Assistance in building student models using knowledge representation and machine learning

Sébastien Lallé[1,2], Vanda Luengo[1], Nathalie Guin[2]

[1] LIG METAH, Joseph Fourier University, Grenoble, France
[2] Université de Lyon, CNRS Université Lyon 1,
LIRIS, UMR5205, F-69622, France
{sebastien.lalle, vanda.luengo}@imag.fr,
Nathalie.Guin@liris.univ-lyon1.fr

Abstract. We propose a method and a first authoring tool to assist the design and implementation of diagnostic techniques. This method is independent from the domain and allows building more than one technique at once. The method is based on knowledge representation and a semi-automatic machine learning algorithm. We tested the method in two domains, surgery and reading English. Techniques built with our method beat the majority class in terms of accuracy.

Keywords: Knowledge diagnostic, authoring tool, machine learning

1 Introduction

In Technology Enhanced Learning (TEL) systems, knowledge diagnostic is the process of inferring a student model using traces collected from a TEL system during the interactions with the learner. Traces are the record of all actions or interactions of the student with the TEL system. Knowledge diagnostic can be used to adapt the behavior of a TEL system to the learner, like providing feedback or choosing the next exercise to practice. A diagnostic technique is a way to do knowledge diagnostic (i.e. infer a student model), like knowledge tracing [4] or constraint-based [12].

A complex and expensive task is the design and the implementation of diagnostic techniques. Actual methods usually require manual work and strongly depend on a particular diagnostic technique. The problem we address is to propose a more generic method to build and evaluate more than just one diagnostic technique.

The content of this paper is organized as follows: previous work and motivations, presentation of our methodology of assistance, experimental results and conclusion