Bayesian Agent in e-Learning

Maomi Ueno
University of Electro-Communications
Japan

1. Introduction

The constructivist approach has pervaded the area of educational technology in recent decades. It has been argued in this approach that the responsibility for learning should be increasingly with the learner (Von Glasersfeld, 1995). Therefore, the role of instructor has changed to facilitator from that of teacher (Bauersfeld, 1995). A teacher gives a didactic lecture that covers the subject matter, but a facilitator assists the autonomous learning process. The learner plays a passive role in the former scenario and in the latter the learner plays an active role in the learning process. The emphasis thus shifts from the instructor and content-centred approach toward the learner-centred approach (Gamoran, Secada & Marrett, 2000).

A central feature of this facilitation is individualizing learners and helping them to achieve self-growth through self-evaluation and cooperation with others (Merriam & Brockett, 2007). For example, according to the well known theory by Knowles, facilitation is designing a pattern of learning experiences, conducting these learning experiences with suitable techniques and materials, and evaluating the learning outcomes and rediagnosing learning needs (Knowles, 1983) (Knowles, Holton & Swanson , 1998). e-Learning, which emerged as a method of attaining the learner-centred approach, provides a new autonomous-learning environment that combines 1. multimedia content, 2. collaboration among learners, and 3. computer-supported learning (Ueno 2007). e-Learning should work even if there is no human facilitator and a huge number of learners participate in it. It would essentially be impossible for facilitators to individualize such a huge number of learners and facilitate their learning. The main idea in this paper is that a computational agent in a Learning Management System (LMS) plays the role of facilitator instead of human teachers. The proposed agent uses the learners' history data, which is stored in a database, to individualize learners. A computational agent that learns using machine learning or data-mining technologies from data is called a "learning agent". This paper proposes a learning agent for e-Learning.

First, the agent predicts a learner's final status (1. Failed, 2. Abandoned, 3. Successful, or 4. Excellent) from his/her current learning-history data using a Bayesian network that is constructed from the his/her past learning-history data. The agent compares a learner's learning processes with past excellent learners' learning processes in the database, diagnoses the learner's learning processes, and generates adaptive instructional messages to guide the learner.

In addition, some previous research on learning motivation found that the effects of a mentor's motivational messages were adapted to a learner's status in e-Learning. Visser and Keller (1990) reported that motivational messages could reduce dropout rates and later attempted to improve motivation in e-Learning situations using such messages (Visser, Plomp, and Kuiper, 1999). Gabrielle (2000) applied technology-mediated instructional strategies to Gagne's events of instruction and demonstrated how these strategies affected motivation. Thus, agent messages are also expected to be effective in facilitating learner motivation.

A similar idea to that in this study has been proposed by Ueno (2005). He developed an LMS in which the teacher is substituted for an agent as a virtual facilitator. The intelligent agent provides adaptive messages to learners using learner models represented by the decision-tree model (Quinlan, 1986).

Furthermore, some experiments reported in the Ueno's paper demonstrated the effectiveness of this method, but there are still three main problems:

1. The decision-tree model has a cold-start problem and the agent cannot draw any inferences when we provide new courses.
2. The decision-tree model does not predict the target variable until the data for all the other variables have been obtained, because the decision-tree model cannot deal with any missing data.
3. It is difficult for the decision-tree model to provide the reasoning for prediction.

While on the other hand, the proposed agent based on Bayesian networks has three main advantages:

1. The Bayesian-network model can avoid the cold-start problem by providing a valid prior belief for the network structure, even if there are not sufficiently large amounts of data for learning a Bayesian network.
2. The Bayesian-network model can predict the target variable even if the data for all the other variables have not been obtained.
3. The Bayesian network can provide the reasoning for prediction.

Furthermore, this paper shows that the proposed agent is effective as a virtual facilitator from some experiments and actual data.

2. Related work

Various studies have been done that have applied data-mining techniques to learning-history data in e-Learning.

Becker and Vanzin (2003) tried to detect meaningful patterns of learning activities in e-Learning using the association rule.

Minaei-Bidgoli, Kashy, Kortemeyer, and Punch (2003) proposed a method of predicting a learner's final test score by using a combination of multiple classifiers (CMC) constructed from learning-history data in e-Learning, and they reported that a modified method using a genetic algorithm (GA) could improve the accuracy of prediction.

Talavera and Gaudioso (2004) and Hamalainen, Laine, and Sutinen (2006) proposed a method of predicting final test scores using the naïve Bayes model obtained from learning-history data in e-Learning.

However, these studies only tried to predict the learner's performance in e-Learning from learning-history data, and therefore, they did not discuss how to effectively utilize the predicted data-mining results to improve the learners' results. Furthermore, the data-mining
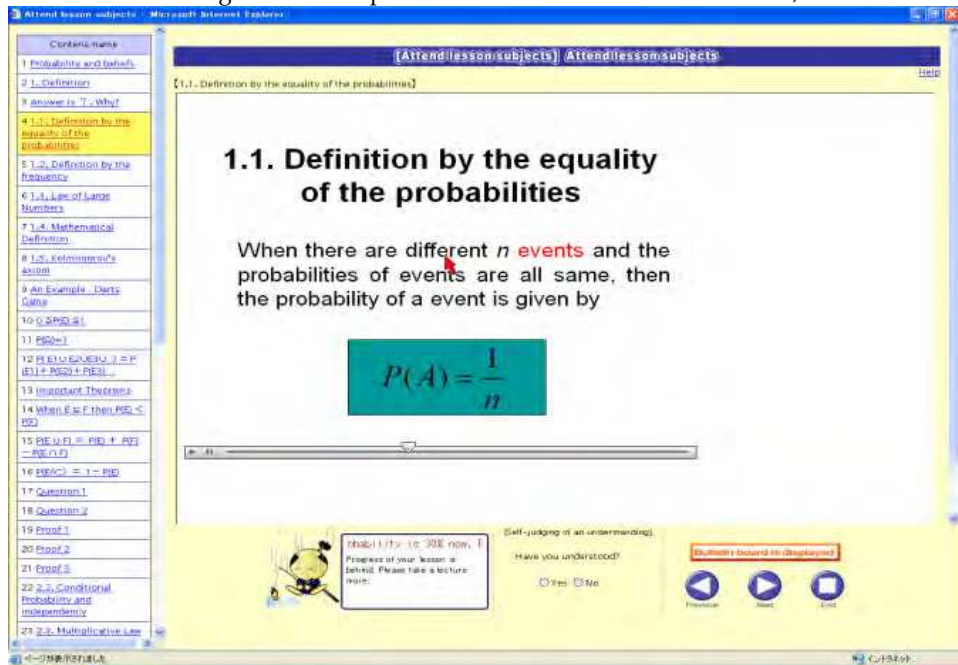

