Deriving Prior Distributions for Bayesian Models Used to Achieve Adaptive E-Learning

Sanghyun S. Jeon*

Office of Academic Technology
University of Florida, Gainesville, FL, USA
E-mail: sjeon@cise.ufl.edu

Stanley Y. W. Su

Database Systems R&D Center
University of Florida, Gainesville, FL, USA
E-mail: su@cise.ufl.edu

*Corresponding author

Abstract: This paper presents an approach of achieving adaptive e-learning by probabilistically evaluating a learner based not only on the profile and performance data of the learner but also on the data of previous learners. In this approach, an adaptation rule specification language and a user interface tool are provided to a content author or instructor to define adaptation rules. The defined rules are activated at different stages of processing the learning activities of an activity tree which models a composite learning object. System facilities are also provided for modeling the correlations among data conditions specified in adaptation rules using Bayesian Networks. Bayesian inference requires a prior distribution of a Bayesian model. This prior distribution is automatically derived by using the formulas presented in this paper together with prior probabilities and weights assigned by the content author or instructor. Each new learner's profile and performance data are used to update the prior distribution, which is then used to evaluate the next new learner. The system thus continues to improve the accuracy of learner evaluation as well as its adaptive capability. This approach enables an e-learning system to make proper adaptation decisions even though a learner's profile and performance data may be incomplete, inaccurate and/or contradictory.

Keywords: Adaptive e-Learning; Bayesian Model; Data Uncertainty; Prior Distribution; Group Profile and Performance Data

Biographical notes: Sanghyun S. Jeon received her PhD from the Department of Computer and Information Science and Engineering, University of Florida in 2010. Her research interests include adaptive e-learning systems using Bayesian networks, probabilistic rule-based systems, and content management systems. Currently, she works in the e-learning system development team at the Office of Academic Technology of the University of Florida.

Stanley Y. W. Su is a Distinguished Professor Emeritus and Adjunct Professor of the Department of Computer and Information Science and Engineering, University of Florida. He was the Director of the Database Systems R&D Center of the University of Florida (1977-2005) and served as Editor and Editor-in-Chief of six major journals in database and information system areas. He is an IEEE Fellow.

1. Introduction

Learners have diverse backgrounds, competencies, and learning objectives. An adaptive e-learning system aims to individualize content selection, sequencing, navigation, and presentation based on the profile data provided by learners and the performance data gathered by the system (Brusilovsky & Maybury, 2002). A popular way of guiding an e-learning system to provide individualized instructions to learners is to use condition-action rules (de Bra, Stash, & de Lange, 2003; Duitama, Defude, Bouzeghoub, & Lecocq, 2005). The condition part of a rule is a Boolean expression for examining the profile and/or performance data of a learner that are relevant to an adaptation decision. If the expression is evaluated to be true, the specified adaptation action is taken by the system. A simple example of this rule is "If a learner did not take the prerequisite course and his/her assessment result is below a specified score, the learner is asked to study the content again".

There are three basic problems with e-learning systems that use this type of rule. First, the condition specification of a rule, which can potentially consist of many profile and performance data conditions, is evaluated deterministically to a true or false value instead of probabilistically. This means that the content author or instructor (called 'the expert' in the remainder of this paper) must be able to define the precise data conditions under which an adaptation action should be taken. However, in reality, the expert may not have the full knowledge necessary to specify these precise data conditions. Second, some profile data provided by a learner can be missing, incorrect, or contradictory to his/her performance data. For example, a learner may not be able to tell the system what his/her preferred learning style. Or, a learner may not be willing to provide a piece of personal information (e.g., disability) because of privacy concerns. Even if he/she provides the system with a piece of information, that information may no longer be accurate as time passes (e.g., a learner's preferred learning style may change with time and with the subject he/she takes). Also, some profile data may contradict with performance data (e.g., a learner may specify that he/she has certain prior knowledge of a subject which contradicts with his/her actual performance). These data anomalies can cause serious problems in evaluating the condition specification of a rule; an error made in even a single data condition can cause the entire condition specification to have a wrong evaluation result, and thus can cause the system to take the wrong action. Third, in the traditional rule-based systems, each data condition is evaluated independently. The correlation between data conditions is not taken into consideration. Since the truth value of one data condition may affect that of some other data condition(s) and the truth value of one data condition may have more influence on the truth value of the entire condition clause than that of another data condition, we believe that the correlations among data conditions are important and should be considered.

Using a Bayesian Network (Pearl, 1988) is one approach to handling these problems. Bayesian Networks have been successfully used in some adaptive e-learning systems for assessing a learner's knowledge level (Martin & van Lehn, 1995; Gamboa & Fred, 2001), predicting a learner's goals (Arroyo & Woolf, 2005; Conati, Gertner, & van Lehn, 2002), providing feedback (Gertner & van Lehn, 2000), and guiding the navigation of content (Butz, Hua, & Maguire, 2008). In our previous paper (Jeon, Su, & Lee, 2007b), we also proposed methods and examples to resolve the problems associated with rule-based systems by using Bayesian Networks. Bayesian Networks are used in our work to capture the correlations among the data conditions specified in adaptation rules, represent the profile and performance data of learners in terms of probability values, and evaluate the condition clauses of these rules probabilistically. The probability values are derived from the profile and performance data of a group of learners including the ones who are currently taking an instructional module and the learners who have learned from the same module. Bayesian Networks allow our adaptive e-learning system to make proper adaptation decisions for each new learner even if the learner's profile and performance data are incomplete, inaccurate and/or contradictory.

However, using a Bayesian Network requires setting up a prior distribution (Kass & Wasserman, 1996) which represents a system's initial assumption on the data of previous learners (Neal, 2001). The prior distribution consists of prior probabilities for the root nodes and conditional probabilities for the non-root nodes of a Bayesian model, which is the Bayesian Network that models the correlations among data conditions specified in an adaptation rule. Choosing an appropriate prior distribution is the key for a successful Bayesian inference (Gelman, 2002) because the prior distribution is combined with the probability distribution of new learners' data to yield the posterior distribution, which in turn is treated as the new 'prior distribution' for deriving future posterior distributions. If the initial prior distribution is not informative, it will take a long time for the e-learning system to 'train' the Bayesian Network by using new learners' data so that the proper inference can be made for the next new learner.

Prior distributions can be obtained from different sources and methods. To the best of our knowledge, there is no single commonly accepted method. It would be ideal if a large empirical dataset that contains the profile and performance information of previous learners was available (Gertner & van Lehn, 2000). However, such a dataset is most likely not available for two reasons. First, there is no accepted standard for data that comprehensively characterize a learner's profile and performance, in spite of the fact that several organizations have been working on such a standard (LIP, 2010; PAPI, 2001). Second, the data conditions that are regarded by one domain expert as relevant to an adaptation rule, thus to its corresponding Bayesian model, can be different from those of another expert. The lack of an established standard and difficulty in finding an adequate dataset may explain why some existing adaptive e-learning systems (Gamboa & Fred, 2001; Butz et al., 2008; Conati et al., 2002; García, Amandi, Schiaffino, & Campo, 2007; Arroyo & Woolf, 2005; Desmarais, Maluf, & Liu, 1995) limit themselves to using only easily obtainable data such as test results, questionnaire results, and students' log files instead of using a full range of attributes that characterize learners' profile and performance.

