

Bayesian Network Based Intelligent Advice Generation for Self-Instructional e-Learner

P. Kuila, C. Basak, and S. Roy

Abstract—This paper presents an agent based Intelligent Tutoring System (ITS). The agent plays the role of an advisor for an e-learner to facilitate the learner to achieve his learning objective. It provides advices to assist an e-learner while solving problems that are normally provided by human experts. A Bayesian network is employed to construct the possible solution states of a problem with the statistics of mistakes; those can be committed by a learner in their problem solving process. The agent will collect statistical data from Bayesian network on learner's mistakes and the way in which one learner may commit a mistake. On the basis of the statistical data on the learner's behaviour, especially with respect to his tendency to commit an error, the agent anticipates the point of difficulty during a problem solving session and accordingly guides the learners to solve the given problem free of errors. The Bayesian network is trained with training data, prior information (e.g., expert knowledge, casual relationships, and estimated graph topology or network structure) and the parameters of the joint probability distribution.

Index Terms—Intelligent advisory system (ias), intelligent agent, Bayesian network.

I. INTRODUCTION

An enormous development in the field of information technology (IT) has opened up some e-systems parallel with traditional ongoing systems such as e-banking systems parallel with banking system, e-business, e-education, e-learning etc. This development has also opened up immense possibilities in the contemporary education and training arena. The traditional classroom based teaching learning systems are augmented by various kinds of e-learning systems e.g. Intelligent Tutoring Systems (ITSs), Learning Management Systems (LMS), and Virtual Laboratory etc. These all software systems can significantly improve their performance if they could adapt to the emotional state of the user, for example if Intelligent Tutoring Systems, ATM's, Ticketing machines could recognise when users were confused, frustrated, angry or whether the learner in ITSs understood the learning materials or not then they could guide the user back to remedial help systems so improving the service. A tremendous amount of research activities are currently going on in developing these software systems. Many researchers now feel strongly that ITSs would be significantly enhanced if computers could adapt to the emotions of students [1]. This idea has spawned

the developing field of Affective Tutoring Systems (ATSs): ATSs are ITSs that are able to adapt to the affective state of students [2]. The term "affective tutoring system" can be traced back as far as Rosalind Picard's book Affective Computing in 1997. In traditional tutoring system student advising is a very important issue. Here, we can estimate the student state, understanding level and according to these we can advice them. However student advising is not given enough attention in Intelligent Tutoring Systems (ITS) [3], [4], [5].

Current researchers in the field of ITSs are at investigation of how to make a computer- based tutors more flexible, autonomous and adaptive to evaluate each student. In basic ITS one student is evaluated by some multiple choice questions. Here the main drawback is, when one student makes some wrong answer, there is no chance of understanding the proper step, where the student has committed the mistake while solving the problem. Usually, solution of a problem consists of a sequence of steps. A learner may commit a mistake at any step during the solution process. So, if we can identify the proper step, where a student has been committed a mistake then we can advice them according to their mistakes.

Student modeling is crucial for an intelligent learning environment to advice a student according to their knowledge and understanding ability. The basic underlying perspective is to consider every student as being unique and advice him according their knowledge, understanding and types of mistake while solving a problem. This paradigm can be built by creation of a user or student model [6]. The main objective of student model is to understand the knowledge level of individual student, how they learn and what their problems are while they learn [7]. Assessing the user state of knowledge and profile requires uncertainty reasoning. Artificial Intelligence has addressed this problem in various ways such as fuzzy logic, Bayesian Networks [8] etc. So, using student model ITS can build an advisory system to advice a student students in their learning process by maintaining an accurate model of a student's understanding level, current knowledge state which allows more intelligent pedagogical decisions and actions to be happened.

Using student modeling and the techniques of artificial intelligence we can build an advisory system to advice a student in their learning process. An advisory system provides the advices and assists for solving problems that are normally provided by human experts. They can be classified as a type of expert systems [9], [10]. Advisory system has the power to make recommendations but not to enforce a decision maker and it does not make decisions but rather help guide the decision maker in the decision-making process.

