Computers & Education 52 (2009) 762–780

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

Factors influencing the performance of Dynamic Decision Network for INQPRO

Choo-Yee Ting *, Somnuk Phon-Amnuaisuk

Faculty of Information Technology, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 30 August 2007
Received in revised form 28 November 2008
Accepted 2 December 2008

Keywords:
Intelligent Tutoring Systems
Learner modeling
Dynamic Decision Networks
Simulations
Interactive Learning Environment

ABSTRACT

There has been an increasing interest in employing decision-theoretic framework for learner modeling and provision of pedagogical support in Intelligent Tutoring Systems (ITSs). Much of the existing learner modeling research work focuses on identifying appropriate learner properties. Little attention, however, has been given to leverage Dynamic Decision Network (DDN) as a *dynamic* learner model to reason and intervene across time. Employing a DDN-based learner model in a scientific inquiry learning environment, however, remains at infant stage because there are factors contributed to the performance the learner model. Three factors have been identified to influence the matching accuracy of INQPRO's learner model. These factors are *the structure of DDN model*, *the variable instantiation approach*, and *the weights assignment method for two consecutive Decision Networks* (DNs). In this research work, a two-phase empirical study involving 107 learners and six domain experts was conducted to determine the optimal conditions for the INQPRO's *dynamic* learner model. The empirical results suggested each time-slice of the INQPRO's DDN should consist of a DN, and that DN should correspond to the Graphical User Interface (GUI) accessed. In light of *evidence*, *observable* variables should be instantiated to their *observed states*; leaving the remaining *observable* nodes uninstantiated. The empirical results also indicated that *varying* weights between two consecutive DNs could optimize the matching accuracy of INQPRO's *dynamic* learner model.

1. Introduction

Bayesian approach (Howard & Matheson, 1981; Jensen, 2002; Pearl, 1988) to learner modeling has been widely employed by probabilistic Intelligent Tutoring Systems (ITS) to reason and infer about a learner's implicit and explicit behaviour (Bunt & Conati, 2003; Conati, Gertner, & VanLehn, 2002; Jameson, 1996; Millán & Pérez-de-la-Cruz, 2002; Murray, VanLehn, & Mostow, 2004; Pek & Poh, 2005; Phon-Amnuaisuk & Chee, 2005; Reye, 1998; Reye, 2004; Ting, Beik Zadeh, & Chong, 2006). Creating a practically sound Bayesian learner model, however, is not a trivial task because of the uncertainty stemmed from two challenges: (a) choosing the appropriate learner properties and interaction behaviour to be modeled is difficult (Beal, Qu, & Lee, 2006; Beck, 2005; Bunt & Conati, 2003; Phon-Amnuaisuk & Chee, 2005; Ting et al., 2006), and (b) employing a sound inference mechanism that allows probabilistic assessment to temporally variable learner properties is difficult (e.g. Jameson, 1996; Millán & Pérez-de-la-Cruz, 2002; Reye, 1998; Schafer & Weyrath, 1997). Research has shown that failing to address appropriate learner properties can lead to ineffectiveness of the learning environment to achieve its learning objectives (Njoo & De Jong, 1993; Reiser, Copen, Ranney, Hamid, & Kimberg. D., 1994; Shute & Glaser, 1990; Ting et al., 2006; van Jollingen & de Jong, 1991). The study conducted by Njoo and De Jong (1993) for instance has highlighted that a *learner's interaction level* can be a key factor to influence the pedagogical effectiveness of a learning environment. The importance to identify the appropriate learner properties has brought researchers from the area of *user modeling* and ITS to include learner properties from multiple dimensions, such as the *prior knowledge*, *metacognition*, *motivation*, and *time* (e.g., see Beal et al., 2006; Beck, 2005; de Jong & van Joolingen, 1998; Recker & Pirolli, 1992; Reiser et al., 1994; van Joolingen, 1999).