The prior distribution can also be obtained by asking a domain expert (Mislevy et al., 2001), who can be the content author or a person who has prior experiences in instructing learners of that content. However, this is time-consuming and error-prone because the expert will have to accurately and consistently assign prior probabilities to the root nodes and different combinations of conditional probabilities to the non-root nodes of a Bayesian model. Reported literature also does not provide all the required probabilities (Xenos, 2004). A considerable amount of data processing and some additional domain knowledge are still required to derive an informative prior distribution (Druzdzel & van der Gagg, 2000). It has been recognized that obtaining an informative prior distribution is the most challenging task in building a probabilistic network (Druzdzel & van der Gaag, 1995). In this work, we ease the task of acquiring the prior distribution of a Bayesian model by providing a user interface to a domain expert to enter prior probability values for the root nodes and weights for the edges of a Bayesian model, and by introducing three formulas for automatically deriving conditional probability tables (CPTs) for the non-root nodes based on the expert's inputs. This paper is organized in the following way: Section 2 presents our approach for achieving adaptive e-learning by using probabilistic rules and Bayesian models in our e-learning system. Section 3 proposes the formulas that can be used to derive conditional probabilities for these models. The implementation and the evaluation of this approach are described in Section 4. Section 5 summarizes what has been presented and the advantages of the approach.

2. A Probabilistic Approach to Adaptive e-Learning

In our opinion, an adaptive e-learning system must gather and accurately evaluate learner's data, and take the proper adaptation actions to tailor an instruction to suit each learner. In order to resolve the aforementioned problems associated with the use of traditional condition-action rules, our system achieves adaptive properties by using probabilistic rules called 'Event-Condition_probability-Action-Alternative_action (ECpAA) rules'. An ECpAA rule has the format 'on [Event], if [Condition_probability specification] then [Action] else [Alternative_action]'. The 'event' is a particular point in time when the processing of a learning activity is reached. This point in time is called an 'adaptation point' because, at this point (or the occurrence of the event), the 'condition_probability specification' of the rule is evaluated to determine if the 'action' or the 'alternative_action' should be taken. We identify six different events: 'beforeActivity' (the time to bind a learning object to the activity before the learning object is processed), 'afterPreAssessment' (the time after a pre-assessment has been performed), 'drillDown' (the time before going down the activity tree from a parent activity to a child activity), 'rollUp' (the time to return to the parent activity after a child activity has been processed), 'afterPostAssessment' (the time after a post-assessment has been carried out), and 'beforeEndActivity' (the time to exit from the activity).

Corresponding to these events, the domain expert would specify if-then-else rules to be evaluated against some selected profile and performance data of a new learner as well as the meta-data of the learning object being processed to determine the proper adaptations to take (e.g., what and how contents should be presented to a learner, in what order, and what degree of navigation control should be given to the learner). Unlike the traditional condition-action rule, the condition part of an ECpAA rule is specified probabilistically in the form of p(condition specification) $\geq$ x (i.e., the probability of the condition specification being true is greater than or equal to a threshold value x) instead of deterministically (i.e., the condition specification is 100% true or false). The condition specification contains a set of data conditions whose attributes are selected from those that define a learner's profile and performance as well as the meta-data of a learning object. These data conditions are deemed by the domain expert as relevant for making an adaptation decision, and are used by him/her to design a Bayesian model. The structure of this model captures the correlations among the data conditions, and its prior distribution contains probability values that represent the domain expert's subjective estimations of the profile and performance data of previous learners. When the system reaches a particular point in time of processing a learning activity for a new learner, the posting of an event will automatically trigger the processing of the CpAA part of the rule. The Bayesian model is used to evaluate the Cp specification to determine if its probability is greater or equal to the given threshold x. The action or alternative action is then taken accordingly. In this paper, the adaptation rules and their corresponding Bayesian models (BMs) are named after the names of the six events; namely, *beforeActivityRule*, *beforeActivityBM*, etc. They can be optionally defined for some or all of the events. Thus, a maximum of six ECpAA rules and six Bayesian models can be activated at six different stages of processing a learning activity. It is important to point out that adaptation rules specified and Bayesian models designed by one domain expert can be different from those of another expert because they represent subjective opinions of these experts. Also, rules and Bayesian models introduced for different learning activities and for activities of different learning objects that model different courses can also be different. Our system is capable of processing different adaptation rules and Bayesian models.

The action and alternative action clauses of our ECpAA rule specify how the system should 1) select a suitable object, 2) present instructions in a way or format suitable to a particular learner, 3) determine how the child activities of a parent activity should be sequenced, and 4) grant the learner the proper degree of freedom to navigate the content of the sub-tree rooted at the parent activity. In processing the action or alternative action clause, our system employs several adaptive and intelligent techniques such as sorting, conditional text inclusion/exclusion, direct guidance, and link hiding proposed in Hauger and Köck (2007).

Two applications of our adaptive e-learning technology have been developed for the instruction in the use of a Virtual Anesthesia Machine (VAM) to demonstrate our system's adaptive features. VAM is a Web-based anesthesia machine simulator developed by the Department of Anesthesiology at the University of Florida (Lampotang, Lizdas, Gravenstein, & Liem, 2006). The first application is designed to teach the medical personnel in the normal functions and operations of anesthesia machines. The second application instructs the medical personnel in the use of the US Food and Drug Administration's (FDA) pre-use check of traditional anesthesia machines (Jeon, Lee, Lampotang, & Su, 2007a). The example shown in Figure 1 is taken from an implemented learning object, which is a part of our first application (Lee & Su, 2006). The parent activity, Part_3_Safty_Exercises, has six child activities, which are connected to the parent activity by a connector denoted by © . These child activities provide instructions for the six subsystems of an anesthesia machine. We shall use our *rollUpRule* given in Figure 1 as an example to explain the ECpAA rule and its corresponding Bayesian model. The *rollUpRule* is associated with a parent activity and is evaluated based on the learner's performance in its child activities to decide the objective status of the parent. Suppose our *rollUpRule* is specified as follows:

Event: when returning to the parent activity after a child activity has been processed,

Condition_probability: if [p(PL, AL, NFS, AS) $\geq$ 0.60], where PL, AL, NFS and AS are defined in Figure 2,

Action: set Parent-Summary-Status as 'Satisfied' and skip the post-assessment of the parent activity,

Alternative_action: set Parent-Summary-Status as 'Unsatisfied' and carry out the post-assessment.

RollUpBM is designed to compute p(PL, AL, NFS, AS) given in the condition_probability specification of *rollUpRule.* As shown in Figure 2, *rollUpBM* is defined by a Directed Acyclic Graph (DAG) consisting of nodes and edges (Russell & Norvig, 2003). The root nodes (those without parent nodes) are explained below:

PL (Pass Limit): if four out of the six child activities have an assessment score greater than or equal to 70, then PL is true,

AL (Attempt Limit): if the number of attempts does not exceed the number of child activities, then AL is true,

NFS (No Failure Score): if none of the assessment results of the child activities is less than 50, then NFS is true,

AS (Average Score): if the average score of the attempted child activities is greater than or equal to 70, then AS is true, where Average Score = $\frac{\textit{Total Score of Child Activities}}{\textit{Number of Attempted Child Activities}}$.