Manuscript received September 15, 2011; revised October 28, 2011

The authors are with the department of Computer Science and Engineering from NITTTR' Kolkata.india (e-mail: pratyay_kuila@yahoo.com.)

While the system is leaving the final decision, it makes authority up to the human user [11]. The decision maker works in collaboration with advisory system to solve a problem. The three main processes in advisory systems are knowledge acquisition, cognition, and interface [12]. In an experiment it is proven that continuous feedback to a learner allows him to achieve correct answers by reducing their errors [13].

In this paper, we propose an Intelligent Advisory System (IAS) to assist a student in their problem solving process. We use Bayesian network and intelligent agent to construct the system. Advisor agent and Expert agent are the two intelligent agents, which are used in our propose system. The Advisor agent will act on behalf of the human advisor and advice the learner at appropriate time. There is another intelligent agent, which is named Expert agent. The Expert agent determines the student module and selects the problem according to learner's level of understanding.

Rest of the paper is organized as follows. Section II presents an overview of Intelligent Tutoring System, Bayesian network and Intelligent Agent. Section III discusses the proposed system as a combination of two intelligent agents. These two intelligent agents are named as Advisor agent and Expert agent. This section also discusses how the Bayesian network is learning. Section IV presents the architecture of the developed system as well as the user interface of the developed system. The last section concludes the paper.

II. PRELIMINARIES

Intelligent Advisory System is a hybrid expert system which incorporates potentials of Rule Based Reasoning and Case Based reasoning, which are majorly two techniques used in expert systems. The integration of these two techniques will provide a powerful system which can be used for complex decision making process. An advisory system has the power to make recommendations but not to enforce them. In the e-learning environment an advisory system is used to advice a learner to assist him attains his learning objectives. Using artificial intelligence techniques it is possible for the advisory system to trace a learner's steps to solve a problem, his skill, motive and so on. The proposed advisory system is based on Bayesian network [14][15] as its structural framework to build an intelligent agent whose goal is to provide useful advices to the e-learner. In our proposed intelligent tutoring system, an intelligent agent will take actions autonomously when it need to advice a learner during their learning process.

A. Intelligent Tutoring Systems:

An intelligent tutoring system (ITS) is educational software containing an artificial intelligence component. The software tracks students' work, tailoring feedback and hints along the way. By collecting information on a particular student's performance, the software can make inferences about strengths and weaknesses, and can suggest additional work [3][4][6]. ITS is an attempts to produce a computer behaviour, which if performed by a human. ITSs are so-called for their 'intelligent' ability to adapt the needs of individual students, by being able to adapt the knowledge,

learning abilities and needs of each individual student. Traditionally, ITS research has assumed that students are modeled according to their answers to questions. These models represent different aspects of the student's cognitive state. ITS applies artificial intelligence techniques to computer-assisted instruction allowing to be developed with individualised instruction.

Fig. 1 shows the major components of an intelligent tutoring system. An ITS is traditionally said to comprise of four interdependent components: the student model, the pedagogical module, domain knowledge and the interface. The student model stores information specific to individual learners, upon which the pedagogical module devises appropriate teaching strategies for the ITS to employ. These strategies are applied to the domain knowledge, generating a subset of knowledge to be presented to the learner using the communication model, which acts as an interface between the learner and the ITS. As the learner responds to the system, the student model is updated and the cycle repeats.

1. Domain Knowledge Base Module
2. Student Model
3. Pedagogical Module
4. Interface
5. Learners

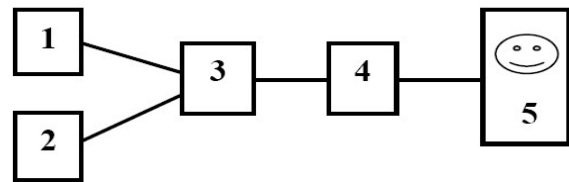


Fig. 1. Major Components of an ITS

B. Bayesian Network:

A Bayesian network (BN) is a graphical model that encodes probabilistic relationships among variables. It is a directed acyclic graph (DAG) [16], where, every child node depends on its parent node. Each node represents some state with some variable. Each variable A with parent B_1, B_2, \dots, B_n has a conditional probability $P(A|B_1, B_2, \dots, B_n)$ associated with it. If the variable A has no parents then the probability is unconditional $P(A)$. Fig. 1 presents an example of a Bayesian networks.