Fig. 1. LMS "Samurai"

engines employed in these studies were not installed into an LMS to automatically analyze the learning-log database.

Here, the author does not simply propose a system of predicting a learner's final status using a data-mining technique, but an agent that acquires domain knowledge related to the content from a learning-history-log database that automatically generates adaptive instructional messages to guide the learners.

3. LMS "Samurai"

The author has developed an LMS called "Samurai"(Ueno, 2004) that is used in many e-Lear ning courses (128 e-Learning courses are now offered by the University of Electro-Communi cations through the LMS). The LMS consists of a content presentation system (CPS), a conte nt database (CD), computer supported collaborative learning (CSCL), a learning history data base (LHD), and a data mining system (DMS). The CPS integrates various kinds of content and presents the integrated information on a Web page. Figure 1 shows typical e-Learning c ontent presented by Samurai. The content is presented by clicking on the menu button. A so und track of the teacher's narration is also presented based on research by Mayer and Ander on (1991), and the red pointer moves automatically as the narration continues. This lesson co

rresponds to a 90-minute lecture at university and includes 42 topics. Although the content i n Figure 1 is text, the system also provides illustrations, animation or computer
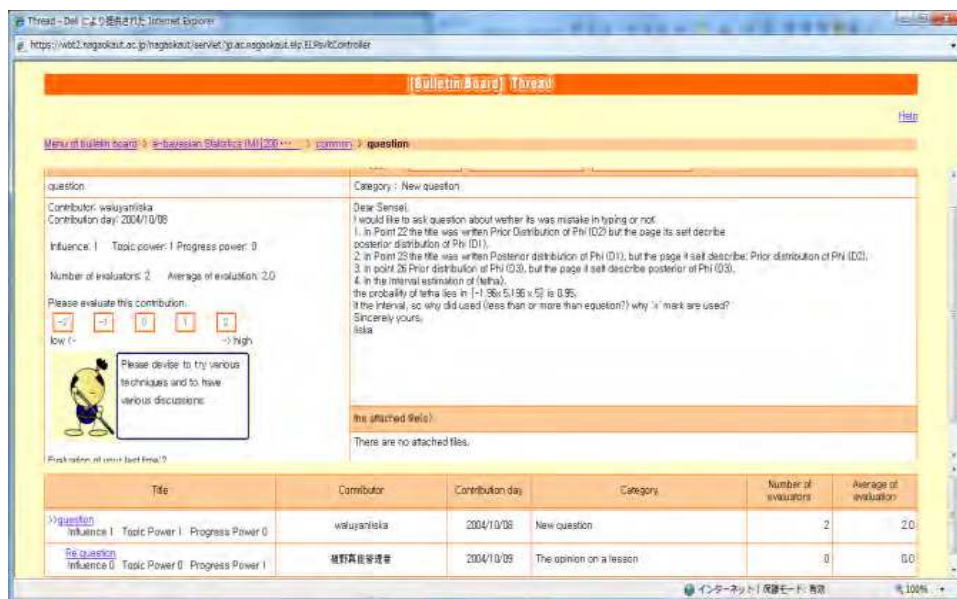

Fig. 2. CSCL in Samurai

lesson corresponds to a 90-minute lecture at the university and includes 42 topics. Although the content in Figure 1 is text, the system also provides illustrations, animation or computer graphics, and video clips. In this lesson, there are 11 items of text content, 11 illustrations, 10 animations, and 10 video clips. The system also presents some test items to assess the learners' degree of comprehension as soon as the lessons have been completed. The CD consists of various kinds of media, such as text, jpeg and mpeg files. The teacher prepares a lecture and saves the content on a CD. Then the CPS automatically integrates the content, and presents this to the learners.

They can also share ideas, questions, and the products of their learning for a given task (e.g., a report or a program source) using the CSCL shown in Figure 2.

The LMS monitors learners' learning processes and stores them as log data in the LHD. The stored data consist of a content ID, a learner ID, the number of topics that the learner has completed, a test-item ID, a record of data input into the DB, an operation-order ID (which indicates what operation was done), a date and time ID (which indicates the date and time that an operation started), and a time ID (which indicates the time it took to complete the operation). These data enable the system to recount the learner's behavior in e-Learning.

4. Bayesian network

A Bayesian network, a Bayesian belief network, or just a belief network is a probabilistic graphical model that allows us to represent and reason about an uncertain domain. A

Bayesian network is represented as a directed acyclic graph of nodes in Figure 3. The nodes in a Bayesian network represent a set of random variables from the domain. A set of directed arcs connects pairs of nodes, representing the direct dependencies between variables. That is, $A \longrightarrow B$ indicates that A causes B. The nodes that the target node depends on are called "parent nodes" of the target node. For $A \longrightarrow B$, A is a parent node of B. Once the topology of the BN is specified, the conditional probabilities corresponding to all arcs should be given. For $A \longrightarrow B$, the value of the conditional probability, $p(B \mid A)$, should be set. If a node has a known value, it is said to be an *evidence* node. Then, the belief probabilities about all the other nodes in the network are updated using a Bayes theorem from the evidence data.

