Student Modelling based on Belief Networks

Jim Reye

International Journal of Artificial Intelligence in Education 14 (2004) 1–33
IOS Press

Student Modelling based on Belief Networks

Jim Reye, *School of Information System, Queensland University of Technology, Brisbane, Australia*
j.reye@qut.edu.au

Abstract. Belief networks provide an important way to represent and reason about uncertainty – significant factors for modelling students. These networks provide a way of structuring such models, and allow a system to use a systematic approach when gathering information about the scope of the student's knowledge. This work also provides a theoretically-sound way to update the student model, based on the concept of a dynamic belief network. The relationship to related research is discussed. Finally, the paper describes why the barren node concept is important for computational efficiency in belief-net-based student models.[1]

INTRODUCTION

Systems that try to model the student's understanding of subject material face a number of major issues. One of these is the inherent lack of certainty as to how much the student understands, at any given point in time. Even though there may be surface evidence that the student does (or does not) know a particular topic in the subject domain, this evidence may just reflect a lucky guess (or a temporary slip). Systems that are able to weigh each new item of evidence, in conjunction with (any) previous evidence about the student's state of knowledge, have a firmer foundation for making pedagogical decisions than those systems which ignore this issue.

While it is possible to design a system in which the student model is merely a collection of isolated, independent beliefs, this is clearly an unrealistic model for many domains. For example, in the database language SQL, it is extremely unlikely that a student would be familiar with the "having" clause while being unfamiliar with the "group by" clause. To model such interdependencies of knowledge, an ITS system could have an ad hoc set of rules so that a change in belief, in one part of the student model, leads (via the rules) to corresponding updates in one or more other parts of the student model. However, rule-based approaches to handling uncertainty usually result in a set of rules that produce inconsistent results.

To avoid such problems, we need a better approach. Fortunately, belief networks (Pearl, 1988; Villano, 1992) provide a foundation for building interdependencies of knowledge into the student model itself. Not only does this automatically guarantee consistency of beliefs, but it is also pleasing to have such knowledge (about interdependencies) as part of the same modelling technique (rather than needing a separate set of rules).

[1] Preliminary versions of this work have appeared in Reye (1996) and Reye (1998).

This paper describes the use of belief networks from several perspectives, commencing with the section "Structuring Knowledge about Related Beliefs" that explains why belief networks are appropriate for modelling an ITS's beliefs about a student. That section also provides a limited introduction to the concepts of belief networks for those who are unfamiliar with them, but such readers should also refer to chapters 14 and 15 of Russell and Norvig (1995) for a much more comprehensive introduction. On the other hand, the reader who is already knowledgeable about belief networks can skip most of that section without loss.

The section "Making use of a Belief-Net-Based Student Model" shows how such networks can be applied to efficiently gather information about the student's current state of knowledge, as well as showing how to model students as (somewhat) unreliable sources of information. Again, the reader who is already knowledgeable about belief networks can skip most of that section without loss. The next section introduces the concept of a belief net backbone, as a way of integrating the ideas described in the second and third sections. In the section "Updating the Student Model: Dynamic Belief Networks", I describe how the updating of the student model should be modelled as a dynamic belief network, and show how this updating relates to the previous work of Corbett and Anderson (1992) and Shute (1995). The section "Computational Efficiency in a Belief-Net-Based Student Model" focuses on computational efficiency by describing why the barren node concept is important in belief-net-based student models.

STRUCTURING KNOWLEDGE ABOUT RELATED BELIEFS

In this section, I explains why belief networks are appropriate for modelling an ITS's beliefs about a student. In general, such beliefs should not be entirely independent of each other. In an ITS, an important interdependency relationship is that representing prerequisites.

The importance of the prerequisite relationship, for structuring beliefs

For tutoring by humans, the prerequisite relationship is clearly a very important one, both for instructional planning purposes and for gathering information about the current state of a student's knowledge. In discussing the approaches of human instructors, Collins and Stevens (1982) state:

> Rather we assume only a partial ordering on the elements in the teacher's theory of the domain. ... The teacher's assumption is that students learn the elements in approximately this same order. Therefore, it is possible to gauge what the student will know or not know based on a few correct and incorrect responses. These responses are used to determine a criterion point in the partial ordering; above this point, the student is likely to know any element and below it, the student is unlikely to know any element.

How should we model such a partial ordering? Where knowledge of topic A is a prerequisite for knowledge of topic B, there are two aspects of this relationship that we wish to model.

(a) Firstly, the obvious constraint that *lack of a student's knowledge* of A implies lack of knowledge of B. In terms of predicate logic, we can express this as:

$$\neg\text{student-knows}(A) \Rightarrow \neg\text{student-knows}(B)$$

or equivalently:

$$\text{student-knows}(B) \Rightarrow \text{student-knows}(A)$$

(b) Secondly, the more subtle reasoning that *evidence of a student's knowledge* of A can be taken as evidence for revising our belief that the student also has knowledge of B.

Where there is a close relationship between A and B, we may wish to assert that it is more likely that the student's "frontier of knowledge" includes both A and B, compared to falling between A and B. For example, in SQL, understanding the "group by" clause is a prerequisite for understanding the "having" clause, and because of their close relationship, if we encounter a student who already knows about the "group by" clause, then it is more likely that they know about the "having" clause. In such circumstances, evidence for knowledge of A increases our belief that the student also has knowledge of B.

On the other hand, where there is a weaker relationship between A and B, we may wish to make no such assertions. For example, in SQL, understanding the "select" statement is a prerequisite for understanding the "view" statement, but knowledge of the former is little or no evidence for knowledge of the latter (for students in the process of learning SQL).

As shown above, aspect (a) can be modelled using predicate logic. But, aspect (b) involves reasoning under uncertainty. Even when knowledge of A makes it highly likely that the student also knows B, we cannot be absolutely certain. Modelling this reasoning as part of the student model enables all reasoning about uncertainty to be done within the student model. Belief networks provide a foundation for representing prerequisite relationships, in a way that satisfies both aspects (a) and (b) above, and automatically guarantees consistency of beliefs about the student's current state of knowledge.

Over the next few pages, I describe an approach using belief networks to formally model prerequisite relationships.

Representing simple prerequisite relationships as probabilistic relationships

As described in the preceding section, where knowledge of topic A is a *prerequisite* for knowledge of topic B, there are two aspects of this relationship that we wish to model.

Firstly, the obvious constraint that *lack of a student's knowledge* of A implies lack of knowledge of B. In probabilistic terms, it is inconsistent to assert that, at any single point of time:

$$\text{both:} \quad p(\text{student-knows}(A)) = 0$$
$$\text{and:} \quad p(\text{student-knows}(B)) = 1$$

In probability theory, this constraint is represented as a *conditional probability*. That is, the probability of a proposition being true is dependent upon what is known about the probability of other propositions. This is written (with the conclusion first) as:

p (student-knows (B) | ¬student-knows (A)) = 0

This may be read formally as "the probability that student-knows (B) is true, *given* that student-knows (A) is not true, *is* 0", or informally as: "You can't know B, if you don't know A." (From an instructional planning perspective, this formula can also be interpreted as specifying that lack of knowledge of topic A is an inhibitor of topic B, which must be removed before B can be covered.)

The above conditional probability can also be expressed in a variety of logically-equivalent forms (by using elementary probability theory). These are:

p (¬student-knows (B) | ¬student-knows (A)) = 1
p (student-knows (A) | student-knows (B)) = 1
p (¬student-knows (A) | student-knows (B)) = 0
p (¬student-knows (A), student-knows (B)) = 0

(where the "," immediately above is read as "and").

A less formal way to see that these are logically equivalent is to turn them back into natural language sentence. For example, the second of these forms may be read as: "If you know B, then you must also know A"; and the last form may be read as: "You can't not know A and still know B."

Although the above description was only in terms of a pair of related topics, conditional probabilities allow the specification of relationships that are more complex than those given above. For example, we can specify that both P and Q are prerequisites for R, as:

p (student-knows (R) | ¬student-knows (P), student-knows (Q)) = 0
p (student-knows (R) | student-knows (P), ¬student-knows (Q)) = 0
p (student-knows (R) | ¬student-knows (P), ¬student-knows (Q)) = 0

This set of conditional probabilities is equivalent to the predicate logic formula:

student-knows (R) ⇒ (student-knows (P) ∧ student-knows (Q))

The *second* requirement is the more subtle reasoning that *evidence of a student's knowledge* of A can be taken as evidence for revising our belief that the student also has knowledge of B. This can be specified by assigning an appropriate value to the conditional probability:

p (student-knows (B) | student-knows (A))

For example, if we let:

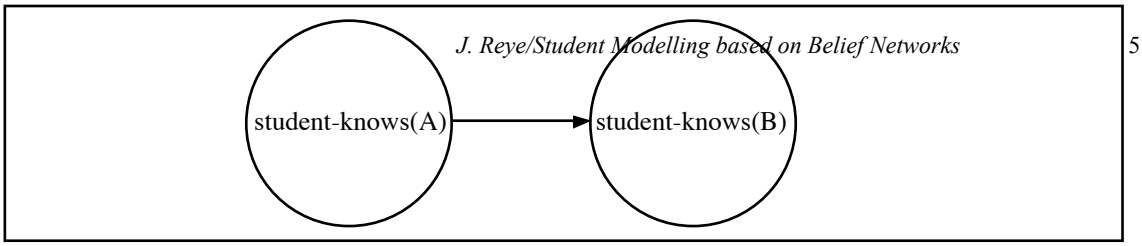

Fig. 1. A simple example of a belief network (showing a prerequisite relationship)

$$p\ (\text{student-knows}\ (B)\ |\ \text{student-knows}\ (A))\ =\ 0.95$$

then this models a close relationship between knowledge of A and B. That is, a student who knows A is highly likely (95%) to also know B.

On the other hand, if we estimate that the prior probability of student-knows (B) is 0.01 and we wish to assert that knowledge of A has no impact on the likelihood of knowledge of B, then we simply specify this same value for the conditional probability:

$$p\ (\text{student-knows}\ (B)\ |\ \text{student-knows}\ (A))\ =\ 0.01$$

So, probability theory allows us to represent a range of relationships, from strong to weak, between knowledge of A and B, according to our modelling needs.

From prerequisite relationships to a belief network

As hinted at by the above examples, the concept of a conditional probability may be regarded as a generalisation of material implication ( ) in traditional logic. This is because it provides a basis for reasoning, while not being restricted to cases where a proposition is known to be true or false with certainty.

