International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 2, Feb.-2015, pp: (109-114), Impact Factor: 1.252, Available online at: www.erpublications.com

A Bayesian Network Model for More Natural Intelligent Tutoring Systems

Oussama H. Hamid[1,*], FatemahH. Alaiwy[2], Intisar O. Hussien[3]
[123]Faculty of Computer Studies, Arab Open University
[123]P.O. Box 830 Al-Ardia, 92400 Kuwait

***Corresponding author:** Oussama H. Hamid – Email: oh.hamid@gmail.com

---

Abstract:A key element in the design of Intelligent Tutoring Systems (ITS) is to fit the learning plan to the individual variances of learners. We studied this problem by modeling a dependency relationship between concepts of a Java programing course within a Bayesian network. Our model was established on the basis of experts' views, corroborated by experimental investigation. We found that the tested programing concepts do, indeed, exhibit dependency relationships, which could minimize the learning time of the individuals, if they were taken into account. This, implies that for the design of a more natural ITS, it is helpful to consider the existent concept dependency, so as toenhance the role of individual differences between the learners and, hence, minimize the learning time.

Keywords: Bayesian network,concept dependency, intelligent tutoring systems, learning plan.

Introduction

The tremendous improvement in intelligent tutoring systems presents new ways for students to learn and attain higher academic achievements. Intelligent Tutoring Systems (ITS) are computerized learning environments that incorporate computational models from a variety of scientific fields, such as cognitive sciences and artificial intelligence. These systems have the potential to meet the students learning requirements based on their distinct attributes (knowledge, skills, strategies, motivation, emotions, etc.). We propose a Personalized Curriculum Planner - Intelligent Tutoring Systems (PCP-ITS) to improve learning effectiveness and student satisfaction.

A key element in the design of Intelligent Tutoring Systems (ITSs) is to fit the learning plan to the individual variances of learners. According to previous studies, individualizing learning results in more effective performance than neglecting the individual differences, which happens inevitably in group-based traditional learning. This is due to the fact that a system with 'individualized learning' approach takes into account personal factors like experience, age and subject-related background of the learner. Consequently, 'individualized learning' helps structure concrete learning paths that are more suitable for each student. The present paper aims at weighing learners' inherent (pre-existing) knowledge in the design of effectively performing intelligent tutoring systems. The paper is organized as follows: section II reviews related works. Our methodology in determining learners' pre-requisite knowledge and the underlying model of the ITS, together with the setup of experimental work are in section III. Corresponding results are discussed in section IV. We conclude and set out for future work in section V.

Related Work

In their review of Development of Intelligent Tutoring, Ahuja and Sille clearly stated that most of the systems they studied neglected the use of a very important factor, specifically, the "match between learner's ability and the difficulty level of the recommended courseware"[1].

Desmarais and Baker presumed that the success of curriculum sequencing lies in tailoring the learning content based on an accurate assessment of a large array of skills with the least possible amount of evidence.This requires a transfer model that can build links among skills, such as prerequisites. For example, such a model would infer that a student's knowledge of English vocabulary terms such as "sibilate" and "quandary" clearly indicates that familiar common terms would probably be unchallenging. Desmarais and Baker proposed the use of Bayesian Network (BN) to represent such a transfer model. According to Desmarais and Baker BN offer a well-defined formalism that lends itself to sound probability computations of unobserved nodes from evidence of observed nodes[2].

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 2, Feb.-2015, pp: (109-114), Impact Factor: 1.252, Available online at: www.erpublications.com

Huang et al studied curriculum interdependencies to determine the order of curriculums to be explored by each student. His goal was not to eliminate some curriculums from students learning path according to their relations, and hence his system starts by measuring all skills regardless of possible existence of relationships[3].

Several researchers (e.g. [4, 5, 6, 7])used Bayesian Knowledge tracing model and students' heuristic performance to compute the probability of knowledge states for each separate skill: known (1) and unknown (0), without considering skills dependency relationship.