The Bayesian network is mathematically formulated as follows. Let $U = \{x_1, x_2, \ldots, x_N\}$ be a set of N discrete variables; each can take values in the set $\{1, \ldots, r_i\}$. We write $x_i = k$ when we observe that variable x_i is state k. We use $p(x_i = k \mid x_j = k', \square)$ to denote the probability of a person with background knowledge $\square$ for observation $x_i = k$ given observation $x_j = k'$.

When we observe the state for all variables in set U, we call this set of observations an instance of U. We use $p(Y \mid Z, \square)$ to denote the set of probabilities for all possible observations of Y given all possible observations of Z, where $Y \subset U$, $Z \subset U$, and $Y \cap Z = \square$.

A Bayesian network represents a joint probability distribution over domain U by encoding assertions of conditional independence as well as a collection of probability distributions. From the chain rule of probability we know

$$p(x_1, x_2, \cdots, x_N \mid \xi) = \prod_{i=1}^{N} p(x_i \mid x_1, x_2, \cdots, x_{i-1}, \xi)$$

(1)

For each variable x_i, let $\prod_i \subseteq \{x_1, x_2, \ldots, x_{i-1}\}$ be a set of variables called parent nodes that renders x_i and is conditionally independent.

That is,

$$p(x_i \mid x_1, x_2, \cdots, x_{i-1}, \xi) = p(x_i \mid \Pi_i, \xi)$$

(2)

A Bayesian network is represented as a pair of a network structure B_S that encodes the assertions of conditional independence in this equation and a set of conditional probability parameters B_P, (B_S, B_P).

Parameter B_S is a directed acyclic graph such that (1) each variable in U corresponds to a node in $B_{S'}$ and (2) the parents of the node corresponding to $\square$ are the nodes corresponding to the variables in $\prod_i$. After this, we will use x_i to refer to both a variable and its corresponding node in a graph. Associated with node x_i in B_S are the probability distributions $p(x_i \mid \prod_i, \square)$.

B_P is the union of these distributions. When (1) and (2) are combined, we can see that any network for U uniquely determines a joint probability distribution for U. That is,

$$p(x_1, x_2, \cdots, x_N \mid B_S) = \prod_{i=1}^{N} p(x_i \mid \Pi_i, B_S)$$

(3)

The problem of learning a Bayesian network can be stated informally as the following: Given training data $X = \{x_1, x_2, \ldots, x_N\}$, find a network, B, that best matches X.

The common approach to this problem is to introduce a scoring metric that evaluates each network with respect to the training data. Then, it is possible to search for the best network according to this function.

Let $\Box_{ijk}$ be the conditional probability parameters of $x_i = k$ when the j-th instance of the parents of x_i is observed (we write $\prod_i = j$). Buntine (1991) assumed a Dirichlet prior and employed an unbiased estimator, the expectation of the Estimated A Priori (EAP), as the parameter estimator, $\hat{\theta}_{ijk}$ .That is,

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + n_{ijk}}{\alpha_{ij} + n_{ij}}, (k = 0, \cdots, r_i - 2) \tag{4}$$

where n_{ijk} is the number of samples of $x_i = k$ when $\prod_i = j$ and α_{ijk} is the hyper-parameter of the Dirichlet prior corresponding to n_{ijk}, $n_{ij} = \sum_{k=0}^{r_i-1} n_{ijk}$, $\alpha_{ij} = \sum_{k=0}^{r_i-1} \alpha_{ijk}$, and $\hat{\theta}_{ij(r_i-1)} = 1 - \sum_{k=0}^{r_i-2} \theta_{ijk}$.

The predictive distribution is obtained as

$$p(X \mid B_S) = \int_{\Theta} p(X \mid \Theta, B_S) p(\Theta) d\Theta$$

$$= \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_{k=0}^{r_i-1} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\alpha_{ijk}} \tag{5}$$

where $\Theta = \{\theta_{ijk}\}(i = 1, \cdots, N, j = 1, \cdots, q_i, k = 0, \cdots, r_i - 1)$, and q_i signifies the number of instances of $\prod_i$, $q_i = \prod_{x_l \in \prod_i} r_l$.

In particular, Heckerman, et al. (1995) presented a sufficient condition for satisfying the likelihood equivalence assumption as the following constraint related to hyper-parameters:

$$\alpha_{ijk} = \alpha p(x_i = k, \prod_i = j \mid B_S^h), \tag{6}$$

where α is the equivalent sample size (ESS) determined by users and B_S^h is the hypothetical Bayesian-network structure that the user constructs with his/her prior knowledge. These score metrics are designated as Bayesian Dirichlert equivalence (BDe) score mertrics.

That is, even if there is not a sufficiently large amount of data, a Bayesian network can be constructed by using user's prior knowledge. In this paper, we solve the cold-start problem of learning the agent system using these unique advantages of the Bayesian network.

5. Bayesian agent

5.1 Prediction of learner' final status

The main idea here is to apply a data-mining method to the huge amount of stored data and construct a learner model to predict each learner's final status: (1) Failed (Final examination score below 60), (2) Abandoned (The learner has withdrawn before the final examination), (3) Successful (The final-examination score is more than 60 but less than 80; and (4) Excellent (The final examination mark is more than 80.) The well-known data-mining

method of the Bayesian network is employed for this propose using nine variables reflecting each learner's status each week:

1. The number of topics that the learner has learned.
2. The number of times the learner accessed the e-Learning system.
3. The average number of times the learner has completed each topic. (This implies the time the learner repeated each topic.)
4. The average learning time for each lecture, which consists of several types of content and runs for 90 minutes)
5. The average degree of understanding for each topic (This is measured by the responses to questions corresponding to each topic)
6. The average learning time for each course, which consists of fifteen lectures
7. The average number of times the learner has changed the answers to questions in e-Learning
8. The number of times that the learner has posted opinions or comments to the discussion board.
9. The average learning time for each topic.