Belief networks provide a graphical way of designing probabilistic models based on the concept of conditional probability. Figure 1 is a simple example.

Rather than just providing a picture, the structure of a belief network is used for automated reasoning about uncertainty, in the most efficient manner that applies to that structure (see Pearl, 1988). A detailed discussion of the many properties of belief networks is beyond the scope of this paper.

Because of the (previously mentioned) logical equivalence of the two formulae:

$$p\ (\text{student-knows}\ (A)\ |\ \ \text{student-knows}\ (B))\ =1$$
$$p\ (\text{student-knows}\ (B)\ |\ \neg\text{student-knows}\ (A))\ =0$$

it is possible to represent "student-knows (A)" and "student-knows (B)" as two nodes in a belief network, either with a directed arc from "student-knows(B)" to "student-knows (A)" or with an arc in the reverse direction.

Here, I have chosen the latter approach, i.e. an arc from "student-knows (A)" to "student-knows (B)". This choice results in a belief network in which the arc-directionality is in tune with the sequence in which the topics must be learned. In other words, the direction of these arcs provides a representation of the partial-ordering on the sequence in which topics must be learned. Consequently, the designer of such a belief network for a particular domain may proceed by first creating that partial-ordering (ignoring uncertainty) and then using it as the backbone for a belief network.

There is another, more subtle reason for preferring to draw the arcs in this direction. Belief networks are easier to create and understand when the arcs represent causal relationships (Pearl (2000)). As a general example, it is more useful to model the relationship between clouds and rain by drawing an arc from “clouds” to “rain”, because clouds cause rain, not the reverse. At first, causality seems inappropriate when modelling the prerequisite relationship, because knowing topic A does not cause knowledge of topic B. However, when causality is seen not just to involve factors that have a positive influence, but also factors that have a negative influence, then not knowing topic A is a cause for not knowing topic B. That is, these two topics are causally related in an *inhibitory* sense, from topic A to topic B. So the arc should be drawn in this direction, not the reverse.

In the above example, there is no uncertainty in the constraint itself. It can be interpreted as "All students who know B, also know A." Being based on probability theory, belief networks also allow the representation of constraints that are less certain.[2] For example:

p (student-knows (A) | student-knows (B)) = 0.9

can be interpreted as "Most students who know B, also know A." Being able to represent and reason with such knowledge is a valuable advantage over approaches based on traditional logic alone.

Although the preceding description was only in terms of a *pair* of related topics, the use of conditional probabilities easily extends to much longer chains and networks of prerequisite dependencies. For example, if we know that A is a prerequisite for B, which is a prerequisite for C, and so on up until Z (say), then if we discover that the student knows Z, then we don't have to ask about the earlier prerequisites. Likewise, if we find out that the student knows F, but not G, then we don't have to ask about A..E or H..Z.

As well as numeric values for the conditional probabilities, we also must specify the prior probabilities of all propositions that are not determined by the conditional probabilities. These prior probabilities specify the system's initial set of beliefs about a (typical) student, prior to the first interaction with that student. As the student uses the system, it directly updates its beliefs about the student's knowledge of topics, where these are observable. (In the simplest case, the probabilities of these topics will be forced to 1 or 0, if it is clear that the student either knows or doesn't know the

[2] Conditional probability values, such as 0.9, are not shown directly on the belief network. i.e., an arrow from one node to another shows that there is a relationship, but not how strong it is. To fully specify a belief network, a set of conditional probabilities – that match the structure of the network – must be specified separately.

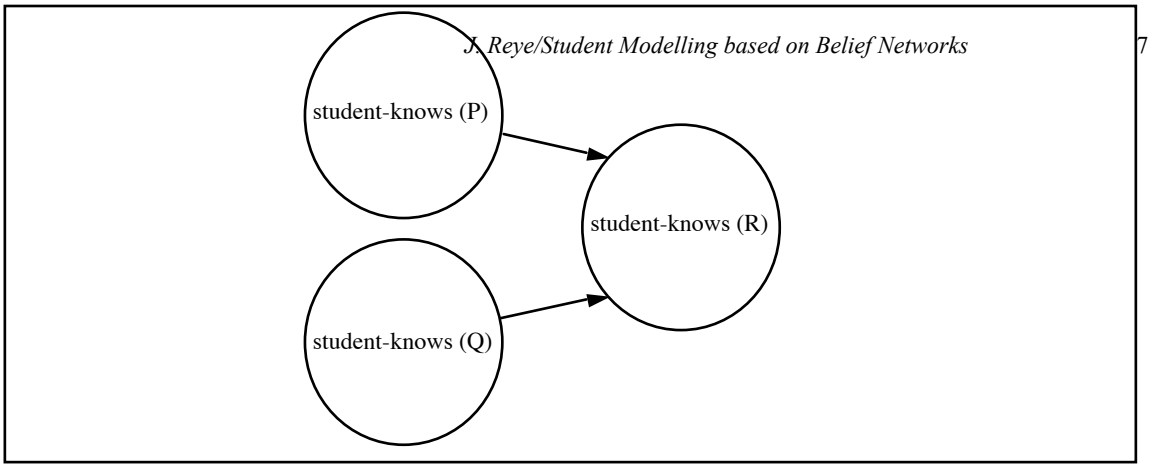

Fig. 2. A topic with two prerequisites

topic.) These changes in belief are then propagated through the belief net, changing the system's belief in the likelihood that the student knows other (as yet) unobserved topics.

Figure 2 shows another simple belief network, in which P and Q are prerequisites for R, corresponding to the second example of conditional probabilities in the previous section.

MAKING USE OF A BELIEF-NET-BASED STUDENT MODEL

The preceding part of this paper argued that a belief network is an appropriate way to model knowledge of the student, based on the need to represent uncertain knowledge. In this part of the paper, we look at further advantages of belief networks for student modelling, by showing how they can be used to support two important ITS tasks:

(a) efficient gathering of information about the student's current state of knowledge; and

(b) modelling students as (somewhat) unreliable sources of information.

Efficient gathering of information about the student's state of knowledge

The earlier quote, from Collins and Stevens, describes how human teachers are able to gauge the extent of a student's knowledge based on a small number of probing questions. This section shows how the structure of a belief-network based student model supports the gathering of information about a student's knowledge, while minimising the number of probing questions required.

Collins and Stevens's teaching strategy can be modelled as a problem-solving procedure within the framework of a classic AI diagnostic task. Although such AI research has typically been concerned with the development of efficient procedures for the automated troubleshooting of faulty mechanical systems, electronic circuits and computer programs, such research can be applied to student modelling by regarding the student's knowledge as being faulty (while learning is occurring), in the sense that certain inputs do not produce the correct outputs.

In particular, there is an analogy between the prerequisite structure of the domain (modelled as part of the belief network) and a digital circuit. Each "student-knows (*topic*)" can be regarded as an AND gate which has one input for each prerequisite topic (except for the base topics which have no prerequisites). Such a gate only outputs a 1 if all its inputs are 1s (analogously, all prerequisites are known) and it is working properly (analogously, that particular topic is known). When the circuit is "functioning normally", all gate outputs are 1, i.e. the student knows everything.

When there is a fault, at least one gate output is 0, leading to zeros in all the gates that follow. Analogously, if the student doesn't know a topic, then he doesn't know the follow-on topics. In other words, the student's lack of knowledge of a topic can be seen as a fault which disables the correct functioning of later parts of the (partially ordered) graph of all the prerequisite knowledge.

Some AI research into diagnosis assumes the simplest case that only a *single* fault must be found. Because a student may initially know very little about a domain (i.e. have many faults), previous work on diagnosing *multiple* faults is more useful. In particular, de Kleer and Williams's (1987) research behind their General Diagnostic Engine (GDE) system is helpful. Self (1993) describes the use of GDE for (non-probabilistic) student modelling, including an interesting application of this approach: diagnosing student's attempts at solving three-column subtraction.

The GDE approach is based on the idea of minimising the number of measurements (analogously, minimising the number of questions asked of the student) by making a series of measurements, each of which maximises the expected amount of information gained by that measurement. Technically, minimising the expected entropy (H) of the belief network after making that measurement:

$$H = -\sum p_i \log p_i$$

So, the expected entropy (H_e) after asking a student whether they know topic t_n is given by the weighted sum of the two possible responses:

$$H_e(sk(t_n)) = p\,(sk(t_n))\,H\,(sk(t_n)) \;+\; p\,(\neg sk(t_n))\,H\,(\neg sk(t_n))$$

where "sk" is an abbreviation of the "student-knows" predicate.

One of the difficulties faced by the GDE procedure is that the number of possible combinations of faults grows exponentially with the number of components. Fortunately, for student diagnosis, the number of possible combinations is far less. This is because, whenever an ITS considers a possible diagnosis involving a particular faulty node, then all subsequent (partially ordered) nodes must also be faulty (at any one point in time). By comparison, in an electronic circuit, subsequent nodes need not be faulty and so there are more cases to consider. For example, consider a simple chain of four items:

A → B → C → D

When this chain is read as representing an electronic circuit of buffers, then there are 16 ($=2^4$) possible combinations of possibly faulty components.

But, when this chain is read as a belief net representing the prerequisite relationships linking four topics, then there are only five combinations which model possible states of the student's knowledge, as given by the following sets: {}, {A}, {A, B}, {A, B, C}, {A, B, C, D}. Such linear growth is clearly better than exponential growth, especially when creating domains containing hundreds of topics.

A small example now illustrates the approach. (As above, I use "sk" as an abbreviation for the "student-knows" predicate.)

An example:

A → B → C → D

with:

$p(sk(A))$	$= 0.75$	$\Rightarrow p_{prior}(sk(A)) = 0.75$
$p(sk(B) \mid sk(A))$	$= 0.75$	$\Rightarrow p_{prior}(sk(B)) = 0.56$
$p(sk(C) \mid sk(B))$	$= 0.75$	$\Rightarrow p_{prior}(sk(C)) = 0.42$
$p(sk(D) \mid sk(C))$	$= 0.75$	$\Rightarrow p_{prior}(sk(D)) = 0.32$

There are four topics, so there are four possible questions which could be asked. The expected entropy for each of these possibilities are calculated as:

$H_e(sk(A))$	$= p(sk(A))\, H(sk(A)) + p(\neg sk(A))\, H(\neg sk(A))$	$= 0.98$
$H_e(sk(B))$	$= p(sk(B))\, H(sk(B)) + p(\neg sk(B))\, H(\neg sk(B))$	$= 0.67$
$H_e(sk(C))$	$= p(sk(C))\, H(sk(C)) + p(\neg sk(C))\, H(\neg sk(C))$	$= 0.69$
$H_e(sk(D))$	$= p(sk(D))\, H(sk(D)) + p(\neg sk(D))\, H(\neg sk(D))$	$= 0.95$

These values confirm what one would expect intuitively in this case, i.e. that more information is gained by asking about B or C, rather than A or D. More precisely, topic B has the lowest expected entropy (i.e. highest expected gain of information) and so should be asked first. The student's reply can then be used to update the probabilities in the student model. These revised probabilities can then be used in subsequent calculations of expected entropy, in order to determine which topic should be queried next.

