Two-Phase Updating of Student Models Based on Dynamic Belief Networks

Jim Reye

Queensland University of Technology
Brisbane, Australia
j.reye@qut.edu.au

Abstract. When a belief network is used to represent a student model, we must have a theoretically-sound way to update this model. 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. I present a general approach as to how student model updates should be made, based on the concept of a dynamic belief network, and then show this work relates to previous research in this area.

1 Introduction

When a belief network is used to represent a student model (e.g. [1], [2]), 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 present a general approach as to how such updates should be made, and show 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 [3] 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).

2 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.

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

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.

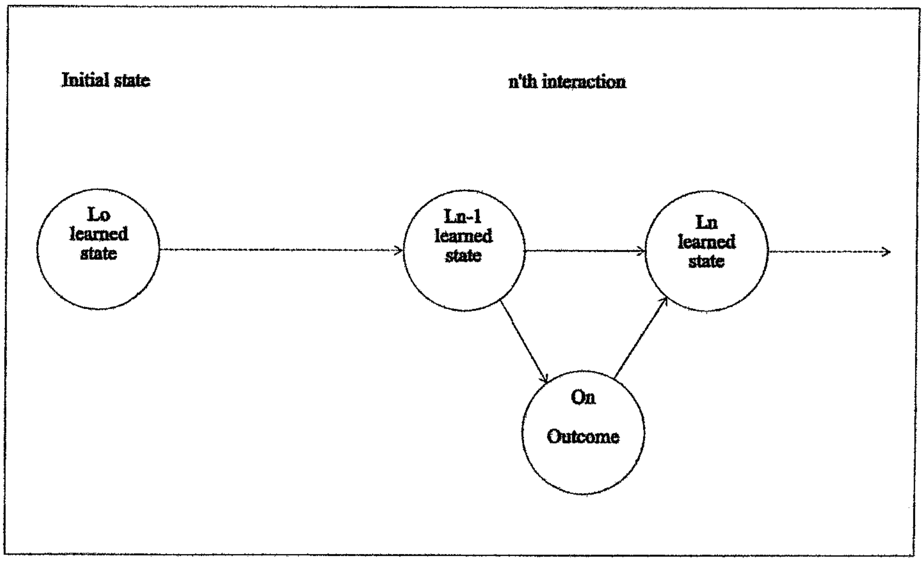

Figure 1. Two-phase updating

Figure 1 is an overall illustration of a dynamic belief network representing this two-phase approach. As this figure shows, following the n'th interaction, the probability that the student is *now* in the *learned state*, L_n (i.e. student-knows (*topic*)), depends both on whether the student was already in the learned state, L_{n-1} , and on the outcome, O_n, of the interaction. (The figure also shows that there must be some initial assumption about how likely it is that the student is in the learned state, prior to any interactions with the system, L_0, so that the updating process has somewhere to begin.)

In the sections that follow, I derive the mathematical formulae that follow from the two-phase approach. I then illustrate my claim for its generality, by showing that the student modelling approaches of Corbett and Anderson [4] and Shute [5] are actually special cases of this approach, even though these two papers appear totally unrelated to each other!

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

With regard to Figure 1, 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 1.

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 $K(O_n)$ be the likelihood ratio:

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

Then, equation (1) can be simplified to:

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

4 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 1, 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"). As we do not expect an ITS's interactions to cause the student to forget something they already know, this probability will have the value 1 in most implementations. However, for completeness, I leave it as a function here.

(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:

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

For notational simplicity, let:

(a) $\Psi(O_n) = p(L_n \mid L_{n-1}, O_n)$; and
(b) $\Sigma(O_n) = p(L_n \mid \neg L_{n-1}, O_n)$.

Then equation (4) can be simplified to:

$$p(L_n \mid O_n) = \lambda(O_n) + [\rho(O_n) - \lambda(O_n)]\, p(L_{n-1} \mid O_n) \qquad (5)$$

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

5 Combining the two phases

Combining (5) with equation (3) gives:

$$p(L_n\,\_\,O_n) = \lambda(O_n) + \frac{[\rho(O_n) - \lambda(O_n)]\,\gamma(O_n)\, p(L_{n-1})}{1 + [\gamma(O_n) - 1]\, p(L_{n-1})} \qquad (6)$$

which can be rewritten as:

$$p(L_n \mid O_n) = \frac{\lambda(O_n) + [\rho(O_n)\gamma(O_n) - \lambda(O_n)]\, p(L_{n-1})}{1 + [\gamma(O_n) - 1]\, p(L_{n-1})} \qquad (7)$$

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

$$p(L_n \mid O_n) = \rho(O_n) \qquad (8)$$

I.e. $\Psi(O_n) = p(L_n \mid O_n)$ when $p(L_{n-1}) = 1$, illustrating the earlier description of $\Psi(O_n)$ as the "rate of remembering".