In the Fig.2 the conditional probability of node C is depend on its parents, that are A and B . The conditional probability of node C is $P(C/A,B)$.

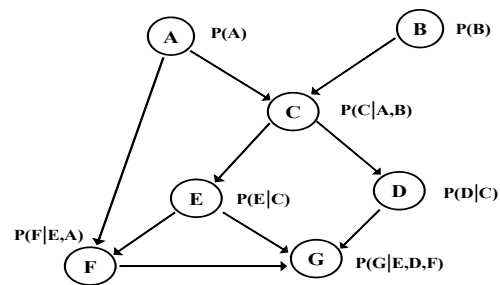


Fig. 2. Bayesian Network

So, the parameters used to represent the uncertainty are the

conditional probabilities of each node, given the combination of states of its parents; that is, if $\{X_i, i=1,2,3,\dots,n\}$ are the variables of the network and $pa(X_i)$ represent the set of parents of X_i for each $i=1, 2, 3, \dots, n$ that is, the set of discrete conditional probability distributions of each variable given its parents. The probability defines the joint probability distribution for the entire network as,

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i / pa(X_i)) \quad (1)$$

As example, we can apply the equation (1) in the following figure. From figure (a) we can say that,

$$P(x_1, x_2, x_3) = P(x_3/x_1).P(x_2/x_1).P(x_1) \quad (a)$$

In same way from figure (b), (c) and (d) we can found these equations accordingly.

$$P(x_1, x_2, x_3) = P(x_3/x_2, x_1).P(x_2).P(x_1) \quad (b)$$

$$P(x_1, x_2, x_3) = P(x_3/x_1).P(x_1/x_2).P(x_2) \quad (c)$$

$$P(x_1, x_2, x_3) = P(x_2/x_3, x_1).P(x_3/x_1).P(x_1) \quad (d)$$

So, from Fig. 2 we can see that,

$$P(A, B, C, D, E, F, G) = P(A).P(B).P(C/A, B).P(D/C).P(E/C).P(F/E, A).P(G/E, D, F)$$

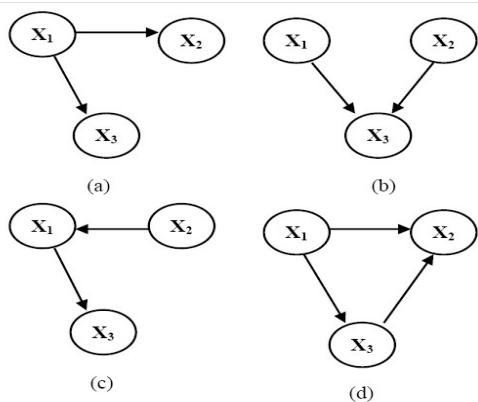


Fig.3. Bayes Rule on arc reversal

Thus, to define a BN, we have to specify:

- The set of variables X_1, X_2, \dots, X_n .
- The set of links (arcs) between those variables. These arcs represent a dependency between variables. The network formed with these variables and arcs must be a Directed Acyclic Graph (DAG).
- For each variable X_i , its probability is conditioned to its parents, that is, $P(X_i/pa(X_i)), i = 1, 2, \dots, N$.

If we are using BNs in ITS then the variables can represent different things depending on the domain. The variables can be rules, concepts, problems, abilities, skills, etc. These variables are linked by relationships between them. Once the links and variables have been defined, the conditional probabilities must be specified. Another important issue of BN is that when we are using BN to

defining a student model then it should be learned. This problem is known as Bayesian Learning Problem [17] [18].

C. Intelligent Agent:

Intelligent agent is a fast emerging technology and has wide range of applications. Although there are several tools for agent development, there is few design tool to assist the conversion from paper based agent mental state design to effective representation of them in abstract data structures which can be used by the agent management system to create intelligent software agents [19], [20], [21].

