a multi-tree modeller.

4. Using Models With Temporal Characteristics

The experiments mentioned in previous sections reveal that prediction accuracy will be affected if resolution of conflicting predictions or action features is forced. We then are more inclined to modify our goal as follows: to seek techniques that enable the system to make more predictions without degrading the prediction accuracy. To achieve this, other factors or issues should be looked to.

One important issue is managing temporal knowledge in agent modeling, particularly in an educational context. Giangrandi and Tasso (1996) argue that a static student model, a model which does not take the time factor into account, does not

truly reflect a student's knowledge. While Giangrandi and Tasso proposed a temporal management mechanism, such that contradicting hypotheses about a student can co-exist within a student model, Webb (1989) embedded in the FBM system a data aging mechanism, which discounts old data by a set factor. These approaches are based on an assumption that the agent's knowledge, beliefs, and skills may alter over time. This assumption seems justified for domains where data reflecting the agents' activities covers an extended time period. FBM-C4.5 does not include any temporal management mechanism. In the domain we studied, students were tested five times, each test separated by one week. At the last round of the model building process, some data was four weeks old. To build more realistic models, this factor should be considered.

4.1. HYPOTHESIS

Considering that successive tests to each subject were administrated at regular time intervals, we believe it would be meaningful to investigate whether there will be significant performance differences for student models that are built on batches of data from different times. In our subtraction domain, different times equate to different tests. Our first experimental hypothesis was that a student model which is built from the data of the most recent test will better explain (in terms of prediction accuracy or prediction rate) the student's future actions than those built from the data of any prior test. Let i be the test number, M_j be the student model built on the data of the jth test, and $p(M_j, T_i)$ be the prediction performance of M_j on the test data T_i. This hypothesis can be expressed as: H1: $p(M_{i-1}, T_i) > p(M_j, T_i)$, for $j < i - 1$.

Observe that the hypothesis is directional; we used one-tailed paired t-tests at 0.05 level to evaluate it. The same data set, which has been described in the Section 2, provides four models (M_1 to M_4) for each student. Each model was evaluated against the data from any subsequent test. That means M_1 was evaluated against Tests 2 to 5, and M_4 was evaluated against Test 5 only. The total number of predictions made and the total number of correct predictions made by all M_i against the test data T_j constituted the overall prediction accuracy of M_i on T_j. Figure 5 depicts the overall behavior of each model. It shows, from Tests 3 to 5, that the prediction accuracy of *fresh* models (built from most recent data) is always higher than that of aged models. This result supports the hypothesis that *fresh* models are better than aged models for predicting future actions in terms of both prediction rate and prediction accuracy (see Table IV).

It is surprising that whereas the most recent model always performs best on a given test, the remaining models are not ordered on performance from more to less recent. We believe that this is due to the manner in which tests were generated from round to round. On each of the first and last rounds all subjects received the same randomly generated test. There is reason to believe that these happened to be different in nature to the average test generated on other rounds. For example,

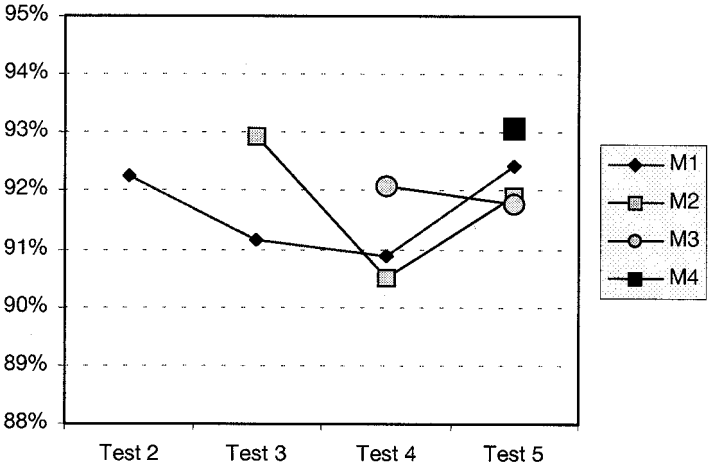

Figure 5. Overall prediction accuracy of individual models on further tests.

