Bayesian student modeling and the problem of parameter specification

E. Millán(1), J.M. Agosta(2), J.L Pérez de la Cruz(1)

(1) Dept. de Lenguajes y Ciencias de la Computación. E.T.S.I. de Informática
Campus Universitario de Teatinos. Málaga 29080. Spain
e-mail: eva@lcc.uma.es

(2) Knowledge Industries
334 Arbor Drive
South San Francisco, CA 94080. USA
e-mail: johnmark@kic.com

Abstract: In this paper, the application of Bayesian networks to student modeling is discussed. A review of related work is made, and then the structural model is defined. Two of the most commonly cited reasons for not using Bayesian networks in student modeling are the computational complexity of the algorithms and the difficulty of the knowledge acquisition process. We propose an approach to simplify knowledge acquisition. Our approach applies causal independence to factor the conditional probabilities and decrease the parameters required for each question to a number linear in the number of concepts. This also provides the new parameters with an intuitive meaning that makes their specification easier. Finally, we present an example to illustrate the use of our approach.

Keywords: Bayesian networks, causal independence, student modeling.

1 INTRODUCTION

Intelligent Tutoring Systems (ITSs) are a particular case of knowledge based systems that are applied to teaching. Specifically, an ITS is an expert computer system that possesses deep knowledge about certain subject matter and guides students in acquiring this knowledge by means of an interactive process. Although there is not an accepted definition of what it means to teach in an intelligent way, a common feature to many existing ITS is that they use a model to infer the degree to which the student has mastered the subject knowledge. This model is used to determine the type of instruction needed by the student. The student model is the component of the ITS that represents the current state of the student’s knowledge, and the process that manipulates this structure is called diagnosis. The student model and the diagnostic process are so closely interrelated that they have been sometimes confused in ITS literature. The two components must be designed together, and this design problem is called the student modeling problem. In terms of Bayesian networks, the student model will be represented by the network structure and its probabilities, and the diagnostic process will be based on the diagnostic capabilities of Bayesian propagation algorithms.

Bayesian networks have already been successfully used to design expert systems for medical diagnosis; See for example Pathfinder (Heckerman et al 1989), and DIAVAL (Díez 1994), an expert Bayesian system for echocardiography. The medical diagnosis
problem consists of inferring a physiological state—for example, an illness—from a set of observable symptoms. For us, the student modeling problem is an exact analog, because it consists of inferring a hidden cognitive state—the student’s knowledge of the subject matter—from a set of observable data, for example, her answers to a set of questions posed by the system. Thus we contend that Bayesian networks can be used to handle the student model in a similar way that they have been used in the medical diagnosis problem.

One of the main reasons that has discouraged researchers from using Bayesian networks is the problem of the specification of the parameters. In fact, in real applications, a great number of conditional probabilities need to be specified, and, usually, they are quite difficult to estimate. In this paper, we suggest a method to simplify the specification of the parameters in the student modeling problem. Our approach consists in factoring the probabilities in such a way that the number of parameters required is much smaller, and the required new parameters are easier to estimate.

2 RELATED WORK

There has been some approaches to build probabilistic student models. We will briefly review some of them.

The first work that proposed the use of probabilistic student models is (Villano 1992). In this paper the application of Knowledge Space Theory (KST) and Bayesian belief networks as probabilistic student models in an ITS was suggested, and some ideas were offered for building and using such student models. Since then, Bayesian networks have been applied successfully to build student models in several applications:

• HYDRIVE (Mislevy and Gitomer 1996) models a student’s competence at troubleshooting an aircraft hydraulics system. The student performance is monitored by evaluating how she uses available information about the system to direct troubleshooting actions. The student model evaluates the student’s actions, and characterizes student understanding in terms of more general constructs (dimensional variables) that express the student’s knowledge of the system, strategies and procedures. A Bayesian network is used to express and update these student-model dimensional variables. Rule based inference still plays a complementary role in the system; thus they use a mixed approach. The initial values for the probabilities were set by qualitative input from expert instructors, and modified based on both simulated inputs and comparison between computed and expected outputs. Later work by Mislevy et al. 1999, pursue estimating model parameters from empirical data.
• ANDES (Conati et al 1997) is an ITS that teaches Newtonian Physics via coached problem solving. This system evolved from OLAE (Martin and VanLehn 1995) and POLA (Conati and VanLehn 1996), and uses Bayesian networks to do long-term knowledge assessment, plan recognition and prediction of student’s action during problem solving. The network is updated in real time, using approximate algorithms as students solve problems with ANDES. The Bayesian network is constructed
automatically from the solution graph associated with each problem, and the probabilities are obtained in the updating process. Since these networks are too large to be solved exactly, and exact propagation in Bayesian networks is known in the worst case to be NP-hard, approximate anytime algorithms based on stochastic sampling are used to update the network. The problem of the specification of the prior probabilities is specifically discussed in (VanLehn et al. 1998).