This minimum expected entropy procedure is easily incorporated into a system, so that it can pursue a series of dialogue goals aimed at minimising the uncertainty in the student model, while minimising the number of questions asked. The resulting dialogue matches the behaviour of the human teachers described previously. (Such goals may be pre-empted by other goals of higher priority, both in an ITS and in a human.)

Having described how belief networks can be used to assist in the gathering of information about a student's knowledge, we now turn to the issue of how to handle the fact that students are not always entirely reliable sources of information about their own state of knowledge.

Modelling students as (somewhat) unreliable sources of information

Clearly, an ITS must make use of the student's actions and responses to questions, in order to obtain information about the student's current knowledge. But, there is a catch. This source of information is *not always reliable*: students sometimes make lucky guesses, sometimes make slips and sometimes even give wrong answers deliberately (e.g. to see how the system will react).

Consequently, it is undesirable that an ITS uses such information to *directly* update its beliefs about the student's current knowledge, For example, if the student gives the correct answer to a multiple-choice question, then the ITS should not simply assume that the student knows the correct answer. (This is especially important when the prerequisite relationship is considered. We wouldn't want a single lucky guess to result in the system believing that the student also knew all the prerequisite material.)

Instead, the system should weigh up all the evidence it has, including the student's response, to decide how likely it is that the student knows the correct answer. This requires reasoning under uncertainty, so a belief network is a better way to do this, rather than requiring a special set of diagnostic rules.[3]

To explain this, we start by modelling the simplest possible case, a domain containing only a single topic. Figure 3 illustrates this, where the upper node (the *learned* state) represents the system's belief that the student knows the domain topic; the lower node (the *outcome*) represents the evidence gained when the student is assessed on that topic; and the arrow represents a (probabilistic) causal relationship, i.e. the student is more likely to give the correct answer if they know the topic. As before, this relationship is represented as a conditional probability.

In this model, the learned state node has only two (unobservable) values: *true* and *false*, but the outcome node either has two values (e.g. *correct* and *incorrect*) or more than this (e.g. representing various levels of hints which the student can request before being able to give the correct answer). Having more than two outcomes has no impact on the *structure* of the model, although it does affect the number of conditional probability values which must be specified.

When looking at Figure 2, it is important to be aware that the direction of the arrow indicates the natural direction of the causal link, but does restrict the direction of information flow. In other words, given an observed outcome, the system revises its belief about the learned state node, even though it may be regarded as having information flowing in the opposite direction to the arrow.

---

[3] See chapter 1 of Pearl (1988) for a discussion of the limitations of rule-based systems for reasoning under uncertainty.

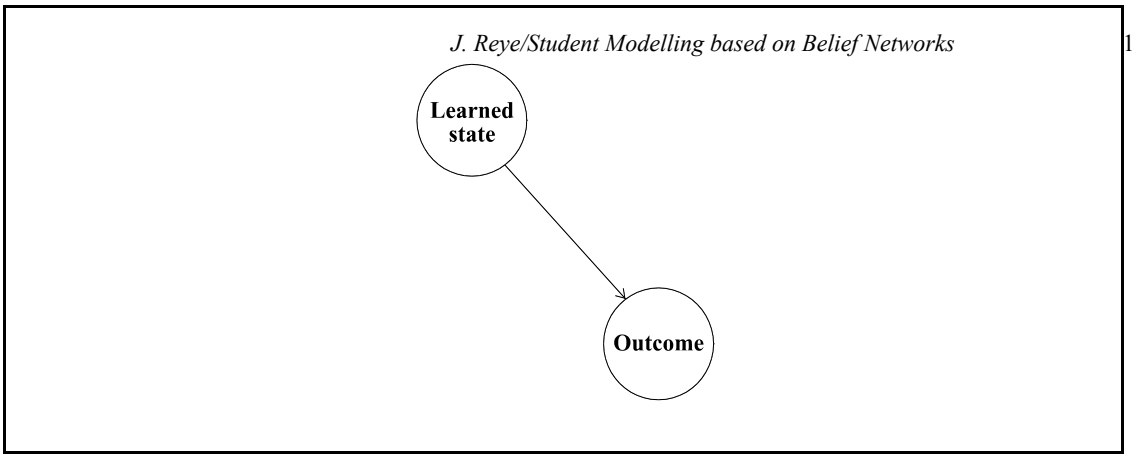

Fig. 3. Using a belief network to model an unreliable source of information

Later in the paper, I give the mathematical formula that specifies how such updating occurs, but we start with two simple examples that illustrate the basic nature of such updating. For the sake of these examples, assume that:

(a) the conditional probability of a correct outcome when in the learned state is 0.95 (allowing for the occasional slip), i.e.

p (outcome = correct | learned) = 0.95; and

(b) the conditional probability of a correct outcome when in the unlearned state is 0.20 (allowing for lucky guesses), i.e.

p (outcome = correct | ¬learned) = 0.20.

Example 1

In this example, let the initial ("prior") probability of the learned state being true be 0.5, i.e. we really have no initial idea as to whether the student knows the topic or not. From this, we can calculate that the initially expected probability of a correct outcome is 0.575.

Case 1: If the student then gives a correct response, the revised ("posterior") probability that of the learned state is approximately 0.78. This value is higher than the initial value, but is not extremely high, because there is a substantial chance of lucky guesses.

Case 2: If the student instead gives an incorrect response, the revised ("posterior") probability that of the learned state is approximately 0.06. This value is much lower than the initial value and is therefore a fairly good indication that the student is in the unlearned state (although there is still a small chance that the student's response was just a slip).

Example 2

In this example, let the initial ("prior") probability of the learned state being true be 0.9, i.e. we are fairly confident that the student already knows the topic. From this, we can calculate that the initially expected probability of a correct outcome is 0.875.

Case 1: If the student then gives a correct response, the revised ("posterior") probability that of the learned state is approximately 0.98, further increasing our confidence.

Case 2: If the student instead gives an incorrect response, the revised ("posterior") probability that of the learned state is approximately 0.36, a big drop in confidence, because the likelihood of a slip is fairly small.

These examples illustrate the basic use of belief networks for revising beliefs in the light of new evidence. But clearly, a domain containing only one topic is not of any practical use. An obvious expansion of this technique, from just a single domain topic to *n* topics, is to simply replicate the structure, in Figure 3, *n* times. While straightforward, the resulting model ignores the prerequisite relationship, which is an important part of the student model. In the next section, we see how these two desirable aspects, prerequisites and indirect evidence, can be incorporated into the same student model.

A BELIEF NET BACKBONE

Introduction

As mentioned previously, a belief network is a set of beliefs, together with a set of conditional probabilities linking those beliefs. In the general theory of belief networks, there are no restrictions on the structure of the network, apart from the prohibition of directed cycles. However, in any system that uses a belief network, an actual structure must be specified.

In this paper, it is proposed that the appropriate belief network structure for an ITS is based on three connected ideas:

(a) a *belief net backbone*, which links *all* the "student-knows (*topic*)" nodes together in a partial ordering, according to their prerequisite relationships;

(b) a *set of topic clusters*, each of which comprises a *single* "student-knows (*topic*)" node, together with a set of additional belief nodes for the purpose of modelling factors that directly or indirectly relate to the system's belief that the student knows the topic, e.g. "when-opportune--student-demonstrates-usage-of (*topic*)";

(c) a (relatively-small) *set of global nodes* that represent the system's belief in overall student characteristics, i.e. beliefs not focussed on a specific topic, e.g. "student-is-bored ()" and "student-overall-aptitude ()".

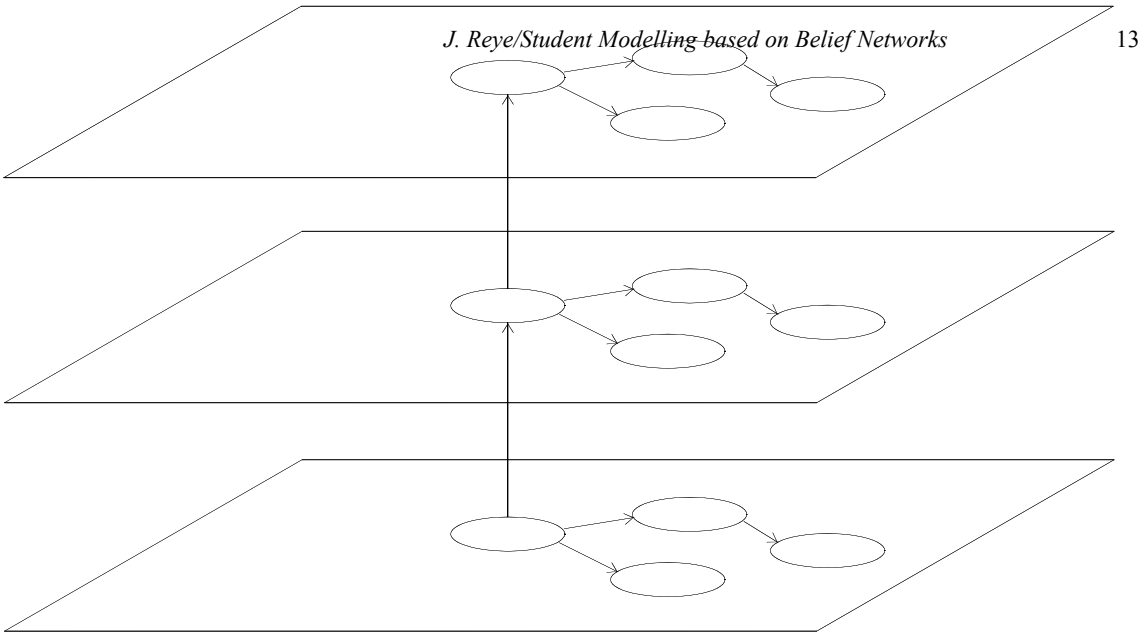

Fig. 4. A belief net backbone

Figure 4 presents a very simple example of this approach, in which the global nodes are not shown, for clarity of presentation. In this figure, the backbone is represented by the vertical arrows, and each topic cluster is represented by the four nodes in each horizontal plane. In comparison, Figure 5 gives a more detailed example of a topic cluster, showing a larger number of nodes. (The detailed meanings of the nodes and the links, in Figure 5, are explained in the section that follows.)