A scientific inquiry learning environment is a learning environment that practices *exploratory* learning approach (Dragon, Woolf, Marshall, & Murray, 2006; Linn, 2000; Paolucci, Suthers, & Weiner, 1996; Pryor & Soloway, 1997; Reiser et al., 2001; Shute & Glaser, 1990; Veermans & van Joolingen, 2004). Learners are granted freedom to interact with the learning environment to enhance their scientific inquiry skills. Among the skills that a learner is expected to master after interacting with such learning environment are *hypotheses generation*, *identifying and controlling variables*, *conduct simulated experiments*, and *compare experiment results*. The importance of these skills to science education can be seen from the recently developed computer-based scientific inquiry learning environments such as the *Belvedere* (Paolucci

* Corresponding author. Tel.: +603 83125217; fax: +603 83125264.
E-mail addresses: cyting@mmu.edu.my (C.-Y. Ting), somnuk.amnuaisuk@mmu.edu.my (S. Phon-Amnuaisuk).

0360-1315/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2008.12.003

et al., 1996), *KIE* (Linn, 2000), *BGuiILE* (Reiser et al., 2001), *SimQuest* (Veermans & van Joolingen, 2004), *SCI-WISE* (Pryor & Soloway, 1997), *Rashi* (Dragon et al., 2006), and *SmithTown* (Shute & Glaser, 1990). To date, limited work has been done to integrate learner modeling into scientific inquiry learning environment, mainly because assessing scientific inquiry skills piece by piece is difficult, and often they are interrelated. In addition, the skills might evolve across time as a learner interacts with the learning environment (Alonzo & Aschbacher, 2004).

ITSs and scientific inquiry learning environments share a common characteristic, in which both contain rich Graphical User Interface (GUI) components to engage learners in the learning. Introducing these components, however, has invited uncertainty during learner modeling task (Baker, Corbett, Koedinger, & Roll, 2005; Njoo & de Jong, 1993; Shute, 1993; Shute & Glaser, 1990). As learners interact with the GUI components, the learner model must leverage on the learners' interactions to model the various reasoning courses exhibited by them (e.g., Bunt & Conati, 2003; Murray et al., 2004; Ting et al., 2006).

Bayesian probabilistic framework particularly the Bayesian Networks (Pearl, 1988) has been employed to tackle uncertainty in learner modeling since almost a decade ago (Jameson, 1996; Reye, 1998). It has been successfully integrated into ITS to provide a sound inference mechanism over a learner's needs and intentions. The *ACE*'s learner model (Bunt & Conati, 2003), *ModelCreator*'s (Tselios, Stoica, Maragoudakis, Avouris, & Komis, 2006) learner model, and the *INQPRO*'s learner model (Ting et al., 2006) for instance employed Bayesian Networks to assess the effectiveness of a learner's exploratory behaviour in an exploratory learning environment and provision of adaptive pedagogical interventions. Bayesian Networks have also been employed in learning environments that focus in facts, concepts, and principles acquisition (e.g., Murray et al., 2004; Pek & Poh, 2005; Tselios et al., 2006). For example in *ANDES* (Conati, Gertner, VanLehn, & Druzdzel, 1997), learners are guided through a series of activities to acquire the physics problem solving techniques. A Bayesian Network is incorporated into *ANDES* to perform on-line probabilistic assessment of a learner's problem solving skills by identifying how the learner progresses down the *solution path* and subsequently determine what hints to provide to maximize the learning experience. In most cases, a *static* Bayesian Network is employed. There are two issues, however, associated with employing a *static* Bayesian Network: (a) unable to model learner properties that evolve across *time*. A *static* Bayesian Network is not appropriate for modeling learner properties that evolve across time mainly because it would ultimately lead to the reinterpretation of previous evidence (Schafer & Weyrath, 1997). In this research work, for instance, learners are given the freedom to navigate from one GUI to another. Freedom in navigation is an important characteristic in scientific inquiry learning environment. Hence, a *static* Bayesian Network can overwrite the *evidence* in previous time-slices; (b) decision-making is not *part of* the network. An explicit look-up table with predefined threshold values required to be setup to allow decision-making. Predefining threshold values, however, is not a trivial task because they may change when the number of variables (nodes) varies.