This section explains how Bayesian networks are learned from learning-history data. Fifteen Bayesian-network structures are estimated corresponding to data from learners' learning histories for the fifteen weeks because all courses run for 15 weeks.

Bayesian networks also suffer from the cold-start problem and no inferences can be drawn when we provide new courses. To solve this problem, this paper uses the prior distribution in (6) for learning a Bayesian network. In detail, a Bayesian-network structure is first estimated using data from all learning histories stored in the database; this does not include data corresponding to the target course. Here, this data from learning histories is called "prior data". The main idea is that the estimated structure from the prior data is used for the prior hypothetical structure, B_S^h, in (6). Next, based on this estimated prior hypothetical structure B_S^h, the Bayesian network is learned by maximizing the BDe in (5) from learning-history data corresponding to the target course. This proposed method enables us to solve the cold-start problem when we start a new course. When there are not sufficiently large amounts of data from the learning histories for the target course, the Bayesian agent follows the estimated Bayesian-network structure from all learning-history data, i.e., the prior data. When there are sufficiently large amounts of data for the target course, the Bayesian agent follows the estimated Bayesian-network structure for the target course. The ESS value, α, means the pseudo-sample size reflecting the prior data and this has been determined as 100.0 in this research.

In addition, we employ the Bayesian-network classifer model since the target variable is one variable (Friedman, Geiger, & Goldszmidt. 1997). In detail, we first add arcs between the final status node and all the explanatory variables, and then construct the network structure between the explanatory variables to maximize BDe given the previously drawn arcs. Here, the greedy search algorithm is employed to learn the network structure.

Figure 3 shows the Bayesian network estimated by maximizing BDe in (5) from prior data (data from 4,344 learners in 64 courses). The network in Figure 6 was propagated using the history data from a learner's fourth week of learning. Furthermore, the probabilities of the variables corresponding to the nodes in Figure 3 indicate the prior-belief probabilities for

the categories. For example, the node corresponding to the predicted final status of a learner indicates that the probability for "abandoned" is 21.0%, the probability for "failed" is 28.4%, the probability for "successful" is 24.4%, and the probability for "excellent" is 26.2%, when there are no data about the learner.

5.2 Learning Bayesian networks using prior belief

This section explains about how to learn Bayesian networks from learning history data. Fifteen Bayesian network structures are estimated corresponding to learners' learning histories data for the fifteen weeks because all courses run for 15 weeks.

The Bayesian network also has the cold start problem and it can not draw any inferences when we provide a new course. To solve this problem, this paper uses the prior distribution in (6) for learning a Bayesian network. For details, first a Bayesian network structure is estimated using all learning histories data which stored in database and does not include the data corresponding the target course. Here, this learning histories data is called as "prior data". The main idea is that the estimated structure from the prior data is used for the prior hypothetical structure B_S^h in (6). Next, based on the estimated prior hypothetical structure B_S^h, the Bayesian network is learned by maximizing the BDe in (5) from learning histroies data corresponding to the target coourse. This proposed method enables to solve the cold start problem when we start a new course.When there are nor sufficiently large data of learning histories data for the target course, the Bayesian agent follows the estimated Bayesian network structure from all learning histories data, the prior data. When there are
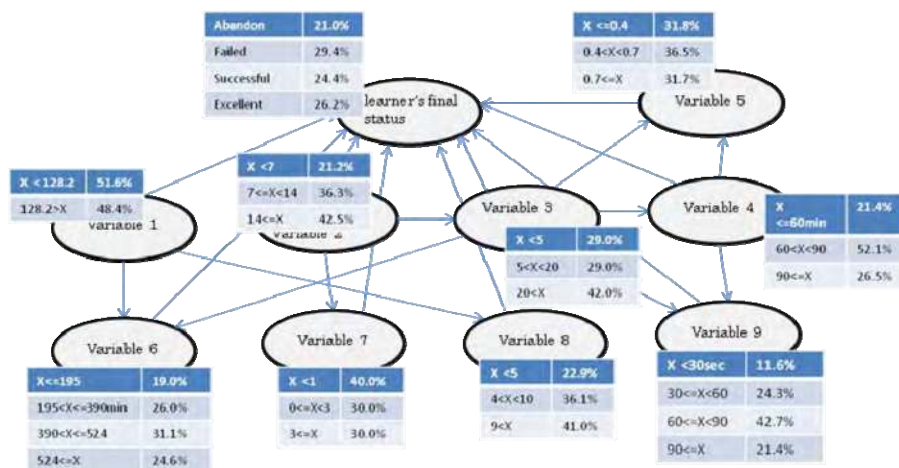

Fig. 3. Example of Bayesian-network structure learned from prior data

Fig. 4. Intelligent agent system (Note that the presented message has not been misspelled. The message is continuously moving within the frame.)

Fig. 5. Various actions by agent

sufficiently large data for the target course, the Bayesian agent follows the estimated Bayesian network structure for the target course. The ESS value α means the pseudo sample size reflecting the prior data and is determined as 100.0 in this research.

In addition, we employ the Bayesian network classifer model since the target variable is one variable(Friedman, Geiger, & Goldszmidt. 1997).