As Figure 4 shows, the backbone and the topic clusters intersect in a deliberately restricted way, i.e. each of the "student-knows" nodes occurs once in the backbone and once in its own topic cluster. So, in an example domain with 100 topics, there will be: (i) one instance of the backbone, containing 100 student-knows nodes; (ii) 100 instances of the topic cluster, each with the same structure, but not the same probability values; and (iii) one instance of the set of global nodes.

For ease of reference in the following, the nodes in the topic clusters are often referred to as *local nodes* (excluding the "student-knows (*topic*)" node). This is because such nodes primarily have only local importance, i.e. primarily affect the system's belief in the cluster's *single* "student-knows (*topic*)" node, unlike the backbone nodes and the global nodes, which can affect the system's belief in *many* "student-knows (*topic*)" nodes.

This backbone approach to student modelling has two advantages. *Firstly*, it gives the designer *a standard methodology* for creating the structure of an ITS belief network, regardless of the particular domain.

Secondly, there are *computational advantages* in that updates to the beliefs in any one topic cluster only affects the other topic clusters via the backbone, rather than there being any direct connection. In particular, this means that the impact of belief updates in a given topic cluster on its "student-knows" node can be calculated locally by considering just the nodes in that topic cluster,

rather than having to propagate such updates through the entire network in order to determine their nett results. The efficiency gained by such local computation is very important during planning, when the impacts of large numbers of possible plans must be determined rapidly. (Once a plan has been chosen and is executed, its effects update one or more topic clusters, and these effects must be propagated through the backbone. Although computationally more expensive, such updates occur much less frequently than those needed during planning.)

The concept of a *belief net backbone* is fairly straightforward, as it only ever involves links between "student-knows" nodes, and those links only ever involve the prerequisite relationship. Consequently, I do not discuss it further. In comparison, the concepts of *topic clusters* and *global nodes* are more complex, because they involve other types of nodes and other types of relationships. So, in the following sections, I give a more detailed description of the types of nodes and relationships that occur within typical topic clusters and with global nodes.

Topic clusters and their local nodes

As described in the preceding section, and illustrated by the example in Figure 5, each topic cluster comprises a single "student-knows (*topic*)" node, together with a set of additional belief nodes for the purpose of modelling factors that directly or indirectly relate to the system's belief that the student knows the particular topic, e.g. "when-opportune-student-demonstrates-usage-of (*topic*)".

Although the "student-knows (*topic*)" node is very important for determining what an ITS should say and do, the system *cannot* directly observe whether this node is true or false, with 100% accuracy, but must instead make inferences about how likely it is that a particular student knows a given topic, based on the observations that *can* be made, e.g. whether a student *claims* to have such knowledge. Supporting such inferencing is the primary purpose of such nodes (although some have other usages, such as providing a basis for determining how well motivated a student is).

To support such inferencing, it is necessary to introduce additional relations (or "nodes") that are not present in the domain-model. It is expected that the same topic cluster *structure* be used for all topics in the student model, i.e. uniformly across the entire student model. (This does not mean the same *level of belief* across the entire student model, because each topic cluster has its own associated probabilities.)

This uniformity of structure is not required by the theory of belief networks. Rather, it is intended as a relatively low design-cost approach to constructing student models that support reasoning about the student's knowledge. As section "Updating the Student Model: Dynamic Belief Networks" shows later, it does no computational harm if some nodes in some clusters are never used. So there is no reason to adopt a non-uniform approach because of concerns for run-time efficiency.

On the other hand, I do not claim that an identical topic cluster structure be used *across* all ITSs. The different user interfaces in different systems provide different opportunities (and restrictions) on what information can be gathered about the student's knowledge, and so the topic cluster structure should reflect these.

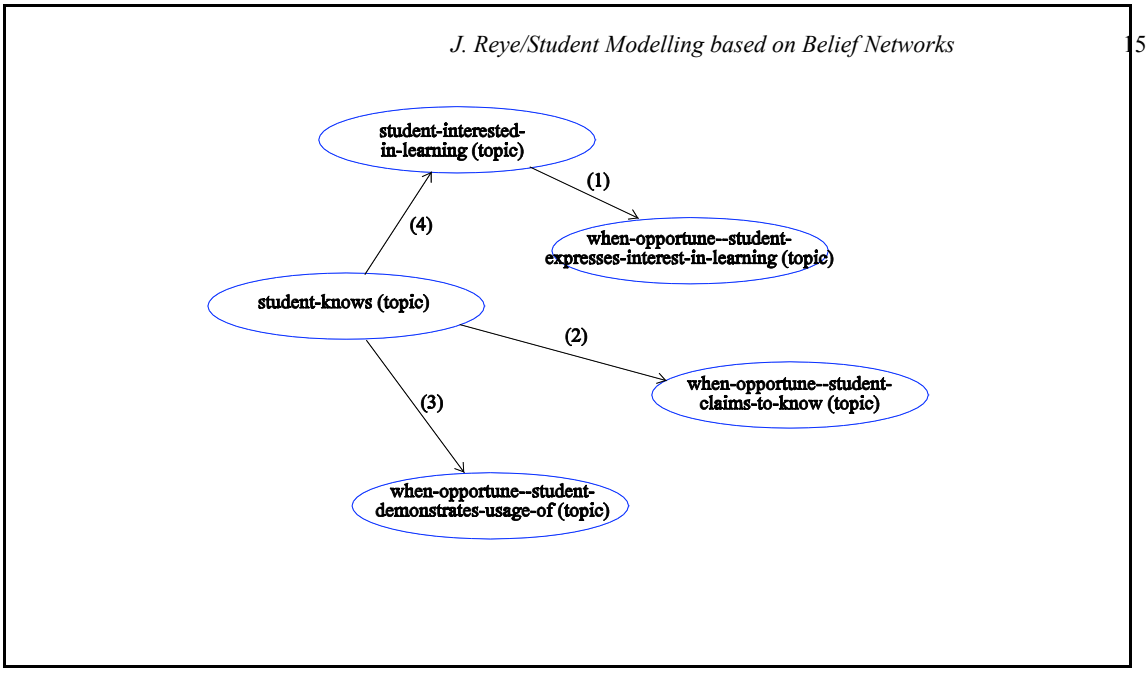

Fig. 5. Example of a topic cluster, containing local nodes

Because of this, the examples of these presented in this paper are meant to be illustrative, not comprehensive. For clarity, I use the nodes in Figure 5 to describe the general categories of such nodes, in the following.

(a) Nodes that can be directly observed (from interactions with the student)

when-opportune--student-expresses-interest-in-learning (*topic*)

> Whenever a suitable opportunity arises, the student expresses an interest in being tutored on the topic, either by making an explicit request, or in response to a question from the system.

when-opportune--student-demonstrates-usage-of (*topic*)

> Whenever a suitable opportunity arises, the student demonstrates her knowledge of the topic by solving problems or answering questions that require such knowledge.

when-opportune--student-claims-to-know (*topic*)

> Whenever a suitable opportunity arises, the student claims to know the topic, either explicitly (e.g. in response to a question from the system), or implicitly (e.g. by not objecting when the system tries to move the dialogue focus on to a new topic).

The repetitive use of the prefix "when-opportune" is somewhat clumsy, but it is helpful to try to make it clearer that the system's belief only applies to student responses in appropriate circumstances. If this prefix was omitted, the reader may be mislead in a number of ways.

For example, if the last predicate was just named "student-claims-to-know (*topic*)", then some readers might take this to be true, only if it happened in the *most recent* interaction, while others might take this to be true, if it happened in *any past* interaction, while still other readers might take it to be a prediction about the *next* interaction. Using the "when-opportune" prefix with "student-claims-to-know (*topic*)", (hopefully) makes it clear that the predicate represents the system's current beliefs about possible future claims that the student might make, if the opportunity ever arises. Thus, purely through inferencing (e.g. via the backbone), the system could have a high degree of belief in "when-opportune--student-claims-to-know (*topic*)", for a given topic, even though the particular student has never made such a direct claim, and may never need to do so, in future interactions.

I also emphasise that I use the phrase "Whenever a suitable opportunity arises" in the broadest sense. For example, if a student is totally uninterested in the system's most recent question (or exercise), then she may legitimately regard it as an opportunity to express interest in another topic, rather than having to wait for the system to directly ask about any such interest. For this reason, I rejected other possible prefixes, such as "when-asked", which are shorter but unfortunately imply a narrower scope.

(b) Nodes that cannot be directly observed

student-knows (*topic*)

Indicates whether the student is familiar with the topic (concept or skill).

student-interested-in-learning (*topic*)

Indicates whether the student is interested in being tutored on the topic.

Here, the prefix "when-opportune" is unnecessary, as there is never an opportunity to directly observe the truth values of these nodes.

(c) Relationships among these nodes

The relationships numbered 1 through 3, in Figure 5, should be thought of as *causal relationships*, in that an increase in belief for the node from which the arrow leaves (e.g. "student-knows") should lead to an increase in belief for the node to which the arrow goes (e.g. "when-opportune--student-demonstrates-usage-of"). As the reader may have already realised, these relationships, and the associated "when-opportune..." nodes, are intended simply to model the fact that the student is not usually an entirely reliable source of information, due to lucky guesses, slips, etc. This issue was described in the earlier section “Representing simple prerequisite relationships as probabilistic

relationships", for the isolated case of just two nodes. Here, the same idea is just being used in three separate instances.

While the specification of precise conditional probabilities is not a major concern of this paper, it is worth mentioning that, in general, one would expect such causal relationships to be fairly strong (otherwise, there is no point in gathering evidence from the student). For example, one would typically expect:

p(when-opportune--student-demonstrates-usage-of (*topic*)|student-knows(*topic*))

to be close in value to 1.0, i.e. just a small allowance for the occasional slip; and:

p(when-opportune--student-demonstrates-usage-of (*topic*)|¬student-knows(*topic*))

to have a fairly small value, i.e. allowing for the occasional lucky guess. (The harder it is to guess, then the lower the value.)