Table IV. Statistical significance of hypothesis H1 (one-tail t-test).

Test	$p(M_{i-1}, T_i) > p(M_j, T_i)$ for prediction rate		$p(M_{i-1}, T_i) > p(M_j, T_i)$ for prediction accuracy	
	p value	t value (df)	p value	t value (df)
$j = i - 2$	0.0009	3.15 (91)	0.0018	2.95 (191)
$j = i - 3$	0.0001	3.90 (119)	0.1147	1.21 (119)
$j = i - 4$	<0.0001	4.33 (51)	0.4874	−0.03 (51)

whereas M_1 accuracy drops round by round from test 2 to test 4, on test 5, four weeks after the data on which they are based were collected, the M_1 models obtain the highest accuracy they ever record.

4.2. A DUAL-MODEL SYSTEM

Given the support for the temporal recency hypothesis, we proposed a simple method to cater for the temporal factor in the domain that has been studied in the Section 2. We develop a Dual-model system. A *fresh* model, which is built using data from the most recent test, is expected to predict a student's future actions, within its scope, more accurately than a model produced from all tests. However, a *fresh* model cannot be expected to cover all aspects of skill in depth, and so it is expected that there will be many tasks for which the *fresh* model will be unable to make a prediction. In these situations, an *extended* model, which is inferred from data of all prior tests, can be consulted. Each model consists of eleven decision trees, with which ambiguous situations can be detected. To predict a student's

action, the system will first consult the *fresh* model. If the fresh model makes no prediction, due to insufficient training data or inconsistency among the training data, the system then consults the extended model.

4.3. EXPERIMENTS AND RESULTS

We present two evaluations of the performance of the Dual-model system against FBM-C4.5. When only one batch of training data is available for learning a model, as is the case when predicting a student's performance on Test 2, the Dual-model system reverts to the original FBM-C4.5, as there is only Test 1, the most recent test, from which to form a model. For this reason, evaluation on Test 2 was not considered for the first comparison of the two systems. Rather, the first evaluation started model testing from Test 3 where multiple models were available. Each system conducted 192 model tests. There were a total of 22,140 student answers, of which 2,687 were incorrect answers. The Dual-model achieved an overall prediction rate of 98.3%, which is significantly higher (one-tailed t-test: $p < 0.0001$) than that of FBM-C4.5 (96.0%). The overall prediction accuracy of the Dual-model and FBM-C4.5 are 92.1% and 92.4% respectively, and there is no significant difference (two-tailed t-test: $p = 0.3886$) in this respect.

To facilitate comparison between approaches, a final performance evaluation with 264 model tests was done by employing the full data set which has been used for evaluating other versions (see Sections 2 and 3). The Dual-model made more predictions than the baseline. The improvement in prediction rate is significant (in bold font) and the difference in prediction accuracy is not significant. With respect to error prediction, Dual-model made more error predictions and improved error prediction accuracy slightly but these differences are not statistically significant. Table V summarizes the performance of Dual-model system against the baseline.

4.4. DISCUSSION

The results of the Dual-model approach show that the simple combination of a *fresh* model and an *extended* model is effective for achieving our objective. We have explored other various forms of model combination, for example, a multiple model approach that lets a system make predictions by consulting a *fresh* model first, then consulting the second freshest model, and so on. Other Dual-model variants have also been evaluated. They include a system that consults an *extended* model if there is no prediction from any tree in the *fresh* model, a system that consults a second model built on a single test if the *fresh* model gives no prediction, and a system that consults the fresh model first, then it consults an extended model which is built on all test data excluding the most current test data. Neither the multiple-model system nor these alternative formulations of Dual-model achieved better prediction performance than the current form of Dual-model. Similar strategies of

Table V. Prediction performance of Dual-model system against the baseline (two-tailed t-test).

