Data mining techniques for design of ITS student models

Ritu Chaturvedi and C. I. Ezeife
School of Computer Science, University of Windsor
Windsor, Canada N9B 3P4
rituch@uwindsor.ca and cezeife@uwindsor.ca

ABSTRACT

An Intelligent Tutoring System (ITS) is a computer system that provides a direct customized instruction or feedback to students while performing a task in a tutoring system without the intervention of a human. One of the modules of an ITS system is student module which helps to understand the student's learning abilities. Several data mining techniques like association rule mining, clustering and mining using Bayesian networks have been proposed to design effective student models in ITS systems. This paper provides a comparative study of the various data mining techniques and tools that are used in student modeling. We also propose an example-driven approach that can integrate mined concept examples at different difficulty levels with the Bayesian networks in order to influence student learning.

1. INTRODUCTION

An ITS system has four main components: interface which provides the means for the students to interact with the ITS through a GUI, expert model which describes the knowledge that represents expertise in the subject matter domain the ITS is teaching, a tutor model that takes or suggests corrective action when necessary and a student model that describes student behavior or knowledge, including his/her misconceptions. Though a lot of work has been done on student model data that describes their cognitive skills like knowledge on a particular concept [2], not too much attention has been paid to outside factors like examples that can be provided as help to students while they use the tutoring system and how they can improve student engagement and their skill level. WebEx [1] is a web-based tool for exploring programming examples that enable teachers to use example-based approach in order to maximize learning opportunity for every student, weak or strong by allowing them to explore program examples in his or her own pace and order. We propose to differ from this approach by rating the examples.

2. DATA MINING TECHNIQUES AND TOOLS IN STUDENT MODELING

A correlation between objectives of creating a student model and the mining methods used for the respective objectives is shown in Table 1. If the student model design is directed towards teaching strategies / course planning, where the stakeholder is an educator, the most effective mining methods are clustering and association rule mining, whereas if the student model design is directed towards student learning and improving, then the mining method most effective is classification and prediction. With student as a stakeholder, the most commonly used classification algorithms used are decision tree algorithms like C4.5 and J48 and Bayesian classifiers such as Bayesian networks. To experiment on the comparative analysis of these mining methods, we used two data mining tools: WEKA (http://www.cs.waikato.ac.nz/ml/weka/) and Bayes Server (http://www.bayesserver.com).

	Objective	Mining method	Algorithms proposed	Most useful for
1	Measure student's motivation	Clustering	K-means Estimation Mazimization	Teachers
2	Categorize students as weak or regular based on the hints used, attempts made etc.			
3	Decide if students use resources (hints etc.) and if it has an impact on the marks	Association Rule Mining	Apriori FP-growth	Teachers
4	Assist students in the sequence they should follow while using the tutor			
5	Find common mistakes			
6	Discover potential student groups with similar characteristics and reactions to a particular pedagogical strategy	Classification Prediction	Decision tree - C4.5, J48 Bayesian Classifiers	Student
7	Detect students' misuse or game-playing			
8	Find common misconceptions that students possess			
9	Identify learners with low motivation			

Table 1: mining methods used in student model

2.1 Overview of Bayesian Networks

A Bayesian Network is a directed acyclic graph (DAG) of random variables (such as concepts, e.g., Add2Fracs) that uses Bayes theorem to depict probabilistic relationships between these variables. BNs used in student models typically have their DAGs designed by experts and the probabilistic relationships between variables are estimated by using some training data such as that in Table 2. Here, it is assumed that Add2Fracs is probabilistically dependent on Find_LCM and Attempt_made, hence Add2Fracs has these two as its parents.

Find_LCM	Attempts_made	Add2Fracs
TRUE	1	Pass
FALSE	2	Fail
TRUE	1	Fail
TRUE	2	Pass
TRUE	3	Pass
TRUE	1	Fail
TRUE	2	Pass
FALSE	2	Fail
FALSE	2	Pass
TRUE	2	Pass

Table 2: An Example BN Training Dataset

For a given sample X=(Find_LCM=False, Attempts_made=2), we have to compute $P(X|C_i)$ for each value of class attribute C_i (In table 2, C is Add2Fracs that can have 2 values Pass and Fail) and find the maximum of them. Since C_i has 2 values and X has 2 attributes, the number of such posterior probabilities is 4, as given in the Table 3. Using these probability values, we can compute P(X|Add2Fracs = Pass) = 1/6* 4/6=0.11 and P(X|Add2Fracs=Fail) = 2/4*2/4 = 0.75. This indicates that the probability of failing Add2Fracs is higher for the given sample X. A variable in BN is conditionally independent of all its nondescendants given its parents. The joint probability of the network computed by multiplying the conditional probabilities of each variable given its parents is $P(X_1 \ldots X_N) = \pi_i \ P(X_i \mid Pa(X_i))$ where Pa stands for parents of, i.e., $P(X_i) = P(X_i \mid Pa(X_i))$.

P(Find_LCM=False | Add2Fracs = Pass)	1/6
P(Find_LCM=False | Add2Fracs = Fail)	2/4
P(Attempts_made=2 | Add2Fracs = Pass	4/6
P(Attempts_made=2 | Add2Fracs = Fail)	2/4

Table 3: Posterior probabilities for Sample X

When building a student model for the domain of adding fractions, each variable (concepts, tests and other student attributes) is assigned a conditional probability (CPT). Root(s) of the BN will store only prior probabilities in their CPTs. All other variables that are child nodes store conditional probabilities in their respective CPTs. This is shown in Figures 1, 2 and 3.