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

$$p(L_n | O_n) = \lambda(O_n) \tag{9}$$

I.e. $\Sigma(O_n) = p(L_n | O_n)$ when $p(L_{n-1}) = 0$, illustrating the earlier description of $\Sigma(O_n)$ as the "rate of learning".

Figure 2 illustrates equation (7) for some particular values of K, Ψ and Σ. 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 Ψ and Σ (in accord with equations (8) and (9)); and the curvature depends directly on the value of K, i.e. concave when $K > 0$, convex when $K < 0$, and straight when $K = 0$.

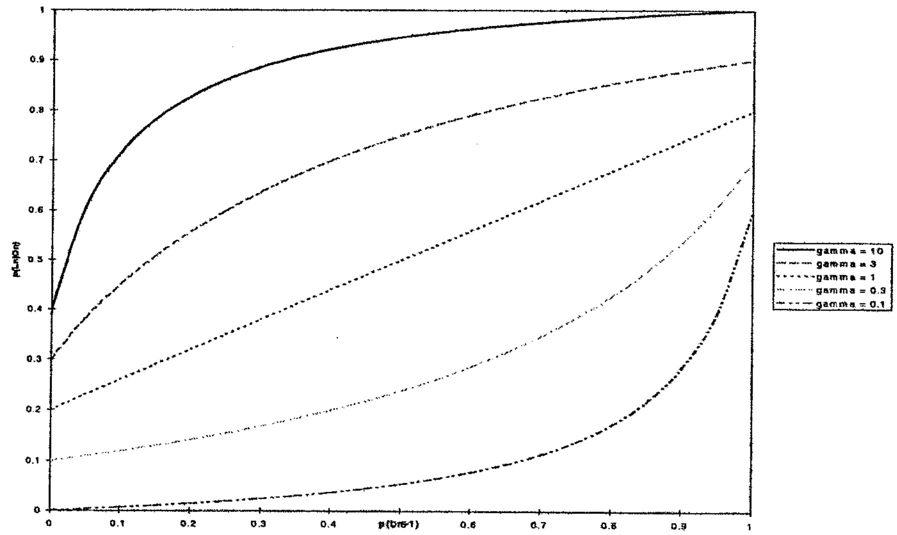

Figure 2. Example curves for equation (7)

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

The ACT Programming Languages Tutor [4] 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 my approach of using dynamic belief networks by showing how Corbett and Anderson's equations can be derived as a special case of my earlier equations. In their paper, Corbett and Anderson did not describe how they derived these equations. Even though they did not make any explicit 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. Here, I show that my work is a generalisation of their model, by adding constraints to my model until I obtain their equations. (This is an extension of the research reported in [2].)

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

(c) $p(L_n \mid \neg L_{n-1}, O_n) = \Sigma(O_n)$ 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 1. 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:

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

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 $p(L_{n-1} \mid E_n)$.

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 $\Sigma\,(O_n)$.

In the most general case, the values of these parameters may be set empirically and may vary from rule to rule. Corbett and Anderson [4] 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 $K\,(C_n) = 4$, $K\,(E_n) = 0.25$ and $\Sigma\,(C_n) = \Sigma\,(E_n) = 0.4$.

7 Another example of a dynamic belief network: SMART

While developing SMART, Shute [5] created a number of functions for updating her student model, as illustrated in Figure 3. The mappings in these functions were developed mainly by hand, based on the opinions of domain experts. Shute's empirical validation of her system, which is based on this approach, makes this student modelling approach worthy of further study.

As is clear from Figure 3, Shute's model is a probabilistic one, thus raising the interesting question as to how it relates to my own work. 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 3.