Methodology

A. Determining Learners' Pre-Existent Knowledge

To better design and organize an educational course, a tutor has to choose the most appropriate training contents that suit each learner. This requires selecting a minimally comprehensive subset of curriculums that practically meets the learner's needs and omit whatever redundant or less useful learning material. Yet the question is on which basis the selecting and omitting of course contents should be achieved?A solution to this problem is to start by testing learners' knowledge and then according to their correct responses, we can identify the concepts that need to be omitted.

Importantly, our model does not only omit concepts according to the observable responses of students. But it also uses students' responses in order to predict the largest set of concepts that a student already masters, avoiding, hence, to directly test them.

B. Bayesian Networks as a Transfer Model

A Bayesian Network (BN) is a probabilistic model for knowledge representation that facilitates reasoning under uncertainty [8]. Formally, it is illustrated as a directed acyclic graph B = (N, E), with N denoting the set of nodes and E describing the set of edges weighted with probabilistic conditional dependencies between the nodes. In our scenario, a node n ϵN represents a domain variable, i.e. a certain programming concept, such as initialization. An edge e ϵ E refers to the relation between a concept and its assumed prerequisite concept, quantified by their probabilistic conditional dependency. For example, in level 1 of (Fig. 1), the directed arrow from question Q8 (called as child node in BN terminology) to question Q1 (parent node) suggests a dependency of Q8 on Q1. As implication of such a dependency relation, we hypothesize that knowing Q8 is sufficient to knowing Q1. We call this the "Dependency Hypothesis". In general, with a dependency relation between two concepts, there are 4 possible configurations. Either the learner shows (i) knowledge to both concepts (child and parent), or (ii) no knowledge to any of the concepts, or (iii) knowledge to the parent but not to the child concept, or (iv) knowledge to the child but not to the parent concept. Only in the last configuration, the dependency hypothesis in our scenario is considered to be not confirmed.

To determine learners' pre-existent knowledge, i.e. the level of subject-related background, we devised a transfer model that connects related concepts using Bayesian Networks. The purpose of the transfer model is to elicit the maximum set of students' skills on the basis of minimum measurements [2].To illuminate existing links between directly measured and conceptually induced skills within the transfer model, a Bayesian Network (BN) was used [9].

To this end, we designed a BN for java-programming concepts. Initially, the BN was mapped solely on the basis of