These root nodes are included in this Bayesian model because they are deemed important for making the roll-up decision by the expert. To specify the correlations among these root nodes, two non-root nodes, Limit Value (LV) and Measure Value (MV), are introduced to form a structure that leads to the final non-root node named Roll Up (RU).

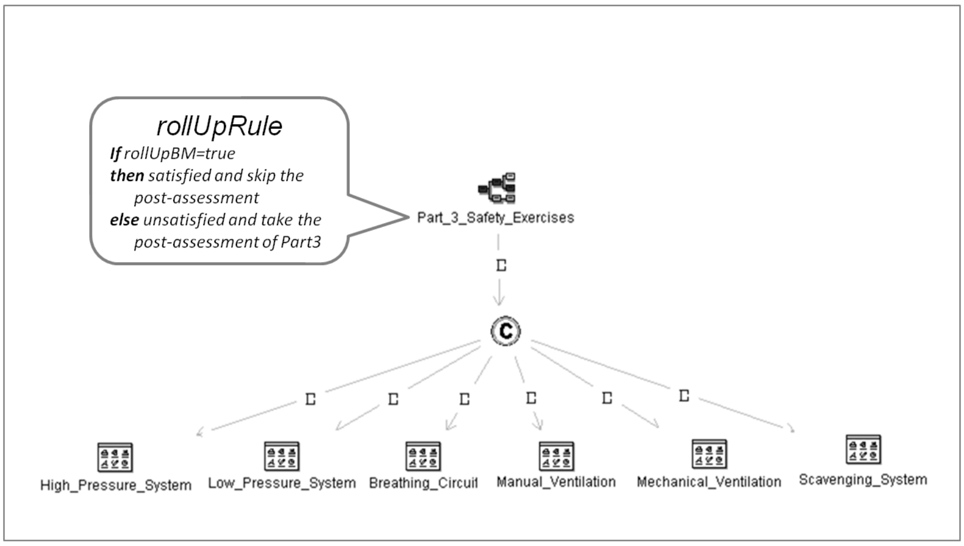


Figure 1. Example of *rollUpRule*

After the specification of the rule's data conditions and the design of the Bayesian model's structure, the prior distribution needed for Bayesian inference must be derived. The prior distribution consists of prior probabilities of the root nodes and CPTs of the non-root nodes. Prior probabilities are assigned to the root nodes based on the expert's knowledge of previous learners. For example, if 90% of previous learners satisfied PL, then the probability of PL being true is 0.9 as denoted by p(PL is true) = 0.9 in Figure 2.

Additionally, weights (i.e., w) can be introduced to the edges that connect the parent nodes to a child node to specify the relative influences of the parent nodes on the child node. For example, as shown in Figure 2, the probability value of PL has more influence on the probability value of LV than that of AL (0.7 vs. 0.3). As we shall show in the next section, the prior probabilities of the root nodes and the weights assigned to all the edges can be used to derive the CPTs for all the non-root nodes. Each table contains entries that show the probability of a child node being true given all the combinations of true and false values of the parent nodes. For example, the probability of MV being true, given that NFS is true (shown by NFS) and AS is false (shown by ~AS), is 0.30 as denoted in Figure 2 by p(MV| NFS, ~AS) = 0.30. Using this prior distribution, *rollUpBM* can determine the probability value of the RU node; if this value is greater than or equal to the threshold specified in the *rollUpRule* (i.e., 0.60), then the action clause of the rule is processed. Otherwise, its alternative action clause is processed. The roll-up decision is made by the system based on a new learner's data as well as the group data. The so-called group data is formed by updating the assigned prior distribution as each new learner's data becomes available to the system. The update results in a posterior probability, which in turn becomes the prior probability for the next new learner. The system updates the prior probabilities of the root nodes and the CPTs of the non-root nodes after a learner completes each stage of processing a learning activity (in this example, the rollup stage). Thus, as more and more learners work through the learning activities of a learning object, the prior distribution of the Bayesian model will become more and more accurate in representing the profile and performance data of previous learners even if the initial prior distribution derived based on the domain expert's inputs is not 100% accurate. The updated prior distribution can thus be used by the system to accurately evaluate the next new learner and take the proper adaptation actions. We have conducted a simulation to show the advantage of continuously updating the probability values of a Bayesian model over not updating the prior distribution by using 1000 simulated users. This simulation and its result can be found in (Jeon & Su, 2010).

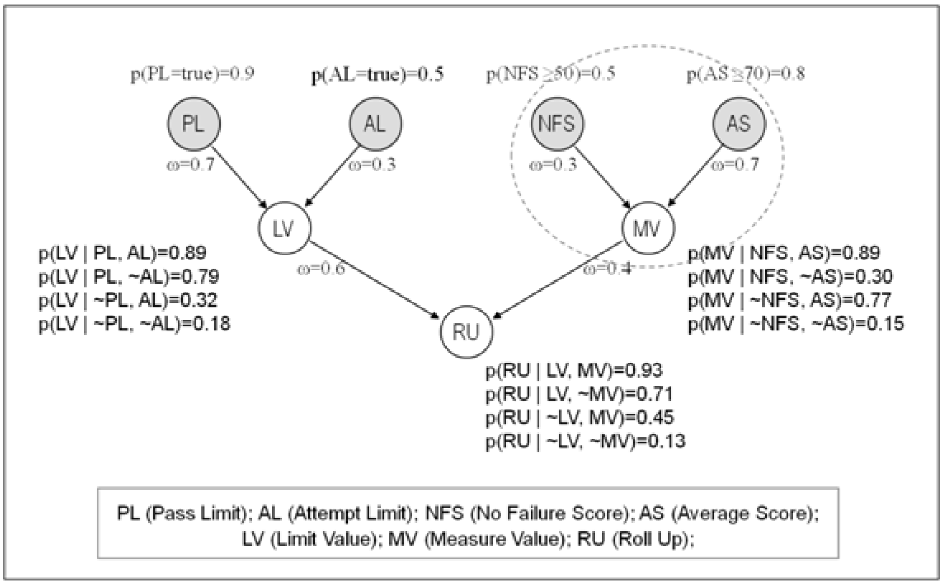


Figure 2. Prior probability distribution and weights of *rollUpBM*

The use of ECpAA rules and Bayesian models for evaluating the Condition_probability clauses of these rules can resolve the data anomalies addressed in the introduction section. In the case of missing data, we use the conditional probability distributions of the data that is correlated with the data attribute that does not have a value. For example, suppose a Bayesian model has two root nodes that specify the data conditions of the following two attributes: 'grade point average' (denoted by GPA) and 'average grade of prerequisites' (denoted by AGP). These two nodes are the parents of a non-root node named as 'prior knowledge' (denoted by PKL). Let us assume that Learner Y satisfies the data condition of GPA, but the value for his/her AGP is missing. In order to derive the conditional probability of PKL given his/her GPA is true and AGP is unknown, we fetch the conditional probability value of PKL given AGP is true (i.e., AGP) and GPA is true (i.e., GPA), and the conditional probability value of PKL given AGP is false (i.e., ~AGP) and GPA is true (i.e., GPA) from the CPT of PKL. Both of these probability values are weighted by the prior probability values of AGP and ~AGP, respectively, and then we take the sum of these weighted probability values, as shown in the following equation (Gonzalez & Dankel, 1993):

$$\begin{aligned} p(PKL|AGP=?,GPA) &= p(PKL|AGP,GPA)*p(AGP)+p(PKL|{\sim}AGP,GPA)*p({\sim}AGP) \\ &= 0.91 * 0.7 + 0.42 * 0.3 = 0.763. \end{aligned}$$

Here, we assume that the values shown in the equation for the corresponding terms are fetched from the Bayesian model. Although the AGP value is not known, as denoted by '?', our system can still derive the conditional probability of PKL. The contradictory data problem can be alleviated by using Bayes' decision rule, which allows the system to select the data condition with a higher conditional probability while minimizing the posterior error (Duda, Hart, & Stork, 2001), and replaces the contradictory data value by one with a higher conditional probability value. Example and the detailed procedure for handling the contradictory problem can be found in (Jeon et al., 2007b). The negative effect of an inaccurate data value can also be reduced because the system considers, not only the inaccurate value associated with a data attribute, but also the values of correlated attributes that are correct and accessible from the CPTs.