- In (Collins et al. 1996), Bayesian networks are applied to granularity hierarchies (McCalla et al. 1994). Assessment is the main purpose rather than diagnostic student modeling. They decompose the domain into learning objectives. The evidence used to evaluate whether the student masters a particular learning objective are her answers to related test items. To choose the most informative test item they use a utility measure that maximizes the change in expected probability for the learning objective being evaluated. Bayesian updating is used to propagate values from evidence to higher-level nodes. In the model used here, each objective is considered either mastered or not (binary valued nodes). For simulations with small networks of 11 or 15 nodes, 122 to 1345 conditional probabilities and 100 updating procedures were required.

Bayesian networks (BN) have also been used in user modeling. For example, Microsoft Office Assistant (see http://research.microsoft.com/~horvitz/lum.htm) uses Bayesian networks to emulate human experts in the task of understanding problems that users might be having with software from the user's behaviors, and, consequently, providing the right kind of help. More information of other applications of Bayesian networks in user modeling can be found in (Jameson 1996), where a complete review of systems that use Bayesian networks in student and user modeling is made.

We take the approach described in (Murray 1998). Murray discusses how the parameters were obtained in the Desktop Associate knowledge-based performance support system. The probability that the student has certain skill is computed using a Bayesian network from the student answers or actions. However, the approach discussed by Murray can assess only one skill at a time.

3 BUILDING THE STUDENT MODEL

A Bayesian network is a directed acyclic graph in which nodes represent variables and arcs represent probabilistic dependence among variables. The parameters used to represent the uncertainty are the conditional probabilities of each node variable given each combination of states of its parents; that is, if \( \{X_i, i = 1, \ldots, n\} \) are the variables of the network and \( Pa(X_i) \) represents the set of the parents of \( X_i \) for each \( i = 1, \ldots, n \); then the parameters of the network are \( \{P(X_i \mid Pa(X_i)), i=1,\ldots, n\} \), that is, the set of discrete conditional probability distributions of each variable given its parents. This set of probabilities defines the joint probability distribution for the entire network as,

\[
P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid pa(X_i))
\]
Thus, to define a Bayesian network, we have to specify:

- The set of variables, \( X_1, X_2, \ldots, X_n \).
- The set of arcs between those variables. These arcs represent causal influence between the variables. The network formed with these variables and arcs must be a Directed Acyclic Graph (DAG).
- For each variable \( X_i \), its probability conditioned on its parents, that is, \( P(X_i|Pa(X_i)) \), \( i=1,2,\ldots,n \).

For an easy introduction to BNs, see (Charniak 1991), and for a complete presentation (Castillo 1997).

If we are using Bayesian networks to define a student model, the variables can represent different things, depending on the domain. The variables can be rules, concepts, problems, or even abilities or skills. These variables are linked by causal influences between them. Once the links and the variables have been defined, it is necessary to specify the conditional probabilities. In the next section, we will present a method of specifying such parameters.

4 SPECIFYING THE PARAMETERS

Imagine that to answer a certain question, the student has to know \( n \) concepts. To construct the Bayesian network, we define the following variables:

- For each question, a variable \( Q \) that represents if the student is able or not to answer the question. \( Q \) takes two values: TRUE if the student answers the question correctly, and FALSE otherwise.
- For each \( i=1,2,\ldots,n \), a variable \( C_i \) that represents if the student knows concept \( i \). \( C_i \) takes also two values: TRUE if the student knows concept \( i \), or FALSE otherwise.

The Bayesian network that represents this information is (all nodes are binary):

![Bayesian Network Diagram]

Given a variable \( X \), we will represent ‘\( X \) is TRUE’ by \( x \) and ‘\( X \) is FALSE’ by \( \neg x \).