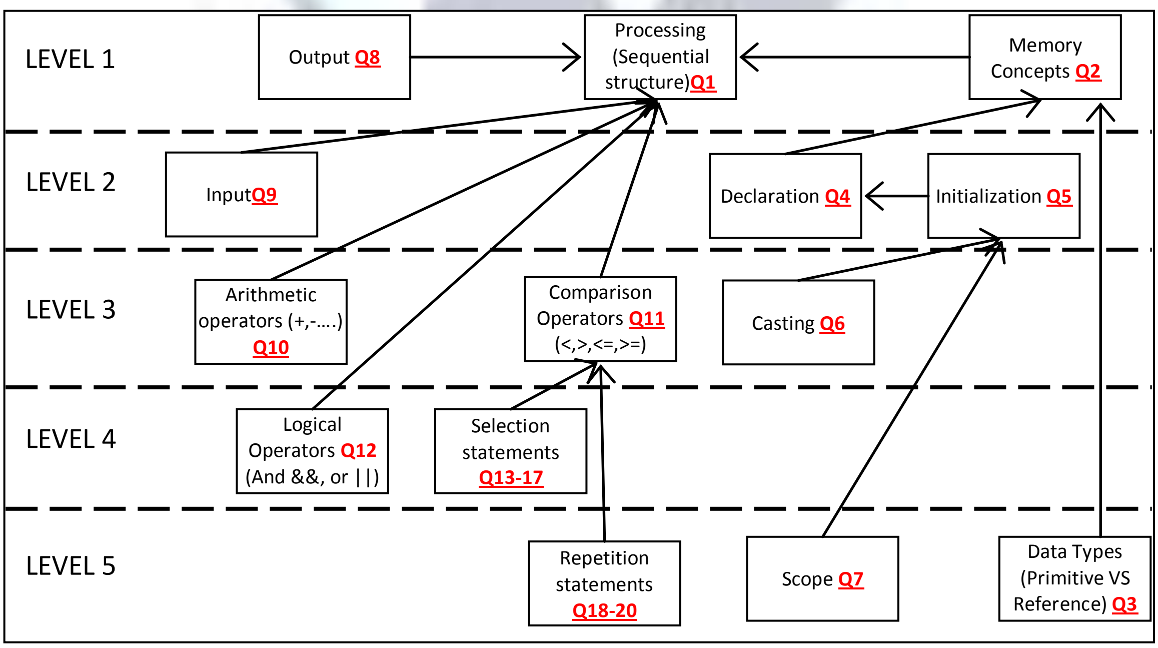

Figure 1 Bayesian network according to experts' views

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 2, Feb.-2015, pp: (109-114), Impact Factor: 1.252, Available online at: www.erpublications.com

prominent Java textbooks [10].In order that we minimize the effect of differences between theory and praxis, we asked tutors, who had an experience in teaching Java for at least 7 years, to review our BN. We, then, modified the BN according to the experts' recommendations(Fig. 1).

C. Experimental Design

To practically affirm the conceptual relations in our BN model, we conducted an experiment, in which an introductory Java-programing crash course was offered to students from the Arab Open University (AOU) in Kuwait. The crash course found the approval of the corresponding officials at AOU and students' participation was made voluntary. The majority (73%) of participating students were completely new to Java. Specifically, 40 students (22 female and 18 male students) attended a 3 hours crash course, comprising two teaching sessions. Each of the teaching sessions lasted for 1.5 hours and was delivered by one of two IT tutors. The tutors (a male and a female) were experts in teaching Java-programming. A break of 15 minutes was awarded to the participating subjects after each session. Subsequently, participants were tested in their acquired knowledge by individually solving a multiple choice questions assessment. The assessment was designed, so that each question tests exactly one of the 20 Java-programming concepts that were discussed in the crash course.

Results

D. Modeling Results: Modifying the Bayesian Network

To know how all concepts (C) depend on the knowledge of a certain conceptC_i, we introduced each of the concepts in terms of test questions (Q). We then calculated the average dependency for each certain questionQ_i as the sum of conditional joint probabilities of the certain concept$C_i$and each other concept $C_j$within the whole set of concepts, given the probability of that certain concept. As a result, we reallocated, if necessary, the level of each concept in the BN according to its dependency relationship with other concepts on the basis of certain expert rules (Tab. 1).Specifically, we obeyed the following steps in determining dependency relationships between concepts:

Step 1: We calculated the average dependency for each concept C_i related to all concepts using the following formula:

Avaerage Dep

$$\sum_{j=1}^{N} P(\,StudentKnowQi\&StudentKnowQj|\,StudnetKnowQj) = \sum_{j=1}^{N} StudentKnowQi\&StudentKnowQj\,/\,StudnetKnowQj$$

Where $N$is the number of all concepts but concept C_i.