By comparison, the relationship numbered 4, in Figure 5, should be thought of as an *inverse causal relationship,* because an increase in belief for the node from which the arrow leaves (i.e. "student-knows (*topic*)") should lead to a decrease in belief for the node to which the arrow goes (i.e. "student-interested-in-learning (*topic*)"). This is because someone who already knows a topic is unlikely to be interested in learning it again (assuming, for the moment, that the student is genuine, and does not have an ulterior motive for trying to get the system to present material that the student has learned previously, e.g. if forced to use the system against their own wishes, because of some reward, such as credit towards passing a subject).

In using conditional probabilities to represent this relationship, one would typically expect:

p (student-interested-in-learning (*topic*) | student-knows (*topic*))

to have a value of zero (or very close to it). But, not much can be generally stated about the value of the other required conditional probability:

p (student-interested-in-learning (*topic*) | ¬student-knows (*topic*))

If this was a topic of broad appeal to students, then one might assign a value close to 1.0, i.e. if they don't know it, then they'll be interested in learning it. But, if the topic has very limited appeal, then a value close to zero would be more appropriate. Obviously, values between these two extremes also make sense in the right circumstances.

Global nodes

Global nodes represent the system's belief in overall student characteristics, i.e. beliefs not focussed on a specific topic. For example:

student-overall-aptitude ()

Measures the system's overall impression of the student's aptitude.

student-reliability-of-claims ()

Measures the system's overall impression of how reliable the student is in making claims about what they do and don't know, etc.

student-is-bored ()

Indicates the system's overall impression that the student is generally bored with the subject material.

In a relatively-simple student model, each such node may have a range of only two values. For example, "student-overall-aptitude ()" may just allow the values "has-aptitude" and "does-not-have-aptitude". In a more complex model, a greater range of values may be desired, e.g.: very-good-aptitude; good-aptitude; average-aptitude; poor-aptitude and very-poor-aptitude. Likewise "student-is-bored ()" could be simple "true" or "false", or it could be broader, e.g.: very-keen; somewhat-keen; somewhat-bored and very-bored.

One of the advantages of using a probabilistic model is that it allows the system to model its uncertainty about these characteristics. For example, for a given student at some point in time, the system might contain the following probabilities for the "student-overall-aptitude ()" node:

p (very-good-aptitude) = 0.1
p (good-aptitude) = 0.4
p (average-aptitude) = 0.3
p (poor-aptitude) = 0.15
p (very-poor-aptitude) = 0.05

(which must total to 1.0, of course). Figure 6 provides an example of the kinds of relationships one might want to include in a student model. Note that the global nodes are not really part of the topic cluster, but are just shown in this same figure for clarity.

As before, these relationships are modelled using conditional probabilities. But these are now more complex than in the earlier examples, when global nodes were not yet introduced, e.g.:

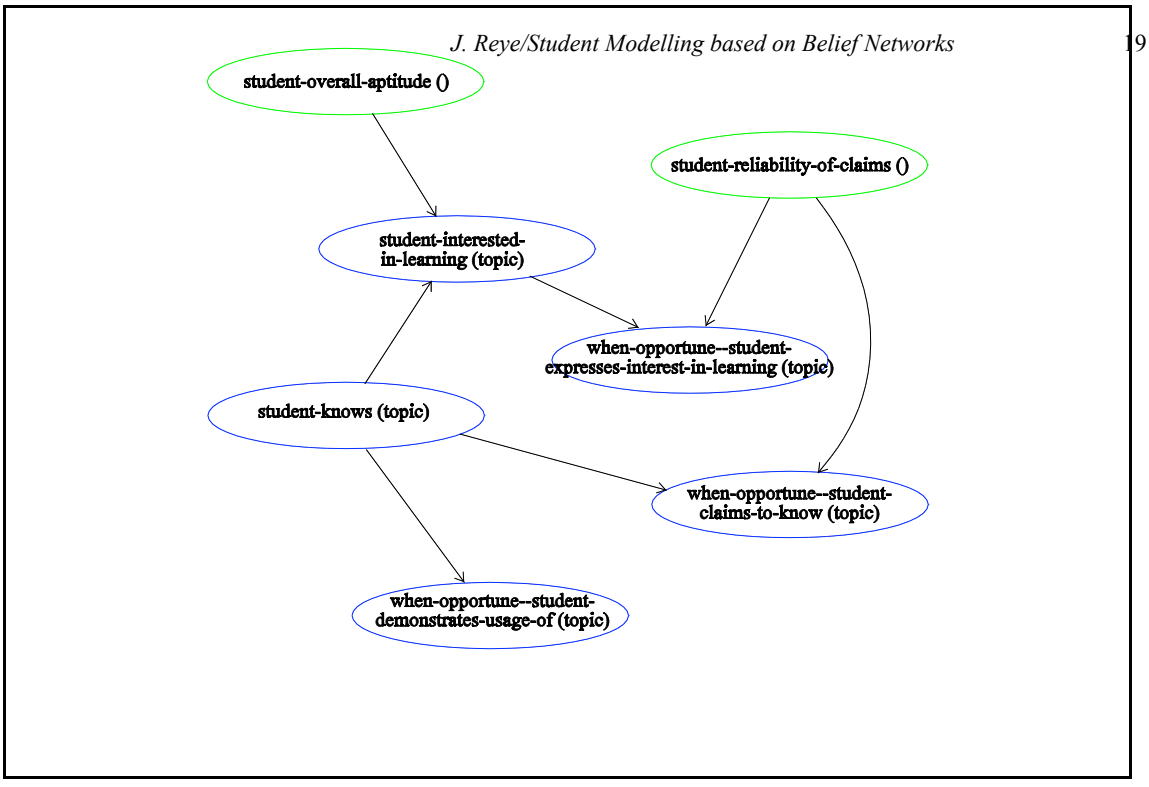

Fig. 6. Relationships between two global nodes and the local nodes in a topic cluster

p (when-opportune--student-interested-in-learning (*topic*)
| student-knows(*topic*), student-overall-aptitude()="good-aptitude")=0
p (when-opportune--student-interested-in-learning (*topic*)
| student-knows(*topic*), student-overall-aptitude()="poor-aptitude")=0
p (when-opportune--student-interested-in-learning (*topic*)
| ¬student-knows(*topic*), student-overall-aptitude()="good-aptitude")=0.7
p (when-opportune--student-interested-in-learning (*topic*)
| ¬student-knows(*topic*), student-overall-aptitude()="poor-aptitude")=0.05

and:

p (when-opportune--student-claims-to-know (*topic*)
| student-knows(*topic*),student-reliability-of-claims()="good-reliability")=0.99
p (when-opportune--student-claims-to-know (*topic*)
| student-knows(*topic*),student-reliability-of-claims()="poor-reliability")=0.7
p (when-opportune--student-claims-to-know (*topic*)
| ¬student-knows(*topic*),student-reliability-of-claims()="good-reliability")=0.1
p (when-opportune--student-claims-to-know (*topic*)
| ¬student-knows(*topic*),student-reliability-of-claims()="poor-reliability")=0.6

In terms of the topic clusters, the global nodes can really be regarded as global parameters that fine-tune the conditional probabilities, based on what is known about the student in overall terms. So, even though they complicate the model, these global nodes are *important for supporting better inferencing* about the individual student's knowledge of a particular topic, taking into account both the local characteristics of that topic (e.g. some topics are harder to learn) and the global characteristics of that student. (Global nodes can also be used for other purposes, such as planning to increase the student's motivation.)

Having described the way that the student model is structured in terms of a belief network, I next address the issue of how the local nodes in the student model are updated, in response to tutorial interactions.

UPDATING THE STUDENT MODEL: DYNAMIC BELIEF NETWORKS

Introduction

When a belief network is used to represent a student model (e.g. Villano, 1992), we must have a theoretically-sound way to update this model. Such updates are based on information from two sources: (i) the student, via their inputs to the system (e.g. requests for help, answers to questions, and attempts at exercises); and (ii) the system, via its outputs (e.g. descriptions and explanations given). In this paper, I give a general approach as to how such updates should be made, and show how this work relates to previous research in this area.

In ordinary belief networks, it is assumed that the properties of the external world, modelled by the network, do not change as we go about gathering evidence related to those properties. That is, even though the system gathers information from the external world that causes it to modify its measures of belief about items in that world, those items remain either true or false. This is useful, for example, in medical diagnosis, where the cause of a disease is assumed not to change during a (single) medical examination.

But, such an approach is clearly inadequate for student modelling in a tutoring system, where we must be able to reason about:

(a) the dynamic evolution of the student's knowledge, over a period of time, as we gain new information about the student; and

(b) the likely effects of future tutorial actions (relative to what is currently known about the student), so that the action with maximum likely benefit to the student can be chosen.

Dynamic belief networks (Dean and Kanazawa, 1989) allow for reasoning about change over time. This is achieved by having a sequence of nodes that represent the state of the external item over a period of time, rather than having just a single temporally-invariant node. For real-world continuous processes, the sequence of nodes may represent the external state as it changes over a sequence of time-slices. For tutoring, it is often more useful to represent changes in the student

model over a sequence of interactions, rather than time-slices (as illustrated by the example in the following section).

Two-phase updating of the student model

In general, an interaction with a student must cause the system to revise its beliefs about the student's state of knowledge. On first consideration, it might appear that this updating of beliefs should be modelled as a single process, representing the transition from prior beliefs to posterior beliefs.

However, in the general case, an interaction with a student may provide clues about two distinct (but related) pieces of information: (i) how likely it is that the student knew a topic *before* the interaction; and (ii) how likely it is that the student knows a topic *after* the interaction, i.e. what change (if any) is caused by the interaction.

Consequently, I advocate a *two-phase approach to updating the student model,* at each interaction:

(a) phase 1: the incorporation of evidence (if any) from the interaction, about the student's *state of knowledge as it was prior to the interaction*; and

(b) phase 2: the expected changes (if any) in the student's *state of knowledge as a result of the interaction*.

Many ITS architectures have clearly distinguishable Analysis (input-processing) and Response (output-generating) components. The two-phase approach maps naturally onto these architectures: phase 1 covers updates made by the Analysis component; and phase 2 covers updates made when executing tutorial actions chosen by the Response component.

This two-phase approach is applicable to any architecture that uses probability theory for student/user modelling. This is true even if probability theory is just used to model uncertainty about isolated nodes (rather than structuring these into a belief network).