Total predictions made	28,700	29,218
Prediction rate (%)	94.2	**95.9**
p value		<0.0001
t value (df)		6.27 (263)
Total predictions that were correct	26,507	26,911
Prediction accuracy (%)	92.4	92.1
p value		0.7769
t value (df)		-0.28 (263)
Error predictions made	1,999	2,101
Error prediction rate (%)	55.1	57.9
p value		0.1148
t value (df)		1,58 (263)
Error predictions that were correct	1,347	1,442
Error prediction accuracy (%)	67.4	68.6
p value		0.6011
t value (df)		0.53 (85)

combining two models for improving prediction performance have been previous studied in machine learning and it was concluded that the use of more than two models contributed no significant advantage over using two models (Ting and Low, 1997; Chan and Stolfo, 1995). While those studies built models on partitioned data with no temporal consideration, the data set of this study is that in which each batch of data is separated by a fixed length of time. The use of a Dual-model that takes the temporal factor into account is therefore more justifiable.

5. Conclusions

We have addressed the problem of how to improve the prediction rate of the FBM-C4.5 agent modeling system without degrading the prediction accuracy. Five techniques for this objective have been presented and evaluated.

Techniques of employing voting and quality measures on decision trees and leaf nodes in resolving conflicting predictions at the testing stage have been shown to be effective for improving the prediction rate. The first method treats all predictions as equal when resolving conflicting predictions. The quality methods, however, cover two aspects of resolving conflicts: selecting a decision using a global level evaluation of tree quality; and selection based on local (leaf) quality. Experimental

results show that resolving multiple predictions into a single alternative may have a negative effect on prediction accuracy.

A technique for merging multiple trees into a single tree allows the system to make more predictions and was expected to lower the risk of affecting prediction accuracy. Empirical evaluation shows that the Single-tree version achieved significant improvement in prediction rate with a slight drop in overall prediction accuracy. However, there was no significant degradation in predicting incorrect answers, when compared with the baseline. This approach generates only one decision tree for each student. This can be an additional advantage, because the theories generated are easier for people to understand.

The Dual-model approach achieved significant improvement in prediction rate without significantly affecting prediction accuracy. Our experimental results reveal, in situations where a series of tests are conducted over an extended period of time, the Dual-model approach, which takes temporal factors into account, provides better prediction performance. The disadvantage of this approach is that it uses twice the number of decision trees to describe a student's competencies. While this approach has been developed and evaluated in the context of FBM-C4.5, it should be equally applicable to any agent modeling system that constructs models from multiple observations over time.

It may be possible to reveal more knowledge about a student by combining the above techniques. For example, employment of the Single-tree technique in a Dual-model system generates only two trees. This may allow a tutor to observe a student's concept changes.

These results relate only to experiments in the domain of subtraction. The FBM method has far wider application, however. Existing applications include tutoring piano scale playing (Amato and Tsang, 1990), English word class identification (Webb, 1989), and unification of terms in the Prolog programming language (Webb, 1991). It would be valuable to extend the comparison of the techniques that we present to other such domains. In the mean time, the current results present strong support for the efficacy of the Dual-model technique in increasing numbers of predictions without significant impact on prediction accuracy, and of the Single-tree technique in simplifying the models that are produced.

References

Ali, K., Brunk, C., and Pazzani, M.: 1994, On learning multiple descriptions of a concept. In *Proceedings of the Sixth International Conference on Tools with Artificial Intelligence*, New Orleans, LA: IEEE Press, 476–483.

Amato, N. H. and Tsang, C. P.: 1990, Student modeling in a keyboard scale tutoring system. In C. J. Barter & M. J. Brooks (Eds.) *Proceedings of the Second Australian Joint Conference on Artificial Intelligence*, Berlin: Springer-Verlag, 225–239.

Anderson, J. R., Boyle, C. F., and Reiser, B. J.: 1985, Intelligent tutoring systems. *Science*, **228** 456–462.

Anderson, J. R., Boyle, C. F., Corbett, A. T., and Lewis, M. W.: 1990, Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, **42** 7–49.

Baffes, P., and Mooney, R.: 1996, Refinement-based student modeling and automated bug library construction. *Journal of Artificial Intelligence in Education*, **7**(1) 75–117.

Balabanovic, M.: 1998, Exploring versus exploiting when learning user models for text recommendation. In this volume.

Breiman, L.: 1996, Bagging predictors. *Machine Learning*, **24** 123–140.

Brown, J. S., and VanLehn, K.: 1980, Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science*, **4** 379–426.

Brown, J. S., and Burton, R. R.: 1978, Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, **2** 155–192.