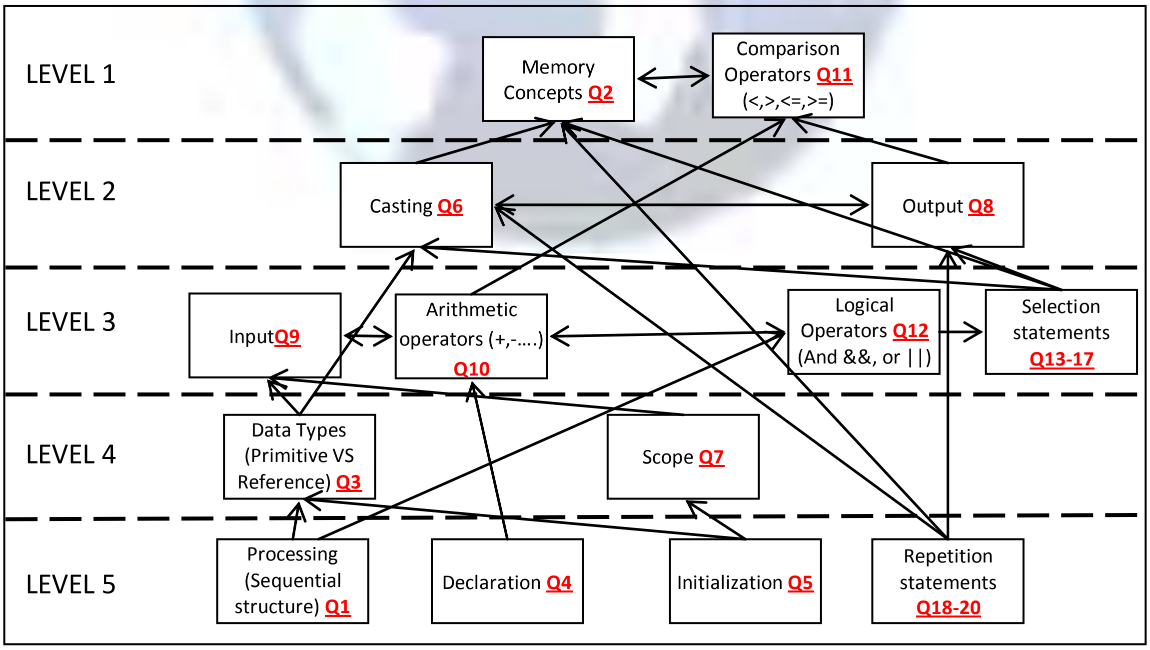

Figure 2 Bayesian network according to students' answers

Step 2: We then changed each concept level in the Bayesian network (Fig. 1) to a new level according to its dependency relationship with other concepts based on the following rules: if the average dependency on C_i was greater than 90%, then that concept should be at level one. If the average dependency on C_i was less than 90% and greater than 80%, then that concept should be at level two. Proceeding like this, we built six different levelsof dependency within the Bayesian network. It remains to link the concepts across the different levels.

Step 3: To establish the relation between concepts in subsequent levels we looked at the value of the formula below, such that concept C_i resides on lower level than concept C_j (for example $C_i$is in level one and C_j isin level 2):

$$\text{Dependency of concept Ci on concept Cj}$$

$$= \text{P(StudentKnow Qi} \,\&\, \mathit{StudentKnow\ Qj} \mid \mathit{StudnetKnow\ Qi})$$

$$= \text{StudentKnow Qi \&StudentKnow Qj} \,/\, \text{StudnetKnow Qi}$$

All values are rounded to one significant figure. This is because we want to distinguish concepts from each other by an interval of 10% accuracy (see Table1). The resulted values are then compared to our threshold value T= 80%: if the dependency value is more than or equal **T** we draw an arrow connecting $C_j \rightarrow C_i$.If there were more than one route connecting two concepts we did omit the shorter one, we also omit arrows showing dependency of concepts on a higher level to a concepts that resides on a lower levels.Implementing the steps from 1 to 3 yields the Bayesian network shown in Fig. 2.

E. Empirical Results: Calculating the Average Dependency of Concepts

Table 1 shows the results of calculating the dependency relationships between concepts on the basis of the devised procedure in steps 1 to 3. In fact, we observed two interesting findings in regards to questions 15, 16 and 17. First, it is worth knowing that these three questions all handle one topic, namely the concept “selection”. Considering them as parent concepts, we noticed that the conditional probabilities for answering all questions starting with question 1 and ending with question 14, have high similar values. This sheds light on two aspects. First, with respect to the value, we may state that there is, indeed, a strong dependency relation between the child and the parent questions. This is evident, since these high values are close to each other in every case of question 15, 16 and 17. Second, that this applies to a large set of questions, starting by question 1 till question 14, suggests that the concept handled by questions 15, 16 and 17 is an essential one in our Bayesian network. Together, this observation excludes one of the possibilities from our experimental paradigm: the possibility of knowing both the child and parent concept on the basis of guessing at least in case of the concept “selection”. However, we may generalize this result to any set of parent questions that have the same child question. It means that the selection concept can be considered as a parent for most of the concepts in java language and therefore if a student correctly answers any programming concept from 1 to 14 there is a high probability that he knows selection.