In general, the parameters required for this network are: the set of prior probabilities of \( C_1, C_2, \ldots, C_n \) and conditional probability of \( Q \) given \( C_1, C_2, \ldots, C_n \), i.e., \( n+2^n-1 \) parameters. In order reduce the number of parameters, to make the specification of the parameters easier, let us define two new parameters for each concept \( C_i \):

- \( s_i \) (slip) will be the probability that if a student knows a concept \( i \) she will fail while trying to apply it,
• $g_i$ (guess) will be the probability that the student guesses the correct answer to a question even when she doesn’t apply the concepts correctly.

Then, the probability $P(Q = q | C_1, C_2, ..., C_n)$ can be computed as follows:

$$P(Q = q | C_i = c_i \ i \in S \wedge C_j = \neg c_j j \notin S) = \prod_{i \in S} (1 - s_i) \prod_{i \in S} g_i$$

where $S = \{ i \in \{1, 2, ..., n\} \text{ such that } C_i = c_i \}$

**Proof:**

Let us create $n$ new fictitious nodes that we will call *concept application* nodes,

$$A_i = \text{the student is able to apply concept } i \text{ correctly } (i = 1, 2)$$

The new Bayesian network would be:

![Bayesian Network Diagram](image)

The conditional probabilities of the new variables $A_i, i=1,2,...,n$ are given in terms of the parameters $s_i$ and $g_i$:

$$P(a_i / c_i) = 1 - s_i \quad P(a_i / \neg c_i) = g_i \quad i = 1, 2, ..., n$$

The conditional probability of $Q$ given $A_1, A_2, ..., A_n$ is given by the AND truth table, that is,

$$P(Q = q | A_1, A_2, ..., A_n) = \begin{cases} 1 & \text{if } A_i = a_i \text{ for all } i = 1, 2, ..., n \\ 0 & \text{otherwise} \end{cases}$$

So the probability $P(Q = q | C_1, C_2, ..., C_n)$ can be computed as:

$$P(Q = q | C_1, C_2, ..., C_n) = \sum_{A_1, A_2, ..., A_n} P(Q = q / A_1, A_2, ..., A_n) \cdot P(A_1/C_1) \cdot P(A_2/C_2) \cdot ... \cdot P(A_n/C_n)$$

That is, if $S$ is the set of indexes $\{ i \in \{1, 2, ..., n\} \text{ such that } C_i = c_i \}$,

$$P(Q = q | C_1, C_2, ..., C_n) = P(a_1/C_1) \cdot P(a_2/C_2) \cdot ... \cdot P(a_n/C_n) = \prod_{i \in S} (1 - s_i) \prod_{i \in S} g_i$$
The interpretation of this result is easy. In order to solve a problem that involves \( n \) concepts, the student has not to fail applying those concepts she knows, and guess those concepts that she doesn’t know. So we can see that, in general, the \( 2^n \) conditional probabilities required for the distributions of the Q node can be obtained in a simple and intuitive way by multiplying the specified combinations of the \( 2n \) parameters.

A similar approach can be taken if any one concept is sufficient to solve a problem. Let us suppose that to solve a problem \( Q \) we can use either concept 1, or concept 2, ..., or concept \( n \). In this case, the probability can be computed as:

\[
P(Q=q / C_1, C_2, ..., C_n) = 1 - P(Q=-q / C_1, C_2, ..., C_n) = 1 - \prod_{i \in S} s_i \prod_{i \in S} (1 - g_i)
\]

Again, this result is easy to interpret, because in order to fail in solving the problem, the student has to fail in applying all the concepts that she knows and not to guess correctly any of the concepts that she doesn’t know.

The model defined in this application is a variety of a causal independence model (Heckerman 1993). Causal independence models are defined by a Bayesian network of the form that contains concept application nodes and a deterministic combining node. They contain the assumption that the causes work independently in generating the effect. The models defined here are extensions to the conventional “noisy-OR” and “noisy-AND” models: The conventional models have guess probabilities equal to zero and instead have a background or leak parameter associated with the effect (the problem in our case) that generates spurious TRUE states of the effect.