Burton, R. R, and Brown, J. S.: 1976, A tutoring and student modeling paradigm for gaming environments. *Computer Science and Education. ACM SIGCSE Bulletin*, **8**(1) 236–246.

Carbonell, J. R.: 1970, AI in CAI: An artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, **11**(4) 190–202.

Chan, P. K., and Stolfo, S. J.: 1995, A comparative evaluation of voting and meta-learning on Partitioned data. In Prieditis, A., and Russell, S., eds., *Proceedings of the 12th International Conference on Machine Learning*, 90–98.

Chiu, B. C., Webb, G. I., and Kuzmycz, M.: 1997, A comparison of first-order and zeroth-order induction for input-output agent modeling. In Jameson A., Paris C., and Tasso C., eds., *Proceedings of the Sixth International Conference on User Modeling*, UM97, 347–358.

Corbett, A. T., and Anderson, J. R.: 1992, Student modeling and mastery learning in a computer-based programming tutor. In Frasson, C., Gauthier, G., and McCalla, G. I., eds., *Intelligent Tutoring Systems*. Berlin: Springer-Verlag, 413–420.

Desmoulins, C., and Van Labeke, N.: 1996, Towards student modeling in geometry with inductive logic programming. In Brna, P., Paiva, A., and Self, J., eds., *Proceedings of the European Conference on Artificial Intelligence in Education*, 94–100.

Dietterich, T., and Bakiri, G.: 1994, Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, **2** 263–286.

Giangrandi, P., and Tasso, C.:1995, Truth maintenance techniques for modeling students' behavior. *Journal of Artificial Intelligence in Education*, **6**(2/3) 153–202.

Giangrandi, P., and Tasso, C.: 1996, Modeling the temporal evolution of student's knowledge. In Brna, P., Paiva, A., and Self, J., eds., *Proceedings of the European Conference on Artificial Intelligence in Education*, 184–190.

Gilmore, D., and Self, J.: 1988, The application of machine learning to intelligent tutoring systems. In Self, J., ed., Artificial Intelligence and Human Learning: *Intelligent Computer-aided Instruction*. London: Chapman and Hall, 179–196.

Goldstein, I. P.: 1979, The genetic graph: A representation for the evolution of procedural knowledge. *International Journal of Man-machine Studies*, **11**(1) 51–77.

Hoppe, H. U.: 1994, Deductive error diagnosis and inductive error generalization for intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, **5**(1) 27–49.

Heath D., Kasif, S. and Salzberg, S.: 1996, Committees of decision trees. In Gorayska, B., and Mey, J., eds., *Cognitive Technology: In Search of a Human Interface*. Amsterdam: Elsevier Science B.V., 305–317.

Ikeda, M., Kono, Y., and Mizoguchi, R.: 1993, Nonmonotonic model inference: A formalization of student modeling. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence: IJCAI'93*, 467–473.

Kohavi, R.: 1995, A study of cross-validation and bootstrap for accuracy estimation and model selection. In Mellish, C. S., ed., *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1137–1145.

Kuzmycz, M.: 1994, A dynamic vocabulary for student modeling. *Proceedings of the Fourth International Conference on User Modeling*, Hyannis, MA, 185–190.

Kuzmycz, M.: 1997, Resolving conflicting knowledge in student models. *Proceedings of the Eighth World Conference on Artificial Intelligence in Education*. Amsterdam: IOS Press, 522–529.

Kuzmycz, M., and Webb, G. I.: 1992, Evaluation of feature based modelling in subtraction. In Frasson, C., Gauthier, G., and McCalla, G. I., eds., *Intelligent Tutoring Systems*. Berlin: Springer-Verlag, 269–276.

Kwok, S., and Carter, C.: 1990, Multiple decision trees. *Uncertainty in Artificial Intelligence*, **4** 327–335.

Langley, P., and Ohlsson, S.: 1984, Automated cognitive modeling. *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas, 193–197.

Langley, P., Wogulis, J., and Ohlsson, S.: 1990, Rules and principles in cognitive diagnosis. *Diagnostic Monitoring of Skill and Knowledge Acquisition*. Hillsdale, NJ: Erlbaum, 217–250.