As an example of this broad applicability, the two-phase approach applies to a relatively simple system that just goes through a list of topics, giving the student multiple-choice or fill-in-the-blank type questions and provides multiple-level hints when the student requests help or gives an incorrect answer. The restricted instructional capabilities of such a system may mean that any prerequisite constraints are implicit in the ordering of the list of topics, rather than being explicitly represented in a belief network. But, such a system still needs probability theory to correctly model the facts that: (i) students sometimes make slips and lucky guesses, and (ii) there are different likelihoods of learning, from different levels of hints[4]. Hence, such a system needs phase 1 to

[4]For example, a brief hint may be a concise reminder to some students who have previously studied the topic, but useless to others who have never seen it before. For students in this latter group, a detailed hint may increase their chance of learning the topic, but has the disadvantages of verbosity and reduced challenge to the student.

updates its beliefs after receiving a student's answer, and phase 2 to update its beliefs after giving the student a hint at a particular level.

In any system, phase 1 is clearly important for gathering information at the *first* interaction on a given topic, i.e. topics for which there has not been any previous interaction with the particular student. But phase 1 is especially important for gathering information at *each* interaction, because the model must allow for the possibility that the student's knowledge will change independently of interactions with the system, i.e. the student may forget, may study independently, etc. It is necessary that the system be able to handle the fact that substantial periods of time (hours, days, weeks) may elapse from one interaction to the next, depending on how the student wishes to make use of the system.

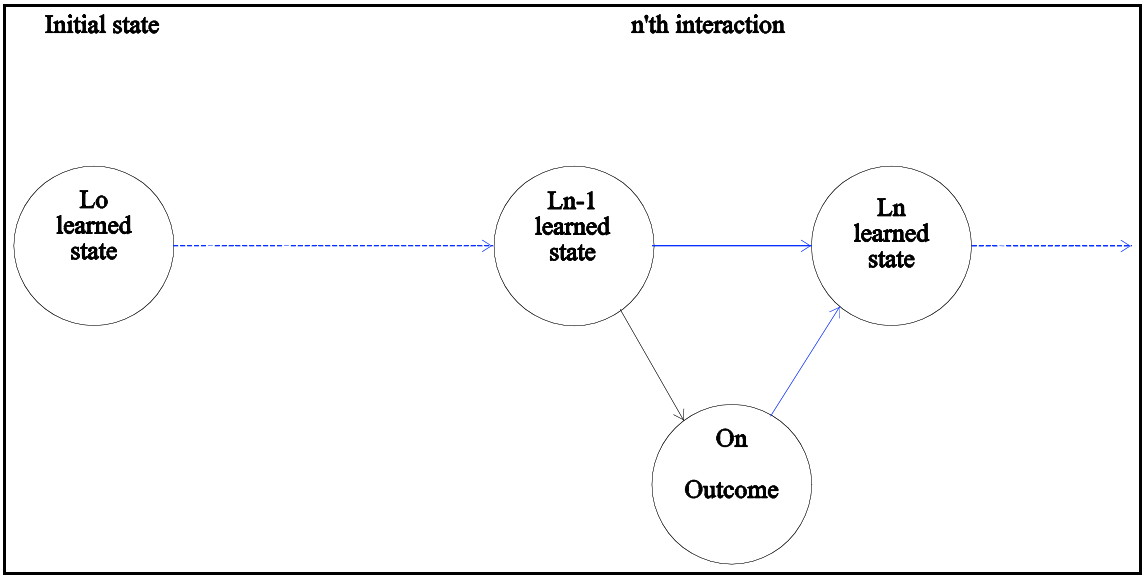

Fig. 7. Two-phase updating

Phase 1: incorporation of evidence about the student's knowledge

With regard to Figure 7, let:

(a) O_n be an element in a *set of possible outcomes* of a given tutorial interaction involving a given domain topic, i.e. the set of allowed student responses for that interaction, e.g. (a) correct or incorrect; (b) no-help, level-1-help, level-2-help, etc;

(b) p (L_{n-1}) represent the system's belief that the student *already knows the given domain topic*, prior to the n'th interaction (where n = 1, 2, ...);

(c) $p(O_n \mid L_{n-1})$ represent the system's belief that *outcome O_n will occur when the student already knows* the domain topic;

(d) $p(O_n \mid \neg L_{n-1})$ represent the system's belief that *outcome O_n will occur when the student does not know* the domain topic;

where (c) and (d) are the two conditional probabilities needed to fully define the single link between the "L_{n-1} learned state" node and the "O_n Outcome" node, shown in Figure 7.

Given values for each of these, the system must be able to revise its belief in $p(L_{n-1})$ when outcome O_n occurs. This is done by using the well-known Bayes's rule:

$$p(L_{n-1} \mid O_n) = \frac{p(O_n \mid L_{n-1})p(L_{n-1})}{p(O_n \mid L_{n-1})p(L_{n-1}) + p(O_n \mid \neg L_{n-1})p(\neg L_{n-1})} \quad (1)$$

Let $\lambda(O_n)$ be the likelihood ratio:

$$\lambda(O_n) = \frac{p(O_n \mid L_{n-1})}{p(O_n \mid \neg L_{n-1})} \quad (2)$$

Then, Equation (1) can be simplified to:

$$p(L_{n-1} \mid O_n) = \frac{\lambda(O_n)p(L_{n-1})}{1 + [\lambda(O_n) - 1]p(L_{n-1})} \quad (3)$$

In passing, I point out that this equation clearly shows why phase 1 is insufficient for updating the student model on its own, i.e. why phase 2 cannot be omitted. When the prior belief $p(L_{n-1})$ is 0, then the posterior belief $p(L_{n-1} \mid O_n)$ must also be 0. Without phase 2, $p(L_n)$ would be the same as $p(L_{n-1} \mid O_n)$ and so would be 0 also, in this case. Similarly, when $p(L_{n-1})$ is 1, then $p(L_n)$ would also be 1. That is, without phase 2, values 0 and 1 represent absorbing states.

Consequently, if the system ever became absolutely convinced that a student did (or didn't) know a topic, then this equation would never allow the system to change that belief. Further, if $p(L_{n-1})$ is very close to 0 or 1, then $p(L_n)$ will also be close to that same value (unless $\lambda(O_n)$ is very large). This makes it hard for the system to move away from values close to 0 or 1, when revising its beliefs using phase 1 alone.

Phase 2: expected changes in the student's knowledge due to tutoring

In phase 2, we model the expected changes in the student's knowledge as a result of the interaction. Doing this requires a formula for $p(L_n \mid O_n)$, so that we know what probability to assign to $p(L_n)$ for each possible outcome, O_n, in the set of possible outcomes.

To fully define the double link from the "L_{n-1} learned state" node and the "O_n Outcome" node to the "L_n learned state" node shown in Figure 7, requires two conditional probabilities for each possible outcome:

(a) $p(L_n \mid L_{n-1}, O_n)$

This function represents the probability that the student will remain in the learned state as a result of the outcome, i.e. it is the *rate of remembering* (or "not forgetting"). An ITS's interaction will not cause the student to forget something they already know, so this probability will have the value 1 when O_n is the best outcome, e.g. giving the correct answer without any help. Progressively poorer outcomes should therefore have progressively lower values.[5]

(b) $p(L_n \mid \neg L_{n-1}, O_n)$

This function represents the probability that the student will make the *transition from the unlearned state to the learned state* as the result of the outcome, i.e. it is the rate of learning.

From this, the revised belief after the interaction is simply given by:

$$p(L_n \mid O_n) = p(L_n \mid L_{n-1}, O_n)\, p(L_{n-1} \mid O_n) + p(L_n \mid \neg L_{n-1}, O_n)\, p(\neg L_{n-1} \mid O_n) \quad (4)$$

[5] It might be thought that we could better estimate the rate of remembering if we took into account the length of time between the n-1'th and the n'th interaction, especially if such intervals may sometimes be large, e.g. between tutoring sessions on different days or weeks. In terms of the basic theory in this paper, this would simply require us to condition on the interval as well, i.e. use conditional probabilities of the form: $p(L_n \mid L_{n-1}, O_n, \text{Interval}_n)$. However, it is extremely difficult to create a psychologically-accurate model of human memory retention, especially if we don't know whether the student was studying the domain material independently during the same interval, i.e. while not using the ITS. Because of these difficulties, there is no guarantee that a more complex model would be more effective than the simple approach used here.

For notational simplicity, let:

(a) $\alpha(O_n) = p(L_n \mid L_{n-1}, O_n)$; and

(b) $\beta(O_n) = p(L_n \mid \neg L_{n-1}, O_n)$.

Then Equation 4 can be simplified to:

$$p(L_n \mid O_n) = \beta(O_n) + [\alpha(O_n) - \beta(O_n)]\ p(L_{n-1} \mid O_n) \tag{5}$$

The equations for each phase, (3) and (5), can be used separately, but it is also useful to combine together, so that one can conveniently describe the effects of an entire interaction. This combination is given in the following section.

Combining the two phases

Combining Equation 5 with Equation 3 gives:

$$p(L_n \mid O_n) = \beta(O_n) + \frac{[\alpha(O_n) - \beta(O_n)]\ \alpha(O_n)\ p(L_{n-1})}{1 + [\alpha(O_n) - 1]\ p(L_{n-1})} \tag{6}$$

which can be rewritten as:

$$p(L_n \mid O_n) = \frac{\beta(O_n) + [\alpha(O_n)\ \alpha(O_n) - \beta(O_n)]\ p(L_{n-1})}{1 + [\alpha(O_n) - 1]\ p(L_{n-1})} \tag{7}$$

When $p(L_{n-1}) = 1$, Equation 7 gives:

$$p(L_n \mid O_n) = \alpha(O_n) \tag{8}$$

That is, $\alpha(O_n)$ is the same as $p(L_n \mid O_n)$ if the student previously knows the topic, illustrating the earlier description of $\alpha(O_n)$ as the "rate of remembering".

When $p(L_{n-1}) = 0$, Equation 7 gives:

$$p(L_n \mid O_n) = \beta(O_n) \tag{9}$$

That is, $\beta(O_n)$ is the same as $p(L_n \mid O_n)$ if the student previously does not know the topic, illustrating the earlier description of $\beta(O_n)$ as the "rate of learning".

Figure 8 illustrates Equation 7 for some particular values of γ, β and α. For each tuple of such values, there is a direct visual interpretation of these three parameters: the height of each end-point directly portrays the values of parameters β and α (in accord with Equations 8 and 9; and the curvature depends directly on the value of γ, i.e. concave when $\gamma > 1$, convex when $\gamma < 1$, and straight when $\gamma = 1$.