Our model with slip and guess parameters is equivalent to a reparameterization of the conventional noisy-OR. This can be shown by equating the expression above with the expression for the noisy-OR, where \( q_i \) is the link parameters, and \( q_0 \) is the leak probability:

\[
P(Q=q / C_1, C_2, ..., C_n) = 1 - \prod_{i \in S} (s_i) \prod_{i \in S} (1 - g_i) = 1 - q_0 \prod_{i \in S} (q_i)
\]

By considering \( n+1 \) of these equations, the solution for equivalent parameters of the noisy-OR are

\[
q_0 = \prod_{i \in S} (1 - g_i) \quad \text{and} \quad q_i = \frac{s_i}{(1 - g_i)}.
\]

Once the Bayesian network has been defined, the student model can be used in many different ways: to select the next topic to learn, to generate an adequate problem, to provide feedback or to select the most informative question to ask (as discussed in (Millán et al 1998)).
5 AN EXAMPLE

A simple example will be used to illustrate the problem and the approach. The domain in the example will be addition of fractions. The concepts/abilities that a student needs in order to solve a problem are:

- $C_1 = \text{Add whole numbers (numerators)}$
- $C_2 = \text{Simplify fractions}$
- $C_3 = \text{Find the least common multiple (LCM)}$
- $C_4 = \text{Find equivalent fractions (using the LCM)}$

We assume that these concepts are marginally independent; that is, knowledge of one concept does not necessarily imply knowledge of the others.

Different types of problems can be grouped attending to their difficulty. The more concepts the student needs to know to solve the problem, the more difficult the problem is. The types of problems and the concepts they require for solution are shown in the following figure:

<table>
<thead>
<tr>
<th>Type of Problem</th>
<th>Concepts involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1 (1/3 + 1/3)$</td>
<td>$C_1$</td>
</tr>
<tr>
<td>$Q_2 (1/4 + 1/4)$</td>
<td>$C_1, C_2$</td>
</tr>
<tr>
<td>$Q_3 (1/3 + 1/4)$</td>
<td>$C_1, C_3, C_2$</td>
</tr>
<tr>
<td>$Q_4 (1/3 + 1/6)$</td>
<td>$C_1, C_2, C_3, C_4$</td>
</tr>
</tbody>
</table>

To represent this information, we use this Bayesian network:

Thus, for this problem, we would need to specify the prior probabilities of each skill or ability and the conditional probabilities of the student being able to solve a certain type of problem given that she has or has not the skills necessary to solve it. That is, even for such a simple example we would need to specify 4 prior probabilities, and $2 + 2^2 + 2^3 + 2^4$ conditional probabilities, that is, 34 parameters. But not only the number of parameters required is important, but also the nature of these parameters. For example, one of these parameters would be the probability that a student is able to solve a type 4 problem, given that she knows how to add numerators and find the LCM, but does not know how to simplify or find equivalent fractions. Is it reasonable to suppose that experts (teachers) will be able to specify these parameters?. And if so, how good or accurate will their estimates be?

With the approach presented here, the only numbers that the teacher will need to provide would be:
• The prior probabilities that the student has each one of the abilities. These numbers can be estimated by teachers or taken from tests given to our particular group of students.

• The probability of a slip or guess by the student for each of the concepts involved. These probabilities have to be estimated by the teacher, and of course will be related to how difficult the concept is to apply to a problem. It will be easier to slip in concepts that involve difficult calculations, and to guess easy concepts.

We have reduced the complexity of the model to 12 instead of 34 numbers. Let us imagine that the teacher defines such parameters for our example:

<table>
<thead>
<tr>
<th></th>
<th>Prior</th>
<th>Slip (s)</th>
<th>Guess (g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add numerators</td>
<td>0.9</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>Simplify fractions</td>
<td>0.8</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>Find LCM</td>
<td>0.6</td>
<td>0.6</td>
<td>0.1</td>
</tr>
<tr>
<td>Find equivalent fractions</td>
<td>0.7</td>
<td>0.3</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Then, the conditional probabilities can be computed from these parameters. So for example, the probability that a student is able to solve a type 4 problem, given that she knows how to add numerators and find the LCM, but does not know how to simplify or find equivalent fractions is:

\[
P(Q_4 = q_4 / c_1, \neg c_2, \neg c_3, c_4) = (1-s_1) g_2 g_3 (1-s_4) = 0.9 \times 0.4 \times 0.1 \times 0.7 = 0.0252
\]

In this way, the 30 conditional probabilities can be computed by multiplying the appropriate combinations of the 8 slips and guesses.