An agent is an entity that perceives the environment and takes actions autonomously to change the environment to reach the desired environmental state [22]. In artificial intelligence, an intelligent agent (IA) is an autonomous entity which observes and acts upon an environment to carry out some task and apply some degree of intelligence to the task. Here, intelligence means the capacity of learning from past experience, reasoning, understanding and have knowledge of an event and about the circumstances. So, an intelligent agent can learn or use knowledge to achieve their goals as example learning intelligent agent [22]. Some intelligent agents can also interact with one another. The intelligent agent may be very simple or very complex depending on their design.

There are many definitions of intelligent agents found in the literature. The definition, "intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in doing so, employ some knowledge or representation of the user's goals or desires," by Maes [23], [24] is the most appropriate for our purposes.

In software system we can apply the concept of the intelligent agent, and then it will be called as an intelligent software agent. Smith *et al.* (1994) defines intelligent software agent as "a persistent software entity dedicated to a specific purpose"; Janca (1995) defines software agent as "a software entity to which tasks can be delegated". An Intelligent Software Agent is a computational software system which has goals, sensors, and effectors, and decides autonomously which actions to take, and also when to take.

In our system, we have used intelligent software agent to trace, advice and also to measure the level of understanding of each learners. When, the system is running or a learner is using this system, then it is very necessary to maintain all those things i.e., tracing the learner's problem solving process and advice them according to their committed mistakes autonomously parallel with the running system. Because of we have introduced two intelligent agents in our system.

III. PROPOSED SYSTEM

The main approach of our system is that, when a learner is going to solve a problem, then the system has a chance to predict his behaviour through the statistics of the previous learner's problem solving process. The key assumption of our system is that each types of problem can be converted to a pre assumed problem solving steps as Fig. 4.

Basically in Intelligent Tutoring Systems, a learner is evaluated by some mathematical problems. To evaluate a

learner, inspecting a descriptive theoretical problem is more complex than checking a mathematical problem.

Let we are going to solve a mathematical problem on Newton's Laws of Motions. For a mathematical problem on Newton's laws of motions, directly or indirectly there are given some values of the data like, distance, initial velocity, final velocity, speed, acceleration, time etc. Then, a learner has to find out the values of some data, which are not given in that problem. In this case, the learner has to use some equations on Newton's Laws of Motions to solve this problem. All possible necessary equations related to the Newton's law had already been stored in our system. To solve this problem, at first the learner has to carefully choose the given data and also the required data. Then he has to select the necessary equations to manipulate the required data. Next step is the manipulation step to calculate the required data.

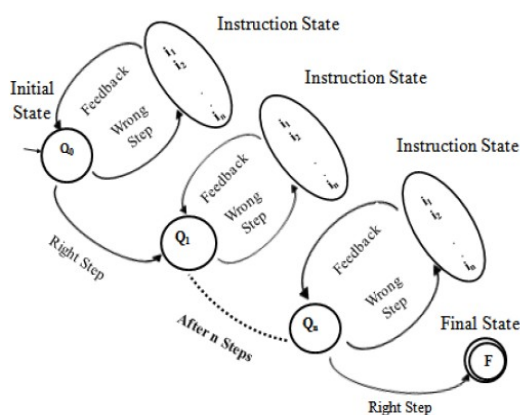


Fig. 4. State wise model for solving a problem

The system has all the necessary information and data about the particular problem, which is going to be solved by a learner. Even, the system has level of difficulty of that problem, which is judged by some expert in practical experiment and also the previous statistics about that particular problem, when it was solved by previous learners. Here, the previous statistics means the particular step, where the learners had committed mostly mistakes, the percentage of learners, who able to solve this problem etc.

Now, when a new learner is going to solve this problem the system can trace him in each problem solving steps and can advice the learner according to their committed mistakes. So, we can say that our system will not only drive a learner at proper route, it also behaves as a friend, as a guide such that a learner never thought him alone in front of a computer.