Moreover, in terms of the dependency hypothesis, we found that in 25.3% of dependency relations, students’ responses did not confirm the hypothesis. Theoretically, this could be due to two possibilities: either the dependency relations in these cases were not established so as to reflect the actual connection between the participating concepts or the students simply guessed their answers. We believe that the first possibility does not apply, since our initial Bayesian network was constructed on experts’ instructions and the designed test consisted of multiple choice questions, which does not exclude guessing.

Conclusion and Scope for Future Work

We have studied the problem of designing an Intelligent Tutoring System (ITS) that fits the learning plan to the individual variances of the learners. As a first step, we proposed an initial Bayesian network (BN) that models the dependency relationships between various topics of a Java programing course. Our main hypothesis was: given a dependency relationship between a child and a parent concept in the BN, then knowing the child concept implies knowing the parent concept. In a next step, we modified the initial BN on the basis of empirical findings gathered and analyzed from a conducted crash course experiment. Our results confirmed qualitatively the dependency hypothesis. Accordingly, for an ITS that takes into account individual differences between the learners, an existent concept dependency could minimize the learning time for learners. This can be achieved by removing unnecessary (to the learner) topics from the learning curriculum. Such topics can be detected by the ITS along the path from the child concept to the root concept of the BN, once the learner shows knowledge of the child concept. For the future, it remains to challenge the efficiency of the underlying ITS model by conducting further experiments that takes into account the individual variances of the learners.

Table 1: Average dependency for each question Q related to all concepts

	Q1 Processing (Sequential structure)	Q2 Memory Concepts	Q3 Data Types (Primitive VS Reference)	Q4 Declaration	Q5 Initialization	Q6 Casting	Q7 Scope	Q8 Output	Q9 Input	Q10 Arithmetic operators (+,-….)	Q11 Comparison Operators (<,>,<=,>=)	Q12 Logical Operators (And &&, or ||)	Q13-17 Selection statements	Q18-20 Repetition statements
Q1 Processing (Sequential structure)		100%	80%	20%	30%	90%	50%	80%	80%	80%	90%	80%	80%	40%
Q2 Memory Concepts	30%		60%	30%	30%	80%	60%	80%	80%	70%	90%	70%	80%	50%
Q3 Data Types (Primitive VS Reference)	40%	100%		30%	40%	80%	70%	80%	90%	80%	100%	70%	80%	50%
Q4 Declaration	20%	100%	60%		40%	60%	70%	100%	70%	80%	80%	60%	80%	50%
Q5 Initialization	30%	100%	80%	30%		80%	80%	100%	100%	80%	100%	70%	80%	40%
Q6 Casting	40%	100%	60%	20%	30%		60%	80%	70%	80%	100%	80%	80%	50%
Q7 Scope	30%	100%	70%	30%	40%	80%		100%	80%	80%	100%	70%	80%	50%
Q8 Output	30%	90%	60%	30%	40%	80%	70%		70%	80%	90%	60%	80%	50%
Q9 Input	30%	100%	70%	30%	40%	80%	70%	90%		80%	100%	80%	80%	50%
Q10 Arithmetic operators (+,-….)	30%	100%	70%	30%	40%	90%	70%	90%	80%		90%	70%	80%	50%
Q11 Comparison Operators (<,>,<=,>=)	30%	90%	60%	20%	30%	80%	60%	80%	80%	70%		70%	80%	50%
Q12 Logical Operators (And &&, or ||)	30%	100%	60%	20%	30%	90%	60%	80%	80%	70%	100%		80%	50%
Q13-17 Selection statements	30%	90%	60%	30%	30%	80%	60%	80%	80%	80%	100%	70%		70%
Q18-20 Repetition statements	20%	100%	60%	30%	20%	80%	60%	80%	70%	70%	100%	70%	60%	
Average dependency (parent relation) with all questions	**30%**	**90%**	**60%**	**30%**	**30%**	**80%**	**60%**	**80%**	**70%**	**70%**	**90%**	**70%**	**70%**	**50%**
Question number	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Q11**	**Q12**	**Q13-17**	**Q18-20**
Question Level	**L5**	**L1**	**L4**	**L5**	**L5**	**L2**	**L4**	**L2**	**L3**	**L3**	**L1**	**L3**	**L3**	**L5**