The system components that support the ECpAA rule evaluation are shown in Figure 3. When the Learning Process Execution Engine (LPEE), reaches a particular stage of processing a learning activity, its Activity Handler calls the ECpAA Rule Engine, which has two subcomponents: an Event-Trigger-Rule (ETR) Server and a Bayesian Model Processor (BMP). Reaching the roll-up stage is treated as an event by the ETR Server, which fetches the adaptation rule that is linked to the event in a trigger specification. The ETR Server then processes the fetched ECpAA rule. When it processes the Condition_probability specification of the rule (i.e., Cp), it calls the BMP to evaluate the specification and return a probability value. Based on the returned value, the ETR Server processes the action clause or the alternative action clause of the rule. In our implementation, the Bayes Net Toolbox (an open-source MATLAB package) is used to build Bayesian models and perform Bayesian reasoning (Murphy, 2004), and Java's MATLAB interface is used to enable the BMP to communicate with the ETR Server and the repositories. The latter are used to store rules, group profile data, and performance data.

We have implemented the adaptive e-learning system called Gator E-Learning System (GELS). GELS is designed to enable Web users who have the same interest on a subject of learning to form an e-learning community. People in the community play the following major roles: content author, content learner, and community host. A member of the community can play multiple roles. Content authors develop and register learning objects for the virtual community by using our developed learning object authoring tools and repositories. Content learners select and learn from learning objects delivered by GELS through a Web browser. The community host manages software components installed at the host site and communicates with both learners and authors. Therefore, GELS' system components are grouped into three sets installed at different network sites of a virtual e-learning community: the Learning Objects Tools and Repositories (LOTRs) installed at each content author's site for authoring, registering, and storing learning objects; the Adaptive and Collaborative E-learning Service System (ACESS) installed at the community host site for processing adaptive learning activities; and the facility (i.e., Web browser) needed at a content learner site. More details about our system architecture and implementation can be found in (Jeon et al., 2007b).

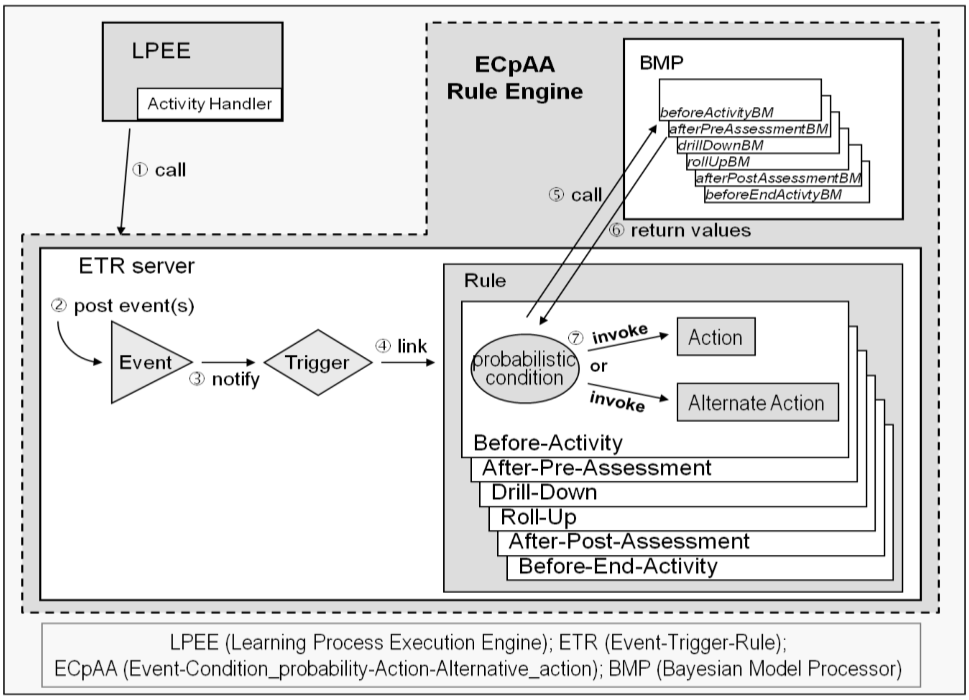


Figure 3. System components for ECpAA rule execution

3. Generating Conditional Probability Tables for Bayesian Models

Before a Bayesian model can be used to process an adaptation rule, a prior distribution (i.e. prior probabilities and conditional probabilities) needs to be derived. While assigning prior probability values to root nodes is relatively simple, assigning conditional probability values to non-root nodes is not. This is because the prior probabilities can be determined by the expert based on the estimated percentages of learners who satisfy the data conditions given in the corresponding adaptation rule. On the other hand, the conditional probabilities consist of multiple values computed from different combinations of true/false values of all the parent nodes to form the CPTs. Our challenge is therefore to automatically derive CPTs for all the non-root nodes using a limited amount of inputs from the expert. Our approach is to ask the expert to assign prior probabilities to root nodes and weights to all the edges of a Bayesian model though our developed user interface, and to introduce three formulas to automatically derive the CPTs. The next subsection explains our approach.

3.1. Deriving initial conditional probability tables

We use a simple example to explain our approach. Figure 4 shows that the truth value of a child node (C) is influenced by two parent nodes P_1 and P_2, and the weights assigned to them to show the relative strengths of their influence. Note that we assume P_1 and P_2 are independent. Here, the conditional probability is the probability of C being true given the probabilities of P_1 and P_2 being true. Suppose each node has two states: true (shown by P_1) and false (shown by $\sim P_1$). There are eight possible conditional probabilities to quantify the parent-child dependency: p(C|P$_1$, P$_2$), p(~C|P$_1$, P$_2$), p(C|~P$_1$, P$_2$), p(~C|~P$_1$, P$_2$), p(C|P$_1$, ~P$_2$), p(~C|P$_1$, ~P$_2$), p(C|~P$_1$ ~,P$_2$), and p(~C|~P$_1$ ~,P$_2$).