In our system, we have used intelligent agent to maintain these whole tasks. Fig. 4 shows the possible behavior of the proposed intelligent agent as a state instruction diagram. When one learner is going to solve a problem he will have to go through some steps as mentioned. When a learner is in initial state, then system has a model of all next steps including final step and also statistics of previous learners, who had done some wrong steps or right steps. So, system has the information regarding probabilities of committing a mistake at various steps. According to this information, system can advice the learner at appropriate time and prevent him from committing a mistake.

When the system is used very first time then system has not nay information about previous learners. So, in this case we will impose some information, which are taken by some practical experiment.

In every problem solving step, a learner can commit a mistake or he can work out it in proper way. So, after every problem solving step a learner may move along with either the wrong step or the right step. If a learner takes right step then he will reach to the next state and after few right step learners will reach to final state, which means the problem is solved. If the learner commits any mistake then learner will reach instruction state (Fig .5). An instruction state is a combination of some sub states, which are the pre assumed possible states. These pre assumed states are indicating the possible mistakes that can be committed by a learner exact before this instruction state. According to these corresponding possible mistakes there are some predefined instructions or advice for the learners. Let a learner has committed a mistake which is not already defined in the instruction state. In this case the system will return the learner in previous problem solving state without giving any advice. In addition the system will automatically update itself by creating a new sub state of this instruction state. Now, for the next learners this sub state will act and give advice if need.

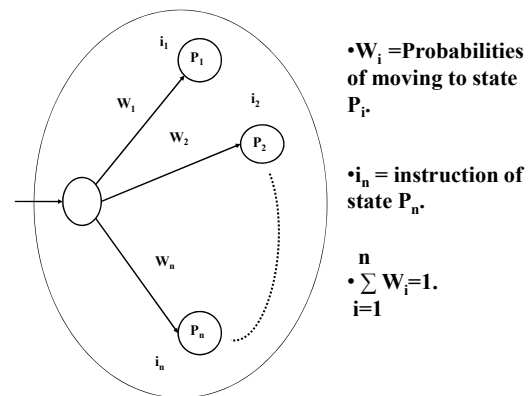


Fig. 5. The model of Instruction State

Here intelligent agent has the statistics of previous mistakes, which were committed by other learners. The intelligent agents will guide a learner before committing a mistake in a particular step, according to higher frequency of mistakes that had been committed previously in that step by the previous learners.

A. The Advisor Agent

In our proposed system, few numbers of problems are stored a priori. Each problem has their corresponding Bayesian network, which represent the step wise possible solution state of that problem, as well as the possible mistakes, which may be committed by a learner. Advisor agent will collect information from this Bayesian network and advices the learners according to their response in a particular problem. Performance of the system depends heavily upon its feedback in proper time and the way how it advices. Here, proper time refers to when the learner is giving a response to the solution, when the feedback is presented to the learner should be governed by what the learner has done. Tutors are better than teachers in this respect they can provide a learner

with timely feedback better than most teachers.

The advisor agent will collect information from Bayesian network about previous learner's mistakes and the way in which one learner can commits some mistakes. As this network is updating according to the use of this system and the way a learner can solve the problem and commit mistakes.

However, the initial practical settings the Bayesian network is unknown and one needs to learn it from the data. This problem is known as the *Bayesian network (BN) learning problem* [15], [17]. We can solve this problem by given practical training data and prior information (e.g., expert knowledge, casual relationships, estimated graph topology or network structure) and the parameters of the joint probability distribution in the Bayesian network.

B. The Expert Agent

This agent determines the learner's performance, learner's level of understanding and learner knowledge of concept from the given problem statement, the problem specification and learner's feedback. Suppose, after giving an easy level problem to the learner, the learner is unable to find out the given necessary data and required data from the specified problem. In this case the system will assume that the learner should study the learning content once again. So, the system will refer the learner to the study portion. When the learner reaches a certain score above the current level of difficulty, for example 80 percent or more, the expert agent increase the level of difficulty of the problem to be given for the learner.

IV. THE SYSTEM DEVELOPED

In our proposed system, the Bayesian network model is very important part of this system. The performance and response of this system is heavily depends on the BN model. To work with a system, users have to be able to control and assess the state of the system. The user interface of the system is the whole composed of the instruments (i.e., button, textbox, label, text field etc.), the user can use to accomplish the tasks of the system and maintaining the system. User interface is another important part of such software system, which are directly interacted buy end users.