Martin, J., and VanLehn, K.: 1995, Student assessment using Bayesian nets. *International Journal of Human-Computer Studies*, **42**(6) 575–591.

Nock, R., and Cascuel, O.: 1995, On learning decision committees. *Proceedings of the Twelfth International Conference on Machine Learning*, San Mateo, CA: Morgan Kaufmann. 413–420.

Ohlsson, S., and Langley, P.: 1985, Identifying solution paths in cognitive diagnosis. *Technical Report CMU-RI-TR-85-2*, Carnegie-Mellon University, Pittsburgh, PA.

Oliver, J. J., and Hand, D. J.: 1995, On pruning and averaging decision trees. *Proceedings of the Twelfth International Conference on Machine Learning*, San Mateo, CA: Morgan Kaufmann, 430–437.

Quinlan, J. R.: 1986, The effect of noise on concept learning. In Michalski, R. S., Carbonell J. G. and Mitchell T.M. eds., *Machine Learning: An Artificial Intelligence Approach*, Vol. 2. San Mateo, CA: Morgan Kaufmann, 149–166.

Quinlan, J. R.: 1993, C4.5: *Programs for Machine Learning* San Mateo, CA: Morgan Kaufmann.

Salvia, A. A.: 1990, *Introduction to Statistics*. Saunders College Pub., Philadelphia.

Schapire, R. E.: 1990, The strength of weak learnability. *Machine Learning*, **5** 197–227.

Sleeman, D.: 1982, Assessing aspects of competence in basic algebra. In Sleeman, D. H., and Brown, J. S., eds., *Intelligent Tutoring Systems*. London: Academic Press, 185–199.

Sleeman, D., Ward, R. D., Kelly, E., Martinak, R., and Moore, J.: 1991, An overview of recent studies with Pixie. In Goodyear, P., ed., *Teaching Knowledge and Intelligent Tutoring*. Norwood, NJ: Ablex, 173–185.

Ting, K. M., and Low, B. T.: 1997, Model combination in the multiple-data-batches scenario. *Proceedings of the Ninth European Conference on Machine Learning, Lecture Notes in Artificial Intelligence*, Vol. 1224, Berlin: Springer-Verlag, 250–265.

Webb, G. I.: 1989, A machine learning approach to student modeling. *Proceedings of the Third Australian Joint Conference on Artificial Intelligence*, Melbourne, 195–205.

Webb, G. I.: 1991, Inside the Unification Tutor: The architecture of an intelligent educational system. In *Proceedings of the Fourth Australian Society for Computers in Learning in Tertiary Education Conference*, Launceston, 677–684.

Webb, G. I., Chiu, B. C., and Kuzmycz, M.: 1997, Comparative evaluation of alternative induction engines for Feature Based Modelling. *International Journal of Artificial Intelligence in Education*, 8, to be printed.

Webb, G. I., and Kuzmycz, M.: 1996, Feature based modeling: A methodology for producing coherent, dynamically changing models of agent's competencies. *User Modeling and User-Adapted Interaction*, **5**(2) 117–150.

Wolpert, D. H.: 1992, Stacked generalisation. *Neural Networks*, **5** 241–259.

Young, R., and O'Shea, T.: 1981, Errors in children's subtraction. *Cognitive Science*, **5**(1) 153–177.

Zheng, Z., and Webb, G. I.: 1997, Lazy Bayesian Tree. *Technical Report TC97/07*, School of computing and mathematics, Deakin University.

Authors' Vitae

Bark C. Chiu
Ph.D. candidate in Computing at Deakin University. He received his B.Sc. from the University of Hong Kong in 1982, and his M.InfoCom. degree in Information Technology from the University of Wollongong. His primary interests lie in the areas of user modeling and machine learning, although previous research has included work in multimedia systems for tutoring.

Dr Geoff Webb
Reader in Computing at Deakin University where he has established and leads the Deakin University Knowledge Acquisition and Processing research group. He received his B.A. and Ph.D. degrees in Computer Science from La Trobe University. The author of more than sixty technical papers, he is active in several areas of artificial intelligence research, including machine learning, knowledge acquisition, agent modelling, and artificial intelligence in education.