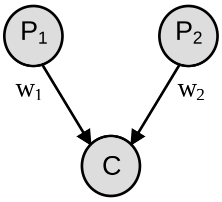


Figure 4. Two-parent-one-child relationship with weights

In order to compute these conditional probabilities, Bayes' rule can be used. For example, p(C|P$_1$, P$_2$) is calculated as:

$$p(C|\,P_1,P_2) \;=\; \frac{p(P_1P_2\,|\,C)\cdot p(C)}{p(P_1P_2)}$$

$$= \frac{p(P_1\,|\,C)\cdot p(P_2\,|\,C)\cdot p(C)}{p(P_1\,|\,C)\cdot p(P_2\,|\,C)\cdot p(C) + p(P_1\,|\!\sim C)\cdot p(P_2\,|\!\sim C)\cdot p(\sim C)} \quad \text{------ (1)}$$

Note that in order to compute p(C| P$_1$, P$_2$), we need to know the numerical values of these six terms: p(C), p(~C), p(P$_1$|C), p(P$_1$|~C), p(P$_2$|C), and p(P$_2$|~C). Calculations of p(C|~P$_1$,P$_2$), p(C|P$_1$,~P$_2$), and p(C|~P$_1$,~P$_2$) can be done in a similar way:

$$p(C\,|\!\sim P_1,P_2) = \frac{p(\sim P_1\,|\,C)\cdot p(P_2\,|\,C)\cdot p(C)}{p(\sim P_1\,|\,C)\cdot p(P_2\,|\,C)\cdot p(C) + p(\sim P_1\,|\!\sim C)\cdot p(P_2\,|\!\sim C)\cdot p(\sim C)}$$

$$p(C\,|\,P_1,\sim P_2) = \frac{p(P_1\,|\,C)\cdot p(\sim P_2\,|\,C)\cdot p(C)}{p(P_1\,|\,C)\cdot p(\sim P_2\,|\,C)\cdot p(C) + p(P_1\,|\!\sim C)\cdot p(\sim P_2\,|\!\sim C)\cdot p(\sim C)}$$

$$p(C\,|\!\sim P_1,\sim P_2) = \frac{p(\sim P_1\,|\,C)\cdot p(\sim P_2\,|\,C)\cdot p(C)}{p(\sim P_1\,|\,C)\cdot p(\sim P_2\,|\,C)\cdot p(C) + p(\sim P_1\,|\!\sim C)\cdot p(\sim P_2\,|\!\sim C)\cdot p(\sim C)}\,.$$

These three equations show that we must know four more terms other than the six terms previously identified. The total ten probabilities required to compute the CPT are shown in Table 1.

Table 1. Ten probabilities required for CPT computation

Probabilities	*p(C)*	*p(P$_1$|C)*	*p(P$_1$|~C)*	*p(P$_2$|C)*	*p(P$_2$|~C)*
	p(~C)	*p(~P$_1$|C)*	*p(~P$_1$|~C)*	*p(~P$_2$|C)*	*p(~P$_2$|~C)*

The values for the probabilities shown in the upper row of Table 1 are complements of the corresponding values shown in the lower row. Within the five probabilities shown in the upper row, there are two pairs, which can be calculated in a similar manner: the method for finding p(P$_1$|C) is the same for finding p(P$_2$|C), only with a different parent. The same goes for p(P$_1$|~C) and p(P$_2$|~C). Therefore, we only need to show how the three highlighted probabilities in Table 1 can be derived in order to compute the CPT. In the remainder of this section, we present the three formulas used for estimating the values of p(C), p(P$_1$|C), and p(P$_1$|~C), respectively.

3.2. Formula 1: weighted sum for p(C)

In order to find p(C), the weighted sum is used. Given $p(P_1)$ and $p(P_2)$, p(C) can be found if relative weights w_1 and w_2 are assigned to P_1 and P_2, respectively, where $0 < w_{1,2} < 1$, and $w_1 + w_2 = 1$.

Formula 1: $p(C) = p(P_1)\cdot w_1 + p(P_2)\cdot w_2$

3.3. Formula 2: correlation coefficient for $p(P_1|C)$

According to the definition of a conditional probability, the conditional probability of P_1 given C is:

$$p(P_1|C) = \frac{p(C\cap P_1)}{p(C)} \text{ --- } (2)$$

Therefore, to find the value of $p(P_1|C)$, we need to know $p(C\cap P_1)$, or 'the intersecting probability of C and P_1', which depends on the correlation coefficient of the two.

- If the relationship between P_1 and C is proportional (i.e., if P_1 is true then C is true, and if P_1 is false then C is false), then the correlation coefficient would be in the range of 0 to 1. A correlation coefficient equal to 1 would mean that $p(C\cap P_1)$ has the maximum value.
- If the relationship is inversely proportional (i.e., if P_1 is true, then C is false and vice versa), then the correlation coefficient would be in the range of –1 to 0. A correlation coefficient equal to –1 would mean that $p(C\cap P_1)$ has the minimum value.
- A correlation coefficient equal to 0 means that P_1 and C are independent. In this case, we can compute $p(C\cap P_1) = p(P_1)\cdot p(C)$ based on the probability independence theory.

If we assume that the relationship between P_1 and C is proportional, then the correlation coefficient must be between 0 and 1. Therefore, our task becomes finding a suitable value in the range of 0 to 1.

In the example of 'two parents (P_1 and P_2) and one child (C)', the influence of P_1 on C can be different from or equal to that of P_2. The relative strengths of their influence are represented by the weights assigned to them. Therefore, we can use these weights to determine the proper correlation coefficient values for $p(C\cap P_1)$ and $p(C\cap P_2)$. Let us use $p(C\cap P_1)_0$ to denote the probability of $C\cap P_1$ when the correlation coefficient is 0, and $p(C\cap P_1)_1$ to denote its probability when the correlation coefficient is 1. Then $p(C\cap P_1)_{w1}$ is the probability of $C\cap P_1$ when the correlation coefficient is w_1. As it lies between $p(C\cap P_1)_0$ and $p(C\cap P_1)_1$, we can get $p(C\cap P_1)_{w1}$ by multiplying the difference $p(C\cap P_1)_1 – p(C\cap P_1)_0$ with the weight of P_1 (i.e., w_1) then adding $p(C\cap P_1)_0$. Thus, the probability of $C\cap P_1$ can be derived by the following equation:

$$p(C\cap P_1) = p(C\cap P_1)_0 + \{p(C\cap P_1)_1 - p(C\cap P_1)_0\}\cdot w_1. \text{ ------------------------------------- } (3)$$

Equation 3 allows us to use the influence of P_1 on C (i.e., the weight) to express the intersection of P_1 and C (i.e., $p(C\cap P_1)$). The value of $p(C\cap P_2)$ can be derived in a similar fashion by replacing P_1 with P_2 and w_1 with w_2.

Formula 2: $p(P_1|C) = \dfrac{p(C \cap P_1)}{p(C)} = \dfrac{p(C \cap P_1)_0 + \{p(C \cap P_1)_1 - p(C \cap P_1)_0\} \cdot w_1}{p(C)}$,

where p(C) is not equal to zero

3.4. Formula 3: complement conversion for p(P_1|~C)

Theoretically, p(P_1|~C) can be derived using the method described in Section 3.3. However, C and ~C have a complementary relationship, thus, p(P_1|~C) can be calculated by using the existing value of p(C) from Formula 1 and that of p(P_1|C) from Formula 2. The formula for its calculation is shown below:

$$p(P_1|{\sim}C) = \frac{p(P_1) - p(C) \cdot p(P_1 \mid C)}{p({\sim}\, C)} \quad \text{---} \ (4)$$

This formula is proven below:

By definition, $p(P_1|C) = \dfrac{p(C \cap P_1)}{p(C)}$, where p(C) is not equal to zero.