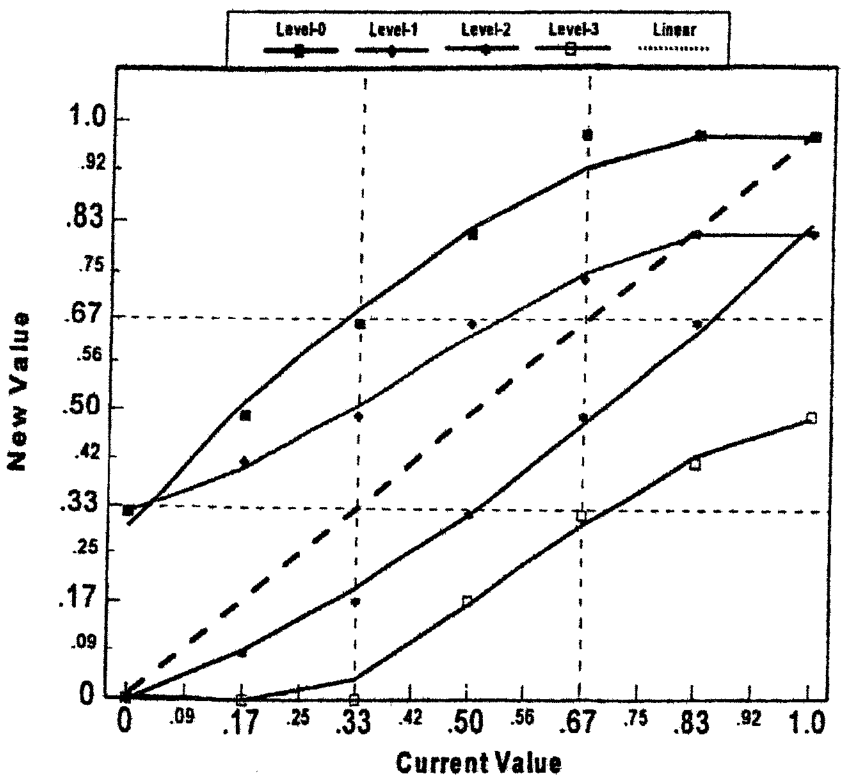

Figure 3. SMART's updating function

Figure 4 shows the curves obtained when plotting equation (7) with the following values for K, Σ and Ψ:

Outcome	K	Σ	Ψ
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

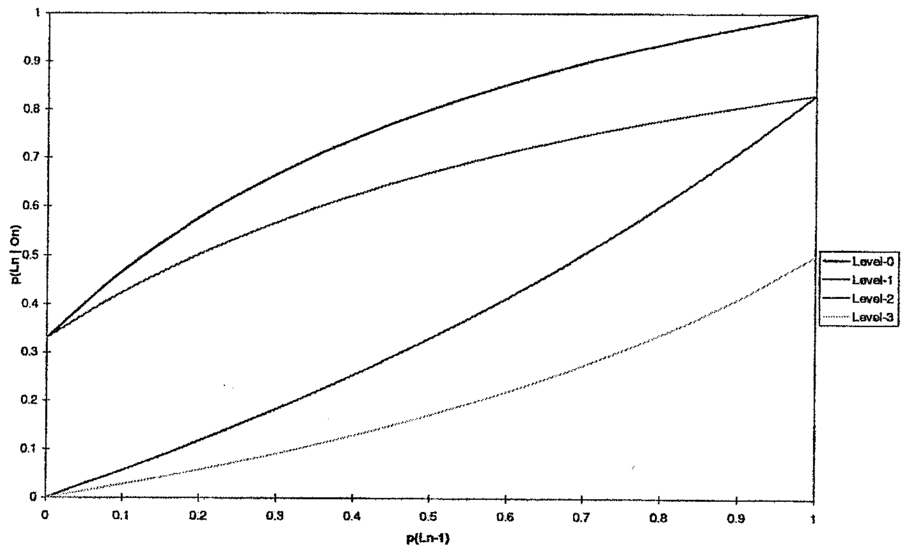

Figure 4. Curves from equation (7) for chosen values of K, Σ and Ψ

By inspecting this figure, it is clear that equation (7) provides a good theoretical basis for Shute's graphs.

8 Conclusion

This paper gives a general theory of how a student model should be updated, based on the concept of a dynamic belief network. This work shows that the student modelling approaches of Corbett and Anderson [4] and Shute [5] are actually special cases of this general approach, even though this is far from obvious at first glance.

References

1. Villano, M. (1992) Probabilistic student models: Bayesian belief networks and knowledge space theory. In Frasson, C., Gauthier, C. and McCalla, G.I. (eds.) *Intelligent Tutoring Systems (Proceeding of the Second International Conference, ITS'92, Montreal, Canada)*, 491-498. Berlin: Springer-Verlag.
2. Reye, J. (1996) A belief net backbone for student modelling. In Frasson, C., Gauthier, C. and Lesgold, A. (eds.) *Intelligent Tutoring Systems (Proceeding of the Third International Conference, ITS'96, Montreal, Canada)*, 596-604. Berlin: Springer-Verlag.
3. Dean, T., and Kanazawa, K. (1989) A model for reasoning about persistence and causation. *Computational Intelligence* 5, 142-150.
4. Corbett, A., and Anderson, J. (1992) Student modeling and mastery learning in a computer-based programming tutor. In Frasson, C., Gauthier, C. and McCalla, G.I. (eds.) *Intelligent Tutoring Systems (Proceeding of the Second International Conference, ITS'92, Montreal, Canada)*, 413-420. Berlin: Springer-Verlag.
5. Shute, V. (1995) SMART evaluation: cognitive diagnosis, mastery learning & remediation. In Greer, J. (ed.) *Artificial Intelligence in Education, 1995*, 123-130. Charlottesville, VA: Association for the Advancement of Computing in Education.