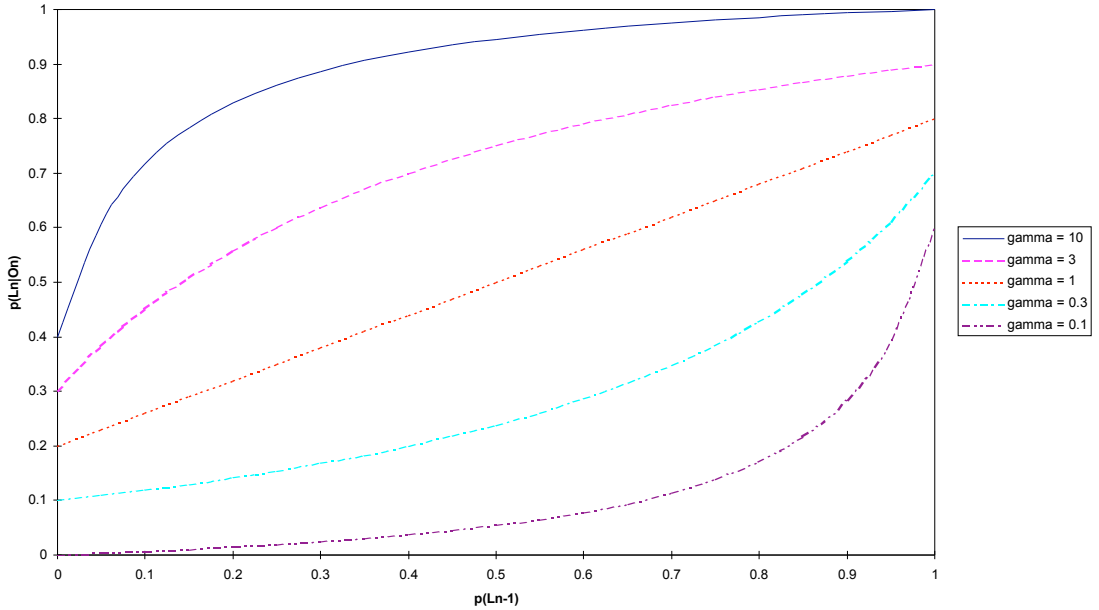

Fig. 8. Example curves for Equation 7

A dynamic belief network for probabilistic modelling in the ACT Programming Languages Tutor

The ACT Programming Languages Tutor (Corbett and Anderson, 1992) uses a two-state psychological learning model, which is updated each time that the student has an opportunity to show their knowledge of a production rule in the (ideal) student model. In an appendix to their paper, the authors briefly state equations for calculating the probability that a production rule is in the learned state following a correct (C_n) or erroneous (E_n) student response, at the nth opportunity.

In this section, I illustrate the applicability of dynamic belief networks by showing how Corbett and Anderson's equations can be derived as a special case of the equations given above.

In their paper, Corbett and Anderson did not describe how they derived these equations. Even though they did not make any use of the concept of dynamic belief networks, their learning model is clearly based on the mathematics of probability theory. So, it is not too surprising that there should be a direct relationship between their work and mine. In the following, I add constraints to my model until I obtain their equations, showing that the approach in this paper generalises their's. The authors make the following simplifying assumptions:

(a) the set of outcomes has only two values: C_n (i.e. correct) and E_n (i.e. error).

(b) $p(L_n \mid L_{n-1}, O_n) = \square(O_n) = 1$, i.e. no forgetting.

(c) $p(L_n \mid \neg L_{n-1}, O_n) = \square(O_n)$ is a constant, i.e. the probability that the student will make the transition from the unlearned to the learned state is *independent of the outcome*. The authors use the symbol p(T) for this constant.

(d) there are no conditional probabilities linking different rules, i.e. no prerequisite constraints.

Assumption (c) means that there is no direct dependency between the L_n node and the O_n node shown previously in Figure 7. By dropping this arrow, it should be clear that this is the simplest possible (structurally) dynamic belief network for student modelling (and simplicity may be a virtue rather than a vice).

Under the above assumptions, Equation 5 becomes:

$$p(L_n \mid O_n) = p(T) + [1 - p(T)]\ p(L_{n-1} \mid O_n)$$

i.e.

$$p(L_n \mid O_n) = p(L_{n-1} \mid O_n) + p(T)\,(1 - p(L_{n-1} \mid O_n))$$

When O_n is replaced by each of the two possible outcomes, C_n and E_n, we obtain:

$$p(L_n \mid C_n) = p(L_{n-1} \mid C_n) + p(T)(1 - p(L_{n-1} \mid C_n))$$

$$p(L_n \mid E_n) = p(L_{n-1} \mid E_n) + p(T)(1 - p(L_{n-1} \mid E_n))$$

which are the two equations that Corbett and Anderson number as [1] and [2], in their paper.

Under these same assumptions, Equation 2 becomes:

$$\square(C_n) = \frac{p(C_n \mid L_{n-1})}{p(C_n \mid \square L_{n-1})}$$

when O_n has the value C_n. Substituting this into Equation 3, a version of Bayes's theorem, gives:

$$p(L_{n-1} \mid C_n) = \frac{p(C_n \mid L_{n-1})p(L_{n-1})}{p(C_n \mid L_{n-1})p(L_{n-1}) + p(\neg L_{n-1})p(C_n \mid \neg L_{n-1})}$$

which is the same as the equation marked [3] in Corbett and Anderson's paper, except for: (i) some rearrangement of terms; and (ii) they use the symbol "U_{n-1}" (for "unlearned") where I use "$\neg L_{n-1}$". For brevity, I omit the analogous derivation of their equation [4] for p ($L_{n-1}|E_n$).

As a result of their assumptions, Corbett and Andersen's model has only four parameters associated with each rule. I list these parameters below, and, for clarity of reference, use the same notation utilised by the authors:

p(L_0) the probability that a rule is in the *learned* state prior to the first opportunity to apply the rule (e.g. from reading text);

p(C|U) the probability that a student will guess correctly if the applicable rule is in the *unlearned* state (same as my p (O_n=C_n|$\neg L_{n-1}$));

p(E|L) the probability that a student will slip and make an error when the applicable rule is in the *learned* state (same as my p (O_n=E_n|L_{n-1}));

p(T) The probability that a rule will make the transition from the *unlearned* state to the *learned* state following an opportunity to apply the rule (same as my τ (O_n)).

In the most general case, the values of these parameters may be set empirically and may vary from rule to rule. Corbett and Anderson (1992) describe a study in which these parameters were held constant across 21 rules, with p(L_0) = 0.5, p(C|U) = 0.2, p(E|L) = 0.2 and p(T) = 0.4. In my notation, these values are equivalent to $\tau(C_n) = 4$, $\tau(E_n) = 0.25$ and $\tau(C_n) = \tau(E_n) = 0.4$.

Another example of a dynamic belief network: SMART

Shute's (1995) SMART student modelling approach uses a number of functions for updating the student model, as illustrated in Figure 9. The points in this figure were developed mainly by hand, based on the opinions of domain experts. These points were used to compute best-fitting curves – see Shute (1995) for details. These curves were then used within her Stat Lady system. This system's estimates of the final knowledge obtained by a group of students correlated well with an independent post-test measure of their knowledge. The success of her approach encourages us to study it further.

As is clear from Figure 9, Shute's model is a probabilistic one, raising the interesting question as to how it relates to the approach described in this paper. Like Corbett and Anderson, Shute does not make any use of the concept of dynamic belief networks. However, in this section, I

show that such networks are a good way to provide a theoretical foundation for her work, by showing how Shute's graphs can be represented using my equations.

When solving each problem posed by Shute's SMART system, the student is allowed to choose from four levels of help (or "hints"), where level-0 covers the case where the student required no help at all. Unlike Corbett and Andersen, Shute does not make the assumption that the probability of learning is independent of the outcome. This is obvious from the fact that there are four separate curves in Figure 9.

Figure 10 shows the curves obtained when plotting Equation 7 with the following values for □, □ and □:

Outcome	□	□	□
Level-0 help	2.35	0.33	1
Level-1 help	2.12	0.33	0.83
Level-2 help	0.66	0	0.83
Level-3 help	0.52	0	0.5

By inspecting this figure, it is clear that Equation 7 provides a good theoretical basis for Shute's graphs. Her graphs were not directly derived from empirical data. So, there is no point in doing a statistical analysis to obtain more precise values for the three parameters. However, the development of methods for estimating these parameters from empirical data is an interesting problem for future research.

This concludes the discussion of the applicability of this student modelling approach in which we have seen that the student modelling approaches of Corbett and Anderson (1992) and Shute (1995) are special cases of this approach, which is far from obvious at first glance.

COMPUTATIONAL EFFICIENCY IN A BELIEF-NET-BASED STUDENT MODEL

For an arbitrarily connected belief network, the computational complexity of probabilistic inferencing is NP-hard (Cooper, 1989). Clearly, the computational complexity of probabilistic inferencing, within the student model, is a significant issue because:

(a) a complex domain may have thousands of fine-grained topics, e.g. full SQL-99;

(b) associated with each topic is a topic cluster that multiplies the total number of nodes in the network, by a constant factor, e.g. typically by 5 to 10 times;

(c) tutoring takes place in real-time;

(d) more than one occurrence of probabilistic inferencing may be required to support a single interaction, i.e. separate inferencing in phase 1 and in phase 2.