Similarly, $p(C|P_1) = \dfrac{p(P_1 \cap C)}{p(P_1)}$, where p($P_1$) is not equal to zero.

So, $p(C \cap P_1) = p(C) \cdot p(P_1|C)$, and $p(P_1 \cap C) = p(P_1) \cdot p(C|P_1)$.

Since, by definition, $p(C \cap P_1) = p(P_1 \cap C)$, we can derive that

$p(P_1 \cap C) = p(P_1) \cdot p(C|P_1) = p(C) \cdot p(P_1|C)$.

Similarly, $p({\sim}C \cap P_1) = p(P_1 \cap {\sim}C) = p(P_1) \cdot p({\sim}C|P_1) = p({\sim}C) \cdot p(P_1|{\sim}C)$. ----------------- (5)

We know from Set theory that

$p(P_1 \cap {\sim}C) = p(P_1 – C) = p(P_1) – p(C \cap P_1) = p(P_1) – p(C) \cdot p(P_1|C)$. ------------------------- (6)

We can derive from equations (5) and (6) that $p({\sim}C) \cdot p(P_1|{\sim}C) = p(P_1) – p(C) \cdot p(P_1|C)$.

Formula 3: $p(P_1|{\sim}C) = \dfrac{p(P_1) - p(C) \cdot p(P_1 \mid C)}{p({\sim}\, C)}$, where p(~ C) is not equal to zero

Formulas 1, 2 and 3 are used to compute the first three probabilities out of the ten listed in Table 1. From those three values, the rest of the probabilities required for the CPT can be derived. By using the three formulas given above, all CPTs can be automatically computed. The expert only needs to provide the prior probabilities of the root nodes and the weights to all the edges of a Bayesian model.

There are two alternative ways to represent p(P_1|C) as shown below:

1. $p(P_1|C) = \dfrac{p(C \mid p_1) \cdot p(P_1)}{p(C)}$, which is based on the Bayes' rule used in Equation 1 to show

$$p(C|\,P_1, P_2) \;=\; \frac{p(P_1P_2\,|\,C)\cdot p(C)}{p(P_1P_2)}$$

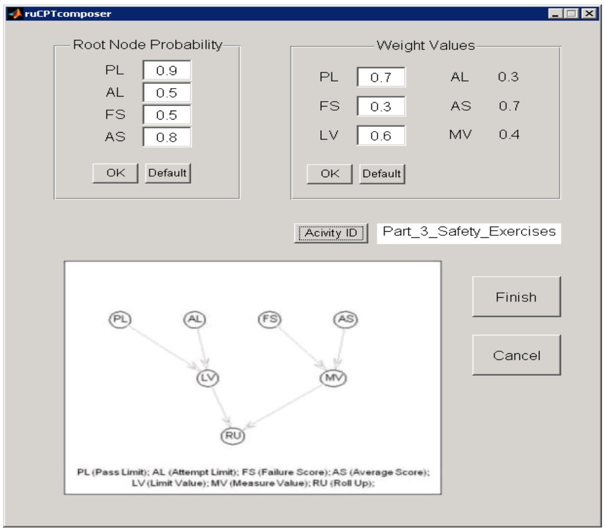
2. $p(P_1|C) = \dfrac{p(C \cap P_1)}{p(C)}$, which is based on the definition of conditional probability; the conditional probability of P_1 given C as shown in equation 2.

We use the second representation instead of the first representation in the derivations of Formulas 2 and 3, because using the set intersection notation '∩' makes it easier for us to explain the three different correlation coefficients given in Formula 2, and also to show that, based on the set theory, $p(P_1 \cap {\sim}C) = p(P_1 – C)$ in the derivation of Formula 3 (see Equation 6).

4. Implementation and Case-based Evaluation

4.1. Implementation and computation: example

Our system provides a graphic user interface, which allows the system to easily obtain all the information necessary to derive the prior distribution of a Bayesian model. This interface is implemented using Matlab and Java. As shown in Figure 5, the interface provides an image of the Bayesian model's structure and allows the expert to assign prior probabilities and weights based on his/her best estimation. Since the sum of the weights of the joined edges is 1.0, when the expert assigns a weight to an edge leading from one parent, the interface automatically sets the weight of the edge leading from the other parent. The system uses these assigned data along with the presented formulas to automatically compute CPTs. Figure 5 shows the assigned values for the example *rollUpBM*.



ruCPTcomposer

Root Node Probability

PL 0.9
AL 0.5
FS 0.5
AS 0.8

Weight Values

PL 0.7 AL 0.3
FS 0.3 AS 0.7
LV 0.6 MV 0.4

Acivity ID Part_3_Safety_Exercises

PL (Pass Limit); AL (Attempt Limit); FS (Failure Score); AS (Average Score);
LV (Limit Value); MV (Measure Value); RU (Roll Up);

Figure 5. Bayesian model editor for assigning prior probabilities and weights in the *rollUpBM*

We now explain the process of generating CPTs using the MV node from Figure 2 as an example. The terms P_1, P_2, C, w_1, and w_2 from Section 3 can now be replaced by NFS, AS, MV, w(NFS), and w(AS) respectively. In *rollUpBM*, after prior probabilities and weights have been assigned by the domain expert, the system uses the three formulas to automatically compute the probability values shown in the right column of Table 2. These probability values are then used to derive the CPT for the MV node as shown in Table 3 by using Bayes' rule (Equation (1)). The CPTs of other non-root nodes of *rollUpBM*, LV and RU, are computed in the same manner, and their results are shown in Figure 2. The derived prior distribution allows our system to aptly evaluate a learner and provide an adaptive e-learning experience to the learner.

Table 2. A set of probability terms/formulas for generating probability values for MV

Probability Term	Formula	Probability value	Probability Term	Formula	Probability value
$p(MV)^1$	$p(NFS)\cdot w(NFS)+p(AS)\cdot w(AS)$	0.71	$p(\sim MV)$	$1-p(MV)$	0.29
$p(NFS\|MV)^2$	$\frac{p(NFS\cap MV)}{p(MV)}$	0.56	$p(\sim NFS\|MV)$	$1-p(NFS\|MV)$	0.44
$p(NFS\|\sim MV)^3$	$\frac{p(NFS)-p(MV)\cdot p(NFS\mid MV)}{p(\sim MV)}$	0.35	$p(\sim NFS\|\sim MV)$	$1-p(NFS\|\sim MV)$	0.65
$p(AS\|MV)^2$	$\frac{p(AS\cap MV)}{p(MV)}$	0.94	$p(\sim AS\|MV)$	$1-p(AS\|MV)$	0.06
$p(AS\|\sim MV)^3$	$\frac{p(AS)-p(MV)\cdot p(AS\mid MV)}{p(\sim MV)}$	0.46	$p(\sim AS\|\sim MV)$	$1-p(AS\|\sim MV)$	0.54

Note: super scripts (^{1,2,3}) denote which of our proposed formulas were used (Section 3).