A. System Architecture

We keep track of learner's knowledge of concept using the number of mistakes they are doing to solve a problem. When they commit a few mistakes then expert agent increases the level of difficulty of the problem for the learner. The expert agent can take information from student model (Fig. 6.) and BN to increase the difficulty level of the problem.

Fig. 6 presents the architecture of the proposed system. In the developed system, when one new learner login, his record is stored in student module. The main focus of this module is to consider every student as being unique and takes record of individual student, how they learn and what their problems are while they learn. While the learner solving the problem, the Bayesian network learns concurrent with the solving process. When the learner commit any mistake while solve the problem, the advisor agent advises the learner according to their mistake. The Bayesian network has the information

regarding probabilities of committing a mistake at various steps. According to this information, system can advice the learner at appropriate time and prevent him from committing a mistake.

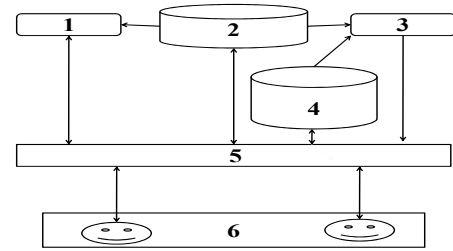


Fig. 6. System Architecture

1. Advisor Agent
2. Previous Information Through BN
3. Expert Agent
4. Learner's Knowledge Base Information or student model
5. User Interface
6. Learner

B. Bayesian network formation

For each problem, which is previously selected in our system has a corresponding Bayesian Network. For any particular problem the corresponding Bayesian network can be constructed as Fig. 6 has been shown bellow.

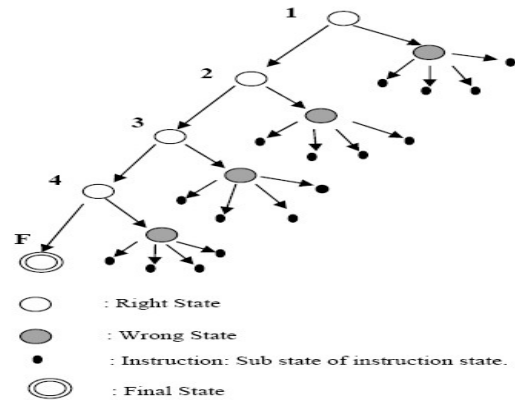


Fig. 7. Bayesian Network of a problem

Let a particular problem, which can be solved using four steps. If a learner can successfully work out the 4th step then he will reach to the final step that means the problem is solved. Here, step 1 (in Fig. 7) is the initial step of the problem then step 2, step 3 and step 4 accordingly. The gray big nodes of this network are indicating the wrong step or wrong movement after every step except the final step. After each wrong move by a learner, the system will search the types of mistakes that had been committed by the learner. According to the mistakes, the system will instruct the learner. In Fig. 6 the instructions are indicating as the small black nodes. Initially, we have surveyed minimum four kinds of mistakes that may be committed by a learner after each problem solving steps. If there found another types of mistakes except these four, then the system will automatically create another instruction node for this particular new mistakes after that wrong steps. So, now there

are five instructions after that particular wrong step, where the learner had committed the wrong move.

C. User Interface

The user interface of ITS have a very important roll and we gave it a careful consideration during the design of the Tutor. The user interface is implemented by popular integrated development environment kits as JAVA. When a learner connects to the Tutor website, the student's browser displays the studying environment for the Tutor. In this environment there is an option for learner to study the tutoring materials. In evaluation environment an appropriate skill-level problem is selected by the expert agent module or the problem that last attempted is presented to the learner.

The student solves the problem in the Solution window, where there will be some step by step options to solve the problem. After answering a step, student presses 'Check' button. Then advisor agent determines the appropriate response based on the diagnosis of the student's answer. The feedback of the advisor agent is sent to the output window. The student, at any time, may explicitly request to the tutor to view the solution, Exit from then current problem and ask for a new one; furthermore, the student can view his performance based on statistics including problems attempted, problems solved, number of attempts on a problem and problem difficulty. A typical conversation is cited below:

Test Result for Learner 1: Name: Suman Dey; ID: IAS012.