Authors' Biographies

Oussama H. Hamid received in 2011 a Ph.D. degree in Natural Sciences from the University of Magdeburg, Germany. The title of his doctoral thesis was: "On the Role of Temporal Context in Human Reinforcement Learning". Before, he was awarded a Master's degree in Computational Visualistics (Germany, 2002) and Bachelor degrees in Computer Science (Germany, 2000) and Mathematics (Germany, 1998). In the years between 2003 and 2015, Dr. Hamid has been investigating both theoretical and practical aspects of Human-Machine-Interaction, Neural Network Functions, Cognitive Systems and Big Data Analysis. Dr. Hamid has published his research findings in several European and international refereed journals and conferences. Currently, Dr. Hamid is an Assistant Professor at the Faculty of Computer Studies at the Arab Open University in Kuwait.

Fatemah H. Alaiwy received in 2007 a Master's degree in Computer Engineering from the University of Kuwait. Before, she was awarded a Bachelor degree in Computer Science (Kuwait, 2004). In the years between 2007 and 2013, Mrs. Fatemah has been investigating computer algorithms and optimization techniques. Mrs. Fatemah has published a research about: Optimization of actors' placement within wireless sensor-actor networks (IEEE, 2007).

Intisar O. Hussien holds a Master's since 2001 a degree in computer science from the University of Khartoum, Sudan. Before, she was rewarded a Bachelor degree in Computer Science (Khartoum University, 1993). Currently, Mrs. Intisar is working on her doctoral thesis.

References

[1]. Neelu Jyothi Ahuja and Roohi Sille: A Critical Review of Development of Intelligent Tutoring Systems: Retrospect, Present and Prospect ,2013.

[2]. Michel C. Desmarais , Ryan S. J. d. Baker: A review of recent advances in learner and skill modeling in intelligent learning environments, Springer Science+Business Media B.V., 2011.

[3]. HUANG, Mu-Jung; HUANG, Hwa-Shan; CHEN, Mu-Yen. Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. Expert Systems with Applications, 2007.

[4]. Yue Gong , Joseph E. Beck, Neil T. Heffernan: Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting, 2010.

[5]. HAWKINS, William J.; HEFFERNAN, Neil T.; BAKER, Ryan SJD. Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities. In: Intelligent Tutoring Systems. Springer International Publishing, 2014. p. 150-155.

[6]. WANG, Yutao; HEFFERNAN, Neil T. The student skill model. In: Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2012. p. 399-404

[7]. WANG, Yutao; BECK, Joseph. Class vs. Student in a Bayesian Network Student Model. In: Artificial Intelligence in Education. Springer Berlin Heidelberg, 2013. p. 151-160.

[8]. BEN‐GAL, Irad. Bayesian networks. Encyclopedia of statistics in quality and reliability, 2007.

[9]. REYE, Jim. Student modelling based on belief networks. International Journal of Artificial Intelligence in Education, 2004.

[10]. Deitel, Harvey M., and Harvey Deitel. Java: How to Program (Late Objects). Pearson Education Limited, 2015.