Table 3. CPT of node MV

Conditional Probability	$p(MV\|NFS, AS)$	$p(MV\|\sim NFS, AS)$	$p(MV\|NFS, \sim AS)$	$p(MV\|\sim NFS, \sim AS)$
Probability Value	0.89	0.77	0.30	0.15

4.2. Evaluation

It is necessary to evaluate the formulas we have proposed to ensure that they provide an informative prior distribution. We introduce seven simulated learners who have different performance data and then apply our approach to determine their roll-up probabilities. The purpose of this evaluation is not to demonstrate the effectiveness of our system in improving learners' ability to learn better and/or faster. This would be a very difficult undertaking because there are too many factors involved in determining a learner's ability to learn and is out of the scope of our current research. Rather, the purpose is to show that, by using the expert's inputs (i.e., prior probabilities for root nodes and weights for edges) and our proposed formulas, the system can automatically generate CPTs for all the non-root nodes to derive an informative prior distribution for the Bayesian model. This section also intends to show the effects of applying the prior distribution in seven cases of simulated learners who have different performance data. We return to the example of Part_3_Safety_Exercises given in Figure 1, and continue to use the *rollUpRule* given in Section 2 and the *rollUpBM* given in Figure 2. The rule says that, **at** the roll-up stage, **if** [p(PL, AL, NFS, AS) ≥ 0.60], **then** set objective status of Part_3_Safety_Excercises as 'Satisfied' and skip the post-assessment of Part_3_Safety_Exercises, **else** set Parent-Summary-Status as 'Unsatisfied' and carry out the post-assessment. Since *rollUpRule* is based on a learner's performance data, the seven learners' performance results of the child activities are given in Table 4.

Table 4. Assessment results and average scores of the simulated learners

Child Activities / Average Score	Nicole	Eva	Michael	Jack	Adam	Steven	Kim
High Pressure System	$\boxed{86}$	$\boxed{74}$	60⇨$\boxed{74}$	$\boxed{100}$	$\boxed{86}$	60⇨$\boxed{74}$	$\boxed{74}$
Low Pressure System	$\boxed{100}$	$\boxed{81}$	46⇨46	$\boxed{100}$	42⇨42	69⇨$\boxed{81}$	54⇨69
Breathing Circuit	$\boxed{86}$	69⇨$\boxed{80}$	$\boxed{86}$	$\boxed{86}$	$\boxed{86}$	51⇨51	37⇨37
Manual Ventilation	$\boxed{81}$	58⇨54	42⇨46	69⇨69	46⇨54	42⇨31	31⇨27
Mechanical Ventilation	X	60⇨60	$\boxed{86}$	66⇨69	$\boxed{86}$	60⇨$\boxed{80}$	34⇨40
Scavenging System	X	69⇨$\boxed{81}$	69⇨$\boxed{81}$	69⇨69	58⇨69	27⇨42	42⇨54
Average Score	$\boxed{88}$	$\boxed{72}$	$\boxed{70}$	$\boxed{82}$	$\boxed{71}$	60	50

(Note) X: no assessment result, ⇨: retry, □: satisfied score, ▨: failed score.

Several notations are used to describe the performance of the learners in detail. The arrow indicates that a learner had to retry a child activity, because the initial score was unsatisfactory. In this experiment, a learner is allowed to retry only once per child node. Boxed numbers indicate satisfactory scores that are greater than or equal to 70, whereas shaded numbers indicate failed scores that are less than 50. Plain numbers indicate unsatisfactory scores. A summary of the *rollUpBM* is provided in Table 5. In our simulation, Nicole, Eva and Michael satisfy the pass limit (PL) in Table 5. Since Nicole satisfies the objectives of her first four child nodes (denoted by PL being true in Table 4), she is not required to take the remaining two child activities. She also has the highest average score (88) and no failed child activities. All of these factors contribute to her high roll-up result (0.86). Michael has four satisfactory scores with an average score of 70, which is above the threshold. However, his two failed child activities and many attempts result in a roll-up probability of 0.78. His roll-up result is higher than the defined threshold (0.60) because PL and AS are weighed much more than AL and NFS.

It is for learners like Jack that our system offers a better adaptive e-learning experience. Jack has an average score of 82, which is almost as high as Nicole's, and has not failed in any child activity (denoted by NFS being true). Unfortunately, he cannot satisfy the data condition PL (Pass Limit). He would have failed if the correlations among the data conditions were not considered. The *rollUpBM* evaluates his result as 60, which meets the defined *rollUpBM*'s threshold (60), because the system not only considers the PL condition but also PL's correlations with other data conditions as shown by the structure of *rollUpBM*. Although PL is weighted heavier than AL and LV is weighted heavier than MV as shown in Figure 2, our system does not allow PL and LV to have absolute influence on the roll-up decision. Rather, it takes all the data conditions and their correlations into consideration to determine that Jack has gained enough knowledge from the instructions given in the child activities and that he can skip the post assessment of the parent activity.

In our user case study, we found that the system can derive a prior distribution based on limited inputs from the expert and the proposed formulas, and use it to accurately evaluate new learners with different performances. As each new learner's data becomes available, it is used to update the prior distribution of a Bayesian model. Thus, the updated prior distribution becomes more and more accurate in representing the characteristics of previous learners. This accumulation of 'group data' will improve the accuracy of evaluating the next new learner and continuously improve the adaptive capability of the system.

Table 5. User case evaluation results

Input / Output	Nicole	Eva	Michael	Jack	Adam	Steven	Kim
PL	T	T	T	F	F	F	F
AL	T	F	F	F	F	F	F
NFS	T	T	F	T	F	T	F
AS	T	T	T	T	T	F	F
p(RU)	0.86	0.81	0.78	0.60	0.57	0.42	0.37

5. Summary and Conclusion

An adaptive e-learning system aims to tailor an instruction to suit each individual learner based on his/her profile and performance data. However, profile data provided by a learner can be incomplete and inaccurate. It may also contradict with the performance data gathered by the system. These data anomalies can cause a rule-based adaptive system to take inappropriate adaptation actions if the traditional condition-action rules are used. In our work, we introduce a new rule specification language and provide a user interface for the domain expert to specify the condition part of an adaptation rule probabilistically instead of deterministically. We use a Bayesian model not only to resolve the data uncertainty but also to evaluate the condition specification of the rule probabilistically. Bayesian models enable our adaptive e-learning system to evaluate and apply the proper adaption rules to tailor an instruction for each new learner in the presence of data anomalies. The conditional probability tables of a Bayesian model are automatically generated based on the expert's input (i.e., the prior probabilities assigned to the root nodes and the weights assigned to the edges that connect the nodes of the model), and the formulas introduced in this paper to derive the prior distribution needed for Bayesian inference. As each new learner's profile and performance data become available, the system uses these data to update the prior distribution, thus improving the accuracy of evaluating the next new learner. Our system has six adaptation points in the processing of each activity of an activity tree, which models a composite learning object. These points give an expert the option of introducing adaptation rules to be activated. They increase the frequency of applying adaptation rules and thus increase the system's adaptive capability. We have evaluated our approach of deriving prior distributions and updating the distributions using simulated learner cases and have found that the approach is effective. It enables the system to deliver individualized instructions to learners with different profiles and performances.