Once the BN has been completely specified, it can be used to diagnose student’s state of knowledge and also to predict her behaviour. In the following table we show the evolution of the probabilities of knowing the concepts by giving a) the initial state b) the state after four questions (one of each type) have been asked and Q_1 and Q_3 were right and Q_2 and Q_4 were wrong c) the state after another four questions have been asked and answered with the same pattern as b)

<table>
<thead>
<tr>
<th></th>
<th>Initial state</th>
<th>After four questions</th>
<th>After eight questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_1</td>
<td>0.9</td>
<td>0.9611</td>
<td>0.9874</td>
</tr>
<tr>
<td>C_2</td>
<td>0.8</td>
<td>0.6194</td>
<td>0.3908</td>
</tr>
<tr>
<td>C_3</td>
<td>0.6</td>
<td>0.8408</td>
<td>0.9495</td>
</tr>
<tr>
<td>C_4</td>
<td>0.7</td>
<td>0.9345</td>
<td>0.9995</td>
</tr>
</tbody>
</table>

Table 1. Evolution of probabilities after eight questions

As we can see, the posterior probabilities that the student has each ability have increased, except for the ability to simplify fractions. That is because the performance of the student
in the problems involving this ability has been poor (the student has failed to answer questions 2 and 4). So after eight problems the probability that the student knows concept 2 is only 0.3908.

Also, prediction capabilities of BNs can give the probabilities that the student will answer each question correctly given her state of knowledge. Such predictions can be used to select which question to ask. For example, if after the first set of four questions we can see that the student does not know concept 2, we can select questions involving the minimum number of concepts (to avoid leakage) and concept 2 in order to determine as quickly as possible if the student knows this concept. Using this approach after the first four questions have been answered, we would ask the student question 2. Thus, if the student fails twice more on problems of the type 2, the prediction is now:

<table>
<thead>
<tr>
<th>Initial state</th>
<th>After four questions</th>
<th>After two Q2 questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>0.9</td>
<td>0.9611</td>
</tr>
<tr>
<td>$C_2$</td>
<td>0.8</td>
<td>0.6194</td>
</tr>
<tr>
<td>$C_3$</td>
<td>0.6</td>
<td>0.8408</td>
</tr>
<tr>
<td>$C_4$</td>
<td>0.7</td>
<td>0.9345</td>
</tr>
</tbody>
</table>

Table 2. Evolution of probabilities after six questions

So after only two questions more we would have a very low probability (0.2639) that the student knows concept 2. As we can see this approach can reduce the number of questions necessary to determine which concept the student is having troubles with so we can provide remediation as soon as possible.

6 CONCLUSIONS

Bayesian modeling techniques provide a rigorous formal approach to student modeling in contrast to ad-hoc approaches that, lacking a formal basis, make the results quite unpredictable. If a Bayesian-based system behaves incorrectly or unexpectedly, we know that this misbehavior is not due to the inference mechanism, but to the assumptions in the model. Numbers obtained by the system might be inaccurate, but not inconsistent. For us, the soundness of a Bayesian student model may be worth the effort to develop it. We have tried to promote the use of probabilistic reasoning techniques to handle this delicate problem by proposing an approach to simplify the knowledge acquisition process. Our approach not only reduces the number of parameters required, but also provides the new parameters with an intuitive meaning that makes its estimation easier.
REFERENCES


Note about the authors

Ms. Eva Millán is T.E.U. professor in the Department of Computer Sciences at Málaga University (Spain), where she lectures on Operations Research and Expert Systems. From April to September 1998 and 1999 she worked as International Fellow at SRI International, California (formerly Stanford Research Institute), where she started the collaboration with Dr. John Mark Agosta. Her research is aimed to the application of AI Techniques to Intelligent Tutoring Systems.

Dr. John Mark Agosta is the Chief Technical Officer at Knowledge Industries, a supplier of Bayes Network diagnostic software tools. Previously, from 1992 to 1998, he worked in Applied AI at SRI International, California (formerly Stanford Research Institute). He joined SRI after completing his doctorate in Engineering Economic Systems in Stanford's School of Engineering. His work includes probabilistic reasoning applied to diagnosis, planning and image understanding.

Dr. J.L. Perez-de-la-Cruz is T.U. professor in the Department of Computer Sciences, at Málaga University (Spain), where he lectures on Artificial Intelligence. His research is aimed to the application of Artificial Intelligence to Engineering problems and to Intelligent Tutoring Systems.