3. PROPOSED ALGORITHM

We are currently working on algorithms to study the effect of relevant examples on student performance. A relational table of examples stores the domain and topic to which they belong, input fraction numbers, detailed solution of the example, its difficulty level and rating. Table 4 shows a sample table.

Example	Domain	Topic	Frac1_num	Frac1_den	Frac2_num	Frac2_den	Solution	Difficulty	Used	Rating
Ex1	Fractions	Add with CD	1	2	1	2	1/2 + 1/2 = 2/2 = 1	low	0	1
Ex2	Fractions	Add with CD	2	3	2	3	2/3 + 2/3 = 4/3 = 1 1/3	medium	0	1
Ex3	Fractions	Add with CD	1	4	2	4	1/4 + 2/4 =3/4	low	0	1
Ex4	Fractions	Add with DD	2	4	3	6	2/4 + 3/6 = (2*3 + 3*2) / 12 = 12 / 12 =1	high	0	1

Table 4: Sample example table (CD = Common denominator, DD= Different Denominator)

When a student uses the ITS for the first time, a new student model is created. This model stores the attributes describing the student and the initial Bayesian network with the respective probabilities for each node. A table of examples is also given as input to the student model. A student is recommended to use examples before attempting a certain activity or task. A comparison of the scores he/she achieves is then made. Thereby, if the student's skill level has increased, the rating of the example is increased. A student is then presented with examples of higher difficulty depending on the skill level he/she has achieved. This method has a two-fold advantage. Examples help students learn a skill better and faster. Secondly, from the logged table of examples, we mine the most useful example (measured by the rating) using an association rule (e.g., Apriori-like or FP-growth like) or sequential pattern algorithm. This information can then be accessed by other student models to help them pick more useful examples. As future work, we propose to use K-means algorithm to cluster examples into different groups so that examples that come before constructing the BN for the student model will ascertain the relationships between variables. Sequential pattern mining can also be applied to the examples to ascertain an effective order in which they are recommended to students. An algorithm is shown below.

Initial BN ↓ Example DB ↓

1. Student chooses a module (for example Add fractions with Common denominator (eg. current_module=ADD_CD))
2. If student wishes to see examples before attempting a task:
Until task is achieved or number of example exceeds 3, do
- show next example E from current_module
- student attempts task from current_module
- if task is done successfully,
rating of example E is incremented by 1
3. If number of example exceeds 3 and the task is still not achieved, then the tutor module is informed (and the student is asked to repeat the module).

4. EXAMPLE STUDENT MODELING

We used WEKA and Bayes Server to describe student attributes and their probabilities. The training dataset D consisted of 100 rows or instances and 10 attributes in a domain of adding fractions. WEKA classified 76% of our instances. There are two steps in creating a Bayesian network: creating DAG and assigning prior and conditional probabilities to each node. These steps can be either done manually by an expert or can be learnt automatically. We preferred to create the DAG manually and input it as an xml file to WEKA but use WEKA to compute the initial probabilities for each node using an input csv file with our dataset D. We used Bayes Server for creating and studying Bayesian networks. Bayes Server does not learn the structure of DAG automatically, but like WEKA, it computes the probabilities from a training data set given as an excel file. Figure 1 shows the DAG generated by Bayes Server for the training set given in table 2. Figure 2 shows the detailed CPT for node Add2Fracs where State1 = False and State2 = True.

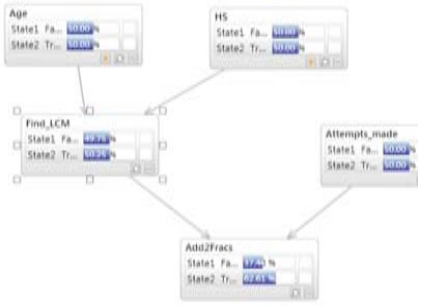

Figure 1: BN for table 2

Figure2: CPT for Add2Fracs

As evidence is introduced, the CPTs of each node get updated. Figure 3 shows the updated BN when evidence Age = False.

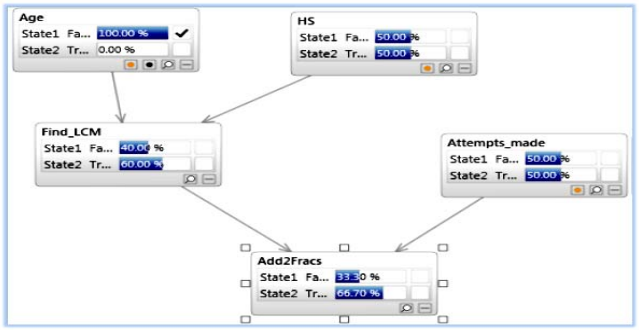

Figure 3: Updated BN (✓ indicates an evidence)

5. CONCLUSION

ITS Systems have progressed greatly in terms of adjusting instruction, using individualized strategies that are effective for students. However, there is little work done on how aids like examples help students improve their learning skills. This paper attempts to analyze the tools and techniques used in ITS's student models. Future work proposed is to mine up the most important example from several and present it to the student to enhance learning and understanding.

6. REFERENCES

[1] Brusilovsky, P. & Yudelson, M. (2008). From WebEx to NavEx: interactive access to annotated program examples. Proceedings of the IEEE, 96, 6, 990–999.

[2] Millán E, Loboda T and Pérez-de-la-Cruz J. 2010. Bayesian networks for student model engineering. Computers & Education 55(4): 1663-1683.