The work reported in this paper deals with 'parameter learning' by updating the probability values of a Bayesian model based on the data of new learners. It does not deal with 'structural learning' by acquiring the structure of a Bayesian model based on learners' data. The latter is a very challenging problem that has been investigated by many researchers as reported in (Cooper & Herskovits, 1992; Spirtes, Glymour, & Scheines, 1993; Buntine, 1994; Lam & Bacchus, 1994; Heckerman, Geiger, & Chickering, 1997). It is out of the scope of our current research.

References

1. Arroyo, I., & Woolf, B. (2005). Inferring learning and attitudes from a Bayesian network of log file data. In C. K. Looie, G. McCalla, B. Bredeweg, & J. Breuker (Eds.), *Proceedings of the 12th International conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology* (pp. 33−40). Amsterdam: IOS Press.

2. Brusilovsky, P., & Maybury, M. T. (2002). From Adaptive Hypermedia to the Adaptive Web. *Communications of the ACM, 45*(5), 30−33.

3. Buntine, W. L. (1994). Operations for Learning with Graphical Models. *Journal of Artificial Intelligence Research, 2,* 159−225.

4. Butz, C. J., Hua, S., & Maguire, R. B. (2008). Web-Based Bayesian Intelligent Tutoring Systems. *Evolution of the Web in Artificial Intelligence Environments: Studies in Computational Intelligence, 130,* 221−242.

5. Conati, C., Gertner, A., & van Lehn, K. (2002). Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User Modeling and User-Adapted Interaction, 12*(4), 371−417.

6. Cooper G. F., & Herskovits E. (1992). A Bayesian Method for the induction of probabilistic networks from data. *Machine Learning, 9,* 309−347.

7. de Bra, P., Stash, N., & de Lange, B. (2003). AHA! Adding adaptive behavior to websites. *Proceedings of the NLUUG 2003 Conference.* Ede, Netherlands, May 21−31.

8. Desmarais, M. C., Maluf, A., & Liu, J. (1995). User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction, 5*(3-4), 283−315.

9. Druzdzel, M. J., & van der Gaag, L. C. (1995). Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information. *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence* (UAI-95), 141−148.

10. Druzdzel, M. J., & van der Gaag, L. C. (2000). Building Probabilistic Networks: Where Do the Numbers Come From? Guest Editors' Introduction. *IEEE Transactions on Knowledge and Data Engineering, 12*(4), 481−486.

11. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification,* (2nd ed.). New York: John Wiley & Sons.

12. Duitama, F., Defude, B., Bouzeghoub, A., & Lecocq, C. (2005). A framework for the generation of adaptive courses based on semantic metadata. *Multimedia Tools and Applications, 25*(3), 377−390.

13. Gamboa, H., & Fred, A. (2001). Designing Intelligent tutoring Systems: A Bayesian Approach. *3rd International Conference on Enterprise Information Systems* (ICEIS 2001) (pp. 452−458). Setubal, Portugal, July 7–10.

14. García, P., Amandi, A., Schiaffino, S., & Campo, M. (2007). Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education, 49* (3), 794−808.

15. Gelman, A. (2002). Prior distribution. *Encyclopedia of Environmetrics* (Vol. 3, pp. 1634−1637). Chichester: John Wiley & Sons, Ltd.

16. Gertner, A., & van Lehn, K. (2000). Andes: A Coached Problem Solving Environment for Physics. In G. Gauthier, C. Frasson, & K. van Lehn (Eds.), *Intelligent Tutoring Systems: 5th International Conference*, *Lecture Notes in Computer Science*, *(1839),* (pp. 133−142). Springer: Berlin.

17. Gonzalez, A. J., & Dankel, D. D. (1993). *The Engineering of Knowledge-Based Systems: Theory and Practice.* NJ: Prentice-Hall.

18. Hauger, D., & Köck, M. (2007). State of the Art of Adaptivity in E-Learning Platforms. *Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems* (ABIS 2007) (pp. 355−360). Hildesheim, Germany, September, 24−26.

19. Heckerman, D., Geiger, D., & Chickering, D. M. (1997). Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning, 20 (3),* 197−243.

20. IEEE LTSC (PAPI). (2001). *IEEE LTSC 1484.2-Standard for Information Technology-Public and Private Information (PAPI) for Learners Specification, version 6.* Retrieved July 10, 2010 from the World Wide Web: http://www.cen-ltso.net/Main.aspx?put=230.

21. IMS Global Learning Consortium (LIP). (2010). *IMS Learner Information Package Specification.* Retrieved July 10, 2010 from the World Wide Web: http://www.imsglobal.org/profiles/.

22. Jeon, S. S., Lee, G., Lampotang, S., & Su, S. Y. W. (2007a). An integrated e-learning system for simulation based instruction of anesthesia machine. *International Journal of Knowledge and Learning (IJKL), 3*(1), 106−120.

23. Jeon, S. S., Su, S. Y. W., & Lee, G. (2007b). Achieving Adaptive E-Learning Using Condition-Action Rules and Bayesian Networks. *Proceedings of the IADIS International Conference, e-LEARNING* 2007 (pp. 359−366). Lisbon, Portugal, July 6-8.

24. Jeon, S. S., & Su, S. Y. W. (2010). Adaptive E-Learning Using ECpAA Rules, Bayesian Models, and Group Profile and Performance Data. *International Journal of Learning Technology, 5*(4), 415−434.

25. Kass, R. E., & Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association, 91*(435), 1343−1370.

26. Lam, W., & Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence, 10*(4), 269−293.

27. Lampotang, S., Lizdas, D. E., Gravenstein, N., & Liem, E. B. (2006). Transparent Reality, a Simulation Based on Interactive Dynamic Graphical Models Emphasizing Visualization. *Educational Technology, 46*(1), 55−59.

28. Lee, G., & Su, S. Y. W. (2006). Learning object models and an e-learning service infrastructure. *International Journal of Distance Education Technology, 2*(1), 1−16.

29. Martin, J., & van Lehn, K. (1995). A Bayesian Approach to Cognitive Assessment. In P. Nichols, S. F. Chipman, & R. L. Brennan, (Eds.), *Cognitively Diagnostic Assessment* (pp. 141−165). Hillsdale, NJ: Erlbaum.

30. Mislevy, R. J., Senturk, E., Almond, R., Dibello, L., Jenkins, F., Steinberg, L., & Yan, D. (2001). Modeling Conditional Probabilities in Complex Educational Assessments. *CSE Technical Report 580.* Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Center for Studies in Education, UCLA.

31. Murphy, K. P. (2004). *The Bayes Net ToolBox for Matlab.* Retrieved from July 10, 2010 from the World Wide Web: http://code.google.com/p/bnt/.

32. Neal, R. M. (2001). Defining Priors for Distributions Using Dirichlet Diffusion Trees. *Technical Report No. 0104,* Department of Statistics, University of Toronto. Retrieved from July 10, 2010 from the World Wide Web: http://www.cs.utoronto.ca/~radford/ftp/dft-paper1.pdf.

33. Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* CA: Morgan Kaufmann Publishers.

34. Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach,* 2nd edition. NJ: Prentice-Hall.

35. Spirtes P., Glymour C., & Scheines R. (1993). *Causation, Prediction and Search*. New York: Springer Verlag.

36. Xenos, M. (2004). Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks. *Computers & Education, 43,* 345–359.