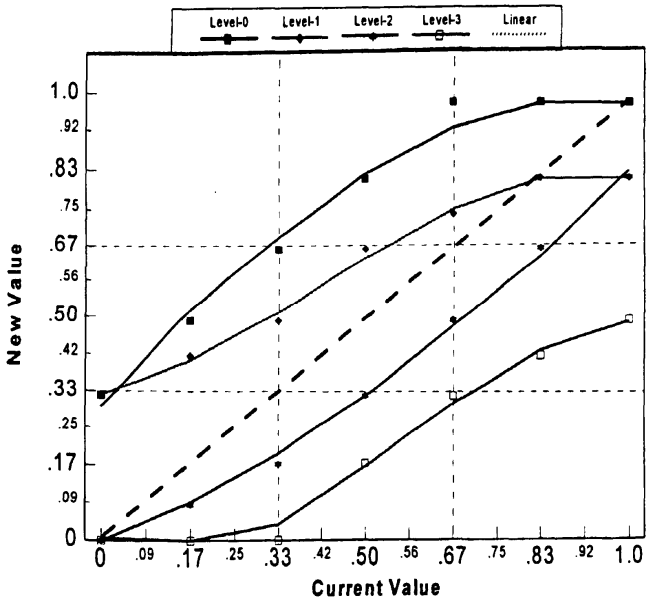

Fig. 9. SMART's Updating Functions

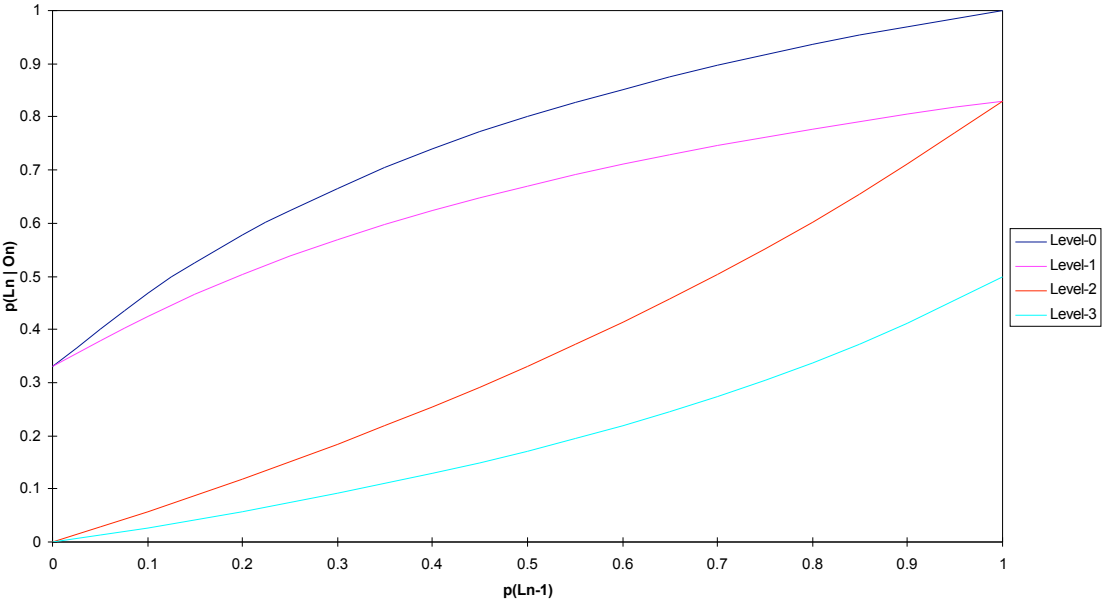

Fig. 10. Curves from Equation 7 for chosen values of □ □ and □

While Cooper's computational complexity result applies to arbitrarily connected networks, it is often possible to take advantage of the particular structure of a network to do probabilistic inferencing in a relatively efficient manner. In particular, efficient algorithms exist for networks that are only singly connected (Pearl, 1988). Unfortunately, student models are not usually singly connected because:

(a) the prerequisite relationship may provide more than one path between two topics, e.g. if A is a prerequisite for each of B and C, and both B and C are prerequisites for D;

(b) as well as being connected by the prerequisite relationship, two topic clusters will be connected by each global node in the student model.

As already mentioned, even when networks are not singly connected, it is still often possible to use structural features of a given network to reason efficiently. Pearl (1988) describes the concept of *d-separation*, which can be used to determine those parts of the network that are affected by each new set of belief revisions, e.g. as new evidence becomes available. In other words, if a given node is d-separated from those nodes that have newly revised beliefs, then the given node is unchanged. So, d-separation can be used to limit the propagation of the effects of each belief revision, rather than propagating through the entire network.

Associated with the idea of d-separation is the concept of a barren node. A *barren node* is one for which no evidence is available (as yet) and the same is true for all its descendant nodes, if it has any. (A *descendant* node is one which can be reached by following the arrows in the belief network.) As Baker and Boult (1991) discuss, the properties of d-separation mean that barren nodes can be removed from the network without affecting the probability values calculated at the remaining nodes. In other words, information about belief revisions cannot flow through a barren node, so their removal has no impact on belief updating in the rest of the network. The removal of such nodes aids computational efficiency by reducing the size of the network.

As far as student modelling is concerned, its temporally-evolving nature clearly means that barren nodes should not be removed literally, but should be marked as being barren (while applicable) so that they can be ignored by the belief propagation routines.

The importance of barren nodes in student modelling lies in the fact that, in a large domain, there will often be large numbers of barren nodes. For each topic in the domain, if the system has not yet gathered any information about whether the student knows that topic, then all the local nodes in the corresponding topic cluster will be barren, including the "student-knows (*topic*)" node, e.g. see Figure 5. In other words, the entire topic cluster can be omitted during belief propagation. Taken to the extreme, it can be seen that the entire network is barren when starting with a new student. (Although the global nodes are barren in this extreme case, they will not remain so for long, so it is usual to view them as not being barren.)

As knowledge of the student accumulates, the number of barren nodes will diminish, gradually increasing the computational overhead. However, unless the student model is a very simple one (i.e. contains very few nodes), the system will be unable to gather sufficient information to eliminate them all. So, recognising them as barren will still provide a computational benefit.

To further illustrate this point, note that some nodes in a given topic cluster can remain barren, even when information is gathered with respect to the student's knowledge of the corresponding topic. For example, in Figure 5, if the only evidence we have is that the student claims to know the given topic, then the "when-opportune--student-claims-to-know (*topic*)" node and the "student-knows (*topic*)" node are no longer barren, but the other three nodes remain barren. An important consequence of this is that we have the freedom to make the structure of the topic cluster substantially more complex, without fearing that the extra complexity will cause a proportional increase in execution time.

For clarity, the preceding description of the usefulness of barren nodes has deliberately ignored one important detail. Even though a node is barren, we may need to know its current probability value so that this can be used for diagnosis or instructional planning (rather than for belief propagation). This is most likely to apply to the "student-knows (*topic*)" nodes (and less likely to apply to the other local nodes).

Fortunately, there is an easy (rapid) way to determine the current probability value for a barren node. It can be simply calculated from the probabilities of its parent nodes, together with the condition probabilities that link the parents to the node of interest. (D-separation guarantees that we do not have to consider other parts of the network in this calculation.)

If any of the parents are barren themselves, then we simply use a "pull-style" recursive procedure to calculate their values first, before calculating the value for the node of interest. (This could be inefficient, if care is not taken. For example, if an unbroken series of enquiries is made for values of barren "student-knows (*topic*)" nodes, a naive implementation could calculate the same ancestor node's value many times. A more sophisticated implementation would either remember such values (as long as they remained valid), or use a "push-style" procedure to visit each such node only once, e.g. using the belief net backbone to walk through all the "student-knows (*topic*)" nodes before querying begins.)

Nodes without parents (i.e. some global nodes and those "student-knows (*topic*)" nodes without prerequisites) must already have associated probability values (and there is no computational advantage to marking such nodes as being barren, in any case).

In summary, the concepts of d-separation and barren nodes allow significant reductions to be made in the computational complexity of belief propagation in belief-net-based student models.

CONCLUSION

This paper gives an approach to student modelling by using probability theory in the form of a belief network. In addition to showing the basic approach of using probability theory to formally model the uncertainty of isolated beliefs about the student, I've shown how to use a belief network to structure knowledge about related beliefs, providing a foundation for two important ITS tasks: gathering information about the student's current state of knowledge; and modelling students as (somewhat) unreliable sources of information.

To further develop this approach of structuring knowledge, this paper proposes that the appropriate belief network structure for an ITS is based on the ideas of a belief net backbone, local nodes in topic clusters, and global nodes, and shown how these connect together. Such

structures have advantages both for ITS designers and for efficient local computation in an implemented system.

Finally, we have examined a general theory of how the student model should be updated, based on the concept of a dynamic belief network. This theory is general in that it can be used in any ITS that uses probabilistic modelling. There are, however, advantages in using this theory in conjunction with other ideas presented here, e.g. topic clusters.

ACKNOWLEDGEMENTS

I wish to thank John Self and the anonymous reviewers for their constructive comments, that have resulted in a better quality paper.

REFERENCES

Baker, M., & Boult, T. (1991). Pruning Bayesian Networks for Efficient Computation. In P. Bonissone et al., (Eds.) *Uncertainty in Artificial Intelligence 6* (pp. 225-232). Elsevier Science Publishers.

Collins, A., & Stevens, A. (1982). Goals and strategies for inquiry teachers. In R. Glaser (Ed.) *Advances in Instructional Psychology (vol. II)*, 65-119.

Cooper, G. (1989). Probabilistic inference using belief networks is NP-hard. *Artificial Intelligence*, 393-405.

Corbett, A., & Anderson, J. (1992). Student modeling and mastery learning in a computer-based programming tutor. In C. Frasson, C. Gauthier, & G.I. McCalla (Eds.) *Intelligent Tutoring Systems, Proceeding of the Second International Conference, ITS'92, Montreal, Canada* (pp. 413-420). Berlin: Springer-Verlag.

De Kleer, J., & Williams, B. (1987). Diagnosing multiple faults. *Artificial Intelligence,* 32, 97-130.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, 5, 142-150.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann.

Pearl, J. (2000). *Causality: models, reasoning and inference*. Cambridge University Press.

Reye, J. (1996. Belief net backbone for student modelling. In C. Frasson, C. Gauthier, & A. Lesgold (Eds.) *Intelligent Tutoring Systems, Proceeding of the Third International Conference, ITS'96, Montreal, Canada* (pp. 596-604). Berlin: Springer-Verlag.

Reye, J. (1998). Two-phase updating of student models based on dynamic belief networks. In B. Goettl, H. Halff, C. Redfield, & V. Shute (Eds.) *Intelligent Tutoring Systems Proceeding of the Fourth International Conference, ITS'98, San Antonio, USA*. Berlin: Springer-Verlag.

Russell, S., & Norvig, P. (1995) *Artificial intelligence: a modern approach*. Prentice-Hall.

Self, J. (1993). Model-based Cognitive Diagnosis. *User Modeling and User-Adapted Interaction,* 3, 89-106.

Shute, V. (1995) SMART evaluation: cognitive diagnosis, mastery learning & remediation. In J. Greer (Ed.) *Artificial Intelligence in Education* (pp. 123-130). Charlottesville, VA: Association for the Advancement of Computing in Education.

Villano, M. (1992) Probabilistic student models: Bayesian belief networks and knowledge space theory. In C. Frasson, C. Gauthier, & G. McCalla (Eds.) *Intelligent Tutoring Systems, Proceeding of the Second International Conference, ITS'92, Montreal, Canada* (pp. 491-498). Berlin: Springer-Verlag.