All these considerations prompted researchers to employ Dynamic Decision Network (DDN) as a *dynamic* learner model to simultaneously assess learner properties and provision of tailored support across time in probabilistic ITSs. Limited research work, however, has been reporting on employing DDN to learner modeling in ITS domain. Murray and VanLehn (2006) reported that DDN outperformed *Fixed-Policy* and *Random Approach* for tutorial action selection in a learning environment named *DT-Tutor*. Similar approach has also been successfully implemented in *Prime Climb* (Conati, 2002), and *i-Tutor* (Pek & Poh, 2005). The similarity among the DDNs employed in those learning environments and other user modeling environments (e.g., Horvitz, Kadie, Peak, & Hovel, 2003; Li & Ji, 2004; Liao, Zhang, Zhu, Ji, & Gray, 2006) is that the variables are identical for all the *n* time-slices. That is, the DDN is formed by repeating the basic structure of the network for every specified time interval. Although such approach has been employed to model the levels of concepts acquisition, motivation, metacognition, and conceptual change process that evolve across time (e.g., see Bunt & Conati, 2003; Murray et al., 2004; Pek & Poh, 2005; Ting, Beik Zadeh, & Chong, 2006), it cannot be generalized for all learning environments, particularly the *exploratory* learning environments because they could consist of multiple GUIs (e.g., INQPRO in this research work). As a proposed solution, the DDN for *exploratory* learning environments with multiple GUIs can be constructed by appending the DN corresponds to the GUI accessed by the learner. However, factors like the DDN structure, Conditional Probability Tables (CPTs) specification for the variables, and the methodology for variable instantiation without the presence of learner interaction could directly influence the overall performance of the learner model.

Employing a DDN is crucial in this research work for three reasons. First, a learner's scientific inquiry skills evolve across time, therefore capturing the dependencies between the temporally variable skills is difficult. Second, freedom to navigate from one GUI to another introduces complexity in predetermining a DDN. A predetermined DDN can easily become computationally intractable because it exhibits 5^n state spaces with $n \in$ {Integer > 0}. That is, the learning path demonstrated by a learner can be represented by *n* time-slices in a DDN, with each time-slice there is a five possible GUIs a learner can navigate to. Third, if a *static* Bayesian Network is employed, the interpretation of new evidence will lead to reinterpretation of previous evidence (Schafer & Weyrath, 1997). In order to overcome such drawback, a DDN should be employed instead of a *static* Bayesian network.

The motivation of this research work lies in finding an optimal DDN model to model a learner's scientific inquiry skills and provision of pedagogical interventions in a timely manner. While existing research work in learner modeling for ITS domain focuses on selecting the most *appropriate* learner properties, this research work investigated factors that influence the performance of a DDN for a learning environment that consists of multiple GUIs and DNs. We believe that our findings can be useful to other ITS domains and those learning environments that leverage DDN.

In the subsequent section, we present an overview of DDN and further with an overview of INQPRO learning environment in section 3. We shall then discuss the construction of INQPRO's *dynamic* learner models in section 4 and the evaluation in section 5. We then further highlight the structure of each DDN model and discuss the performance of each model in Section 6. In the similar section, we also illustrate how two other factors: CPTs specification and variable instantiation approaches influence the matching accuracies. Finally, we end this paper by concluding the contributions of each factor to the performance of DDN employed in the INQPRO learning environment and the future work.

2. Overview of Dynamic Decision Network

We begin with an overview of Bayesian Network (BN) and Decision Network (DN) because these concepts are required to understand the construction of a Dynamic Decision Network (DDN).