Figure 3 hows the estimated Bayesian network by maximizing BDe in (5) from the prior data (4,344 learners' data to 64 courses). The network in Figure 5 is estimated using 14 weeks of learning history data. Furthermore, the probabilities of the variables corresponding to the nodes in figure 3 indicate the prior belief probabilities for the categories. For example, the node corresponding to the predicted final status of a learner indicates that the probability of "abandon" is 21.0%, the probability of "failed" is 28.4, the probability of "successful" is 24.4, and the probability of "Excellent" is 26.2, when there is no data about the learner.
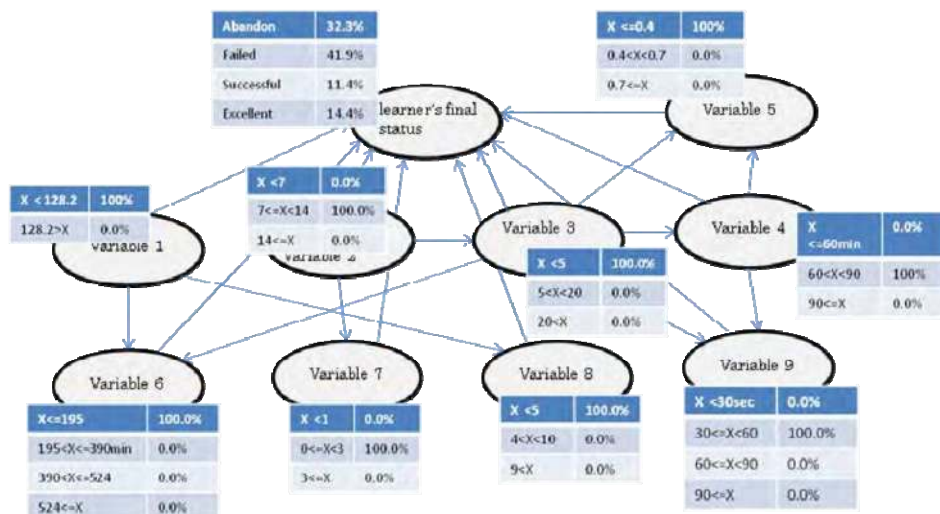
5.3 Bayesian agent

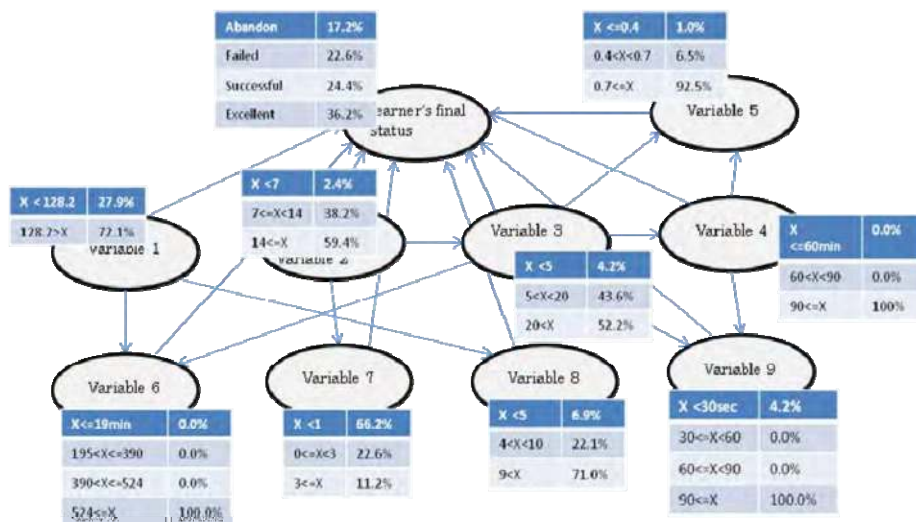Fig. 6. Final status probabilities propagated by learner's current-history data

Fig. 7. Retrieved explanatory variables that change predicted final status to "excellent"

This section explains how Bayesian networks are learned from learning-history data. Fifteen Bayesian-network structures are estimated corresponding to data from learners' learning histories for the fifteen weeks because all courses run for 15 weeks.

Bayesian networks also suffer from the cold-start problem and no inferences can be drawn when we provide new courses. To solve this problem, this paper uses the prior distribution in (6) for learning a Bayesian network. In detail, a Bayesian-network structure is first estimated using data from all learning histories stored in the database; this does not include

data corresponding to the target course. Here, this data from learning histories is called "prior data". The main idea is that the estimated structure from the prior data is used for the prior hypothetical structure, B_S^h, in (6). Next, based on this estimated prior hypothetical structure B_S^h, the Bayesian network is learned by maximizing the BDe in (5) from learning-history data corresponding to the target course. This proposed method enables us to solve the cold-start problem when we start a new course. When there are not sufficiently large amounts of data from the learning histories for the target course, the Bayesian agent follows the estimated Bayesian-network structure from all learning-history data, i.e., the prior data. When there are sufficiently large amounts of data for the target course, the Bayesian agent follows the estimated Bayesian-network structure for the target course. The ESS value, α, means the pseudo-sample size reflecting the prior data and this has been determined as 100.0 in this research.

In addition, we employ the Bayesian-network classifer model since the target variable is one variable (Friedman, Geiger, & Goldszmidt. 1997). In detail, we first add arcs between the final status node and all the explanatory variables, and then construct the network structure between the explanatory variables to maximize BDe given the previously drawn arcs. Here, the greedy search algorithm is employed to learn the network structure.

Figure 3 shows the Bayesian network estimated by maximizing BDe in (5) from prior data (data from 4,344 learners in 64 courses). The network in Figure 6 was propagated using the history data from a learner's fourth week of learning. Furthermore, the probabilities of the variables corresponding to the nodes in Figure 3 indicate the prior-belief probabilities for the categories. For example, the node corresponding to the predicted final status of a learner indicates that the probability for "abandoned" is 32.3%, the probability for "failed" is 41.9%, the probability for "successful" is 11.2%, and the probability for "excellent" is 14.4%, when there are no data about the learner.

5.3 Bayesian agent

The main purpose of the intelligent agent system is to provide optimum instructional messages to a learner using the previous automatically constructed learner model. The agent appears in the LMS as shown in Figure 4. The agent system also performs various actions based on the learner's current status, as shown in Figure 5. The instructional messages given to a learner are generated as follows:

The agent obtains the learner's current learning-history data and predicts his/her final status using the propagated probabilities in Figure 6. If the predicted most likely final status is "excellent", then the agent provides messages like "Looking great!", "Keep doing your best", and "Your probability of success is xx%". If the predicted status is not "excellent", the agent searches for the explanatory variable that will most increase the probability for the predicted final status by changing the value. Next, the agent finds the explanatory variable that most increases the probability for the predicted final status by changing the value, given the changed value of the explanatory variable. Thus, the agent retrieves the explanatory variables in order for the values to increase the final status probability by changing their values until the predicted most likely final status is "excellent". The retrieved explanatory variables that change the predicted final status to "excellent" corresponding to the learner in

Variables	Instructional messages
1. The number of topics the learner has learned.	1. You are behind in progress in the lesson. Please attend more lectures. 2. Your progress in the lesson is liable to slow. Let's attend more lectures.
2. The number of times the learner has accessed the e-Learning system.	3. You have not participated enough in the lesson. Let's access the system more often.
3. The average number of times the learner has completed each topic.	4. Don't forget previously learned content! Let's review the previous content again.
The average time for learning each lecture, which consists of several types of content and runs for 90 minutes.	5. It seems that you are working through the lectures too quickly. Please spend more time on each lecture.
5. The average degree of understanding of each topic (This is measured by responses to questions that corresponds to each topic.).	6. Was the content of the lesson too difficult? Let's repeat the lecture from the beginning. 7. When there is something you don't understand, let's post questions on the discussion board.
6. The average learning time for each course consisting of fifteen lectures.	8. You have not participated enough in the lesson. Let's access the system and study the content more slowly and carefully.
7. The average number of times the learner has changed answers to e-Learning questions.	9. Your knowledge does not appear to be adequate. Let's repeat the lecture from the beginning.
8. The number of times the learner has posted opinions or comments on the discussion board.	10. Learning is more effective when there is interaction between learners. Let's participate in and contribute to the discussion board.
9. The average learning time for each topic.	11. Did you pay sufficient attention to the lecture? Ordinarily, a lesson should take more time to complete.

Table 1. Instructional messages corresponding to detected variables

NC	BN	DT	SVM	Naïve Bayes
2	89.11(84.27)	75.00(88.70)	80.75(89.25)	75.50(76.25)
3	86.47(66.92)	80.00(84.75)	81.00(88.7)	76.00(77.25)
4	94.00(65.00)	82.00(88.75)	74.00(91.5)	77.00(77.75)
5	96.63(60.57)	80.25(84.75)	78.76(91.5)	76.75(77.75)

Note NC: number of categories; BN: Bayesian network, DT: decision tree model using the ID3 algorithm. The parenthetical values indicate the fitting rates for the training data.
network was the most accurate for all NCs.
Table 2. Correct prediction rates (%) obtained in cross-validation experiment

Figure 6 are Variables 4, 6, and 9 as shown in Figure 7. The agent provides messages with the predicted future status, the probability of success estimated by the Bayesian network, and the instructional messages according to Table 1. That is, the agent generates adaptive messages from the gap between the learner's history data and the past-history data of excellent learners.

6. Comparative predictive experiments

Some previous studies have been done on predicting a learner's final test score using several machine-learning methods from learning-history data in e-Learning. Minaei-Bidgoli, Kashy, Kortemeyer, and Punch (2003) compared the accuracy of machine-learning methods (decision-tree model, naive Bayes, and SVM) to predict a learner's final test score from the learning-history data in e-Learning. The decision tree performed the best in the results. However, Talavera and Gaudioso (2004) and Hamalainen et al. (2006) conducted similar experiments and insisted that naive Bayes was the best model.

Finally, Huang et al. (2007) claimed that SVM was the most effective model. Thus, as these previous studies reported different results, this means that the accuracy of prediction depends on the characteristics of the data (i.e., the kinds of variables, data size, domain, and learners' age). Therefore, we also needed to evaluate various models with respect to data obtained from the LMS "Samurai" just as the previous studies had done. We compared the Bayesian-network model with the decision-tree model (ID3), naive-Bayes model, and SVM. Here, we employed the most popular naive-Bayes model, the "multivariate Bernoulli model" (Domingos & Pazzani 1997) and a well known SVM that has a "polynomial kernel" (Vapnik, 2000).

First, the latest data from 800 learners were randomly sampled from the learning-history database for 128 courses in the LMS "Samurai". Furthermore, the learner-history data from 400 out of the 800 learners were randomly sampled as training data, and the remaining 400 learner-history data were used as validation data (test data) for a cross-validation experiment. The cross-validation experiment was carried out to predict learners' final statuses from their learning-history data. The decision-tree (ID3) and naive-Bayes models only use category variables as input data, but the learning-history data use data on continuous variables. Consequently, the data on continuous variables in the learning-history data were categorized as uniformly distributed in each category. Although SVM can use the data on continuous variables for input data, this experiment applied the categorized data to SVM under the same conditions as those for the other models. Here SVM employed the polynomial kernel as a kernel function. To categorize the input data, the range (from the minimum to the maximum value of data) of each variable was divided by the number of categories m into the category ranges. As a result, the continuous data were transformed to category data x_{icj} (if the i-th variable's category $c's$ range includes j-th learner's data then x_{icj} =1, otherwise x_{icj} =0),(i=1,…9, c=1,…,m, j=1,…,N). The number of categories for all variables was increased from two to five in the experiment. In addition, the learninng Bayesian networks in this experiment employed a uniform prior belief for BDe.

The results are listed in Table 2. Each value indicates the correct prediction rates for cross-validation given the number of categories in the corresponding model.

For a large number of categories, DT was very accurate, but not for a small number of categories. For a small number of categories, SVM was very accurate. However, it is clear that SVM overfits the data when there are four or more categories. However, although the decision-

tree model is less accurate than SVM when there are fewer categories, it has the best accuracy with four or more categories.

Naïve Bayes has lower correct prediction rates, which can be explained by the variables all having a mutually strong correlation; nevertheless, the model assumed the variables were conditionally independent. The Bayesian network shows the best performances for all NCs. These results indicate the Bayesian network is the most suitable for data stored in LMS "Samurai" because the proposed agent needs to use four categories as variables.

7. Evaluation of prior belief in BDe

One of unique features of the proposal method is to learn a Bayesian agent from learning-history data using BDe that reflects prior belief previously learned from the prior data. Here, it should be noted that the learned prior belief might be quite different from the true structure since the prior data do not necessarily reflect the characteristics of the course. However, no research has been done on how prior belief (where we employ an incorrect hypothetical structure) affects the learning efficiency of Bayesian networks.

Next, let us consider some simulation experiments using the network structure in Figure 3. The procedure in the simulations involves three steps:

1. 100,000 samples are generated from Figure 3.
2. Using MDL, *BDe* with the hypothetical Bayesian-network structures (all possible structures), and Bdeu (BDe with a uniform prior distribution), Bayesian network structures are estimated based on 500, 1,000, and 10,000 samples, respectively, from the datasets for the structures shown in Figure 5. The search method employs the greedy search method.
3. The average number of missing or extra arcs (mean error: ME) was calculated by repeating Procedure 2 ten times.

Table 3 lists the mean errors (ME: the average number of missing or extra arcs in 10 estimates) for BDeu, MDL, and *BDe* for sample sizes of n = 500, 1,000, 10,000, and 100,000. The column "+" indicates the average number of extra arcs in the estimated structures and the column "-" indicates the average number of missing or extra arcs in the estimated structures. The column "ME" indicates the average number of missing or extra arcs in 10 estimates.

The column "*Bde* (best)" indicates the best results by changing the hypothetical structure given the true structure. In contrast, the column "BDe (worst)" indicates the best results by changing the equivalent sample size given a hypothetical prior structure that is most different from the true one. The results for "*Bde* (best)" overwhelmingly have the best accuracies for small sample sizes.

N	MDL			Bdeu			Bde (Best)			Bde (Worst)		
	+	-	ME	+	-	ME	+	-	ME	+	-	ME
500	7.2	8.1	11.3	6.7	2.4	9.1	6.4	2.0	8.4	6,3	3.7	9.9
1,000	5.1	6.7	7.9	4.2	1.3	5.5	4.4	1.7	6.1	4.4	2.2	6.6
10,000	2.0	0.2	2.2	2.0	0.2	2.2	1.8	0.4	2.2	1.8	0.4	2.2
100,000	1.2	0.0	1.2	1.2	0.0	1.2	1.2	0.0	1.2	1.2	0.0	1.2

Table 3. Comparisons of estimates for various score metrics for Bayesian networks

	With agent system	Without agent system
Subject name	Information & Communication Technology	Information & Communication Technology
Students	Undergraduate students (third and fourth years)	Undergraduate students (third and fourth years)
Learning location	Individual student homes	Individual student homes
Credits	2	2
Number of students	74	92
Term	2003, April 10 - July 31	2004, April 10 - July 31
Number of students who withdrew from course	14 (18.9%)	49 (53.2%)
Final test scores	Average: 93.26 Variance: 43.2 (n=60)	Average: 78.74 Variance: 215.24 (n=43)
P-value	1.33E-07	
Total learning time (minutes)	Average: 1045.13 Variance: 71721.8 (n=60)	Average: 801.88 Variance: 65426.9 (n=43)
P-value	1.25E-05	
Average degree of progress per lesson	Average: 0.93 Variance: 0.64 (n=60)	Average: 0.84 Variance: 2.03 (n=43)
P-value for statistical difference test of two averages	0.00031	
Total number of contributions to discussion board	714	928

Table 4. Comparison between classes with and without system

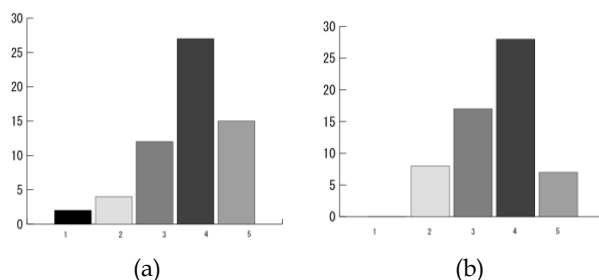
(a)   (b)

Fig. 8. Plotted results for Question **A** given to (a) class with system and (b) one without

This means that our valid prior knowledge about the network structure facilitates more efficient learning of Bayesian networks. In addition, even if we set prior knowledge
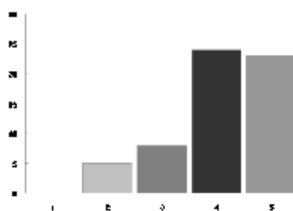
Fig. 9. Results for Question **B**

incorrectly, the results for BDe have better or the same accuracy than one of the traditional MDLs or BDeus. Consequently, the results reveal that the proposed method is effective even if prior belief is quite different from that of the true structure.

8. Evaluation of agent system

The system was evaluated by comparing a class of students that used the agent system with one that did not use it for one semester. The Bayesian network for the agent system was learned using 1,344 histories of learners. The details on the two e-Learning classes are summarized in Table 4. The results reveal that far fewer students withdrew from the class if they had used the LMS with the agent system. In addition, the final test scores, learning-time data, and progress with learning data also indicate that the proposed agent system enhanced learning significantly.

The presentation of the predictive future status of learners and the presentation of adaptive instructional messages help them to maintain the required learning pace. As a result, they can progress until they reach their predicted future status. Furthermore, all learners were asked Question **A: "**How would you rate the system's ability to enhance your e-Learning? 1. Very poor, 2. Poor, 3. Fair, 4. Good, or 5. Very good."

The group with the agent system was asked an additional question, Question **B:** "How would you rate the adequacy of the instructional messages from the agent system? 1. Very poor, 2. Poor, 3. Fair, 4. Good, or 5. Very good." The results for Question **A** are given in Figure 8. The response frequencies for answers 2 and 3, "poor" and "fair" were lower for the class with the system than that without it. This indicates that the system was effective in enhancing learning and the instructional messages had a positive effect on e-Learning. However, it should be noted that the response frequency for "very poor" increased for the class with the system. If we assume that the difference between the results for the two classes are due only to using the agent system, the results mean that learners' opinions about the agent system tended to be polarized compared to the opinions by the class without it.

Figure 9 summarizes the frequency of learners' responses to Question **B**. The results indicate that many learners rated the agent system's messages as "good" or "very good" and this means that the instructional messages from the agent system are acceptable for many learners. However, it should be noted that five learners rated it as "poor".The learners who rated the system as "poor" gave the following reasons :

·   "The messages from the agent were too distracting. I couldn't concentrate on my learning due to the agent's incessant actions."

- "The messages from the agent were interfering with my learning because I knew almost all the message content previously even if the agent hadn't sent it."

This means that the messages from the system interfered with some autonomous learners who could learn by themselves. Therefore, we think that the system needs a function whereby learners can hide the agent from the system whenever they need to.

9. References

Becker, K. & Vanzin, M. (2003). Discovering increasing usage patterns in web based learning environments, *Proc. Int. Conf. on Utility, Usability, and Complexity of e-information systems*, pp. 57–72

Domingos, P. & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss, *Machine Learning*, Vol. 29, 103–137, ISSN:0885-6125

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifier, *Machine Learning*, Vol. 29, 131–161, ISSN: 0885-6125

Gabrielle, D. M. (2000). The effects of technology–mediated instructional strategies on motivation, performance, and self directed learning, *Proc. of ED-Media*, pp. 2569–2575, ISBN: 1-880094-40-7

Gamoran, A. Secada, W. G. & Marrett, C. A. (2000). The organizational context of teaching and learning: changing theoretical perspectives, In: *Handbook of Sociology of Education*, Hallinan, M.T. (Ed.), 37–64, Kluwer Academic/Plenum Publishers, ISBN: 0387325174 , New York

Hamalainen, W., Laine, T. H., & Sutinen, E. (2006). Data mining in personalizing distance education courses, In: *Data Mining in e-Learning*, (Eds.) Romero, C. & Ventura, S., 157–171, WIT Press, ISBN: 1-84564-152-3, New York

Huang, C. J., Chu, S. S. & Guan, C. T. (2007). Implementation and performance evaluation of parameter improvement mechanisms for intelligent e-learning systems, *Computers & Education*, Vol. 49, Issue 3, 597–614, ISSN: 0360-1315

Knowles, M. (1983). How the media can make it or bust it in education, *Media and Adult Learning*, Vol. 5, No. 2, 3–4, ISSN: 09152202

Knowles, M., Holton, E. F. & Swanson, R. A. (1998). *The Adult Learner: The Definitive Classic in Adult Education and Human Resource Development* (5th ed.). Gulf Publishing, ISBN: 0884151158, Houston

Mayer. R. E. and Anderson, R. B (1991). Animations need narrations, *Journal of Educational Psychology*, Vol. 83, No. 4. 484–490, ISSN: 00220663

Merriam, S. B. & Brockett, R. G. (2007). *The Profession and Practice of Adult Education: An Introduction*, Jossey-Bass, ISBN: 0470181532, New York

Minaei-Bidgoli, B., Kashy, D.A., Kortemeyer, G., and Punch, W. F. (2003). Predicting student performance: an application of data mining methods with an educational web-based system, *Proc. IEEE conf. on Frontier in Education*, Vol. 1, pp. 13–18

Quinlan., J. R. (1986) Induction of decision trees. *Machine Learning*, Vol. 1, No. 1, 81-106, ISSN: 0885-6125

Talavera, L. & Gaudioso, E. (2004) Mining student data to characterize similar behavior groups in unstructured collaboration spaces, *Proc. of the Workshop on Artificial Intelligence in CSCL*, *16th European Conference on Artificial Intelligence, ECAI2004*, pp. 17–23

Ueno, M. (2004) Learning Log Database and Data Mining system for e-Learning, *Proc. International Conference on Advanced Learning Technology,* pp. 194–201 ,ISBN 0-7695-2181-9, IEEE Computer Society

Ueno, M. (2005) Intelligent LMS with an agent that learns from log data, *Proc. of e-Learn2005*, pp. 3169–3176

Ueno, M. (2007). *e-Leanirng in knowledge society* (in Japanese), Baihuukan, ISBN: 4563015733, Tokyo

Vapnik, V. (2000). *The Nature of Statistical Learning Theory*, Springer , ISBN: 0-387-98780-0, New York

Visser, L. & Keller, J. M. (1990). The clinical use of motivational messages: An inquiry into the validity of the ARCS model to motivational design., *Instructional Science***,** Vol. 19**,** 467–50, ISSN: 0020-4277

Visser**,** L., Plomp, T., & Kuiper, L. (1999) Development research applied to improve motivation in distance education, *Association for Educational Communications and Technology***,** Houston, 17–28

von Glasersfeld, E. (1995). A constructivist approach to teaching. In: *Constructivism in Education*, Steffe, L & Gale, J. (Eds.), 3–16, Lawrence Erlbaum Associates, Inc., ISBN: 080581096X, New Jersey.

E-learning Experiences and Future

Edited by Safeeullah Soomro

This book is consisting of 24 chapters which are focusing on the basic and applied research regarding e-learning systems. Authors made efforts to provide theoretical as well as practical approaches to solve open problems through their elite research work. This book increases knowledge in the following topics such as e-learning, e-Government, Data mining in e-learning based systems, LMS systems, security in e-learning based systems, surveys regarding teachers to use e-learning systems, analysis of intelligent agents using e-learning, assessment methods for e-learning and barriers to use of effective e-learning systems in education. Basically this book is an open platform for creative discussion for future e-learning based systems which are essential to understand for the students, researchers, academic personals and industry related people to enhance their capabilities to capture new ideas and provides valuable solution to an international community.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Maomi Ueno (2010). Bayesian Agent in E-learning, E-learning Experiences and Future, Safeeullah Soomro (Ed.), ISBN: 978-953-307-092-6, InTech, Available from: http://www.intechopen.com/books/e-learning-experiences-and-future/bayesian-agent-in-e-learning

INTECH
open science | open minds