Problem: A car is moving at 30m/s. It has acceleration 2m/s^2 . What will be its velocity after 20 sec.?

System: What are the given data?

Learner: Initial Velocity (u), Acceleration (f), and Time (t).

System: Good. Now, what is the required data?

Learner: Final Velocity (v).

System: Good. Select the appropriate formula.

Learner: $s = ut + \frac{1}{2}ft^2$.

System: "You have been given u, f and t; and you need to find out v. Is this your right formula?"

Learner: No.

System: Then what should be the correct formula?

Learner: $v = u + ft$.

System: Good. Now put the values for the variables and calculate v.

Learner: $u=30$

$f=2$

$t=20$.

System: Good.

Learner: $v=70\text{m/s}$.

System: Very Good. Well done.

V. CONCLUSION

In this paper we have proposed an agent based intelligent tutoring systems, where a Bayesian network based intelligent advisor has been presented. Performance of the system will improve automatically when the system is utilized many times. Because, the performance of this system is heavily depend on the Bayesian network or the graph topology of a problem; and this graph topology is automatic updated when

a new types of mistake is found by a learner. So, the system cans advice on this type of mistake for the next learners. The system is expected to enhance the learner knowledge, performance in the problem domain of Newton's Laws of motion. The system presented in this paper has the potential to build estimated Bayesian network structure or graph topology relevant for any other problem domain. The capacity of the system can farther be enhanced by a little modification of the structure of the system by which the system can easily move from instruction state to next state while a learner unable to understand the problem and unable to choose the right steps in a problem solving process.

REFERENCES

- [1] A. Sarrafzadeh, S. Alexander, Farhad Dadgostar, F. C, Abbas Bigdeli, "How do you know that I don't understand? A look at the future of intelligent tutoring systems," *Computers in Human Behavior*, Vol. 24, pp.1342-1363. 2008
- [2] Mohamed Ben Ammar, Mahmoud Neji, Adel. M. Alimi, Guy Gouarderes, "The Affective Tutoring System," *Expert Systems with Applications*, Vol.37, pp.3013-3023. 2010
- [3] Hyacinth S. Nwana, "Intelligent Tutoring Systems: An overview", *Artificial Intelligence Review*, (4), pp.251-277. 1990
- [4] S. S. Abu Naser, "Intelligent Tutoring System (ITS) for Teaching Database to Sophomore Students in Gaza and its Effect on their Performance", *Information Technology Journal*, Vol.5 (5), pp. 916-922. 2006
- [5] S. S. Abu Naser, "A comparative study between Animated Intelligent Tutoring Systems (AITS) and Video-based Intelligent Tutoring Systems (VITS)", *Al- Aqsa University Journal*, Vol. 5, No. 1 Part (1). pp. 72-96. 2001.
- [6] VanLehn, K. 1988. Student Modeling. In Polson, M. C., and Richardson, J. J. (Eds.), "Foundations of Intelligent Tutoring Systems," *Hillsdale, NJ: Erlbaum Associates*, Chap. 3, pp.55-78.
- [7] Maria Virvou, Konstantinos Manos and George Katsionis. "An evaluation agent that simulates atudents' behaviour in Intelligent Tutoring Systems", *IEEE International Conference on Systems, Man and Cybernetics*, Vol.5, pp. 4872 - 4877. 2003.
- [8] Pearl, J., "Probabilistic Reasoning in Intelligent Systems," Morgan Kaufmann, San Francisco, CA, 1988.
- [9] Forslund, G., "Toward Cooperative Advice-Giving Systems: A Case Study in Knowledge Based Decision Support," *IEEE Expert*, pp. 56-62. 1995.
- [10] Vanguard Software Corporation, Decision Script, 2006. Accessed via www.vanguardsw.com/decisionscript/jgeneral.htm
- [11] Aronson, J. and E. Turban, *Decision Support Systems and Intelligent Systems*. Upper Saddle River, NJ: Prentice-Hall, 2001.
- [12] A. E. E. ElAlfi, M. E. ElAlami, "Intelligent Advisory System for Supporting University Managers in Law," *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 3, No. 1, 2009.
- [13] Fernando Gutierrez, John Atkinson, "Adaptive feedback selection for intelligent tutoring systems," *Expert Systems with Applications*, Vol. 38, Issue 5, May 2011, pp.6146-6152.
- [14] F. Sahin and J.S. Bay "Learning from Experience using a decision-theoretic intelligent agent in multi agent system" *Proc. of the 2001 IEEE Mountain Workshop on Soft Computing in Industrial Applications, 2001. SMCCia/01*, , pp.109-114. June, 2001
- [15] D. Heckerman, D. Gieger and M.Chickering, "Learning Bayesian network: The combination of knowledge and statistical data." *Machine Learning*, Vol. 20, pp.197-243. 1995
- [16] Narshing Deo, *Graph Theory: with applications to engineering and computer science*, Prentice- Hall of India Private Limited, 2003.
- [17] L. W, J. Q, "Learning Bayesian network parameters under incomplete data with domain knowledge," *Pattern Recognition*, Vol.42, pp. 3046-3056. 2009
- [18] Ben-Gal I., Bayesian Networks, in Ruggeri F., Faltin F. & Kenett R., *Encyclopedia of Statistics in Quality & Reliability*, Wiley & Sons (2007).
- [19] Y. H. S. Z. and M. C., "Intelligent Software Agent Design Tool Using Goal Net Methodology," *ACM International Conference on Intelligent Agent Technology*, IEEE, pp.43-46, 2007

- [20] S. Z. "Goal-oriented Modeling for Intelligent Agents and their Applications", PhD Thesis, Nanyang Technological University, 2005.
- [21] L. Padgham, John Thangarajah Winikoff, M., "Tool support for agent development using the Prometheus methodology", *Fifth International Conference on Quality Software (QSIC 2005)*, R. Melbourne Inst. of Technol., Vic., Australia, 2005.
- [22] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, PHI Learning Private Ltd., 2008.
- [23] K. Sycara *et al.* "Distributed Intelligent Agents," *IEEE Intelligent Systems*, Vol. 11, 1996, pp. 36-46.
- [24] P. Maes, "Agents that Reduce the Work and Information Overload," *Communications of the ACM*, Vol. 37, Issue 7, Jul., pp. 30-40. 1994



Pratyay Kuila was born in Kolkata, India. He received B.Tech in Computer Science and Engineering from Govt. College of Engineering & Ceramic Technology, Kolkata under West Bengal University of Technology in 2008. He has received M.Tech in the department of Computer Science and Engineering from NITTTR' Kolkata under west Bengal University of Technology in 2011. He is currently at research in Indian School of Mines, Dhanbad, India. His interest field is Artificial Intelligence, e-Learning, Algorithms, ad-hoc and Wireless Sensor Networks.
e-mail: pratyay_kuila@yahoo.com.

Chinmay Basak has completed his B.Tech in Computer Science & Engineering from BCET, Durgapur, India under West Bengal University of Technology in 2006. He has received M.Tech in the department of Computer Science and Engineering from NITTTR' Kolkata under west Bengal University of Technology in 2011. Mr. Basak is presently a Revenue Officer, Directorate of Land & Land Reform, Govt. of West Bengal, India.
e-mail: basakchin@gmail.com.



Dr Samir Roy is presently attached to the Dept. of Computer Science & Engineering of National Institute of Technical Teachers' Training and Research (NITTTR), Kolkata, as Associate Professor. After graduating with honours in Physics from the University of Calcutta, he obtained his B.Tech (1989, University of Calcutta), M.E (1991, Jadavpur University) and Ph.D (2004, University of Kalyani) in the field of Computer Science & Engineering.

He has taught various topics of Computer Science at undergraduate and post-graduate level at various institutes for the past 19 years. His areas of interest include Educational Informatics, Artificial Intelligence, Soft Computing and Theory of Computation. (e-mail: samir.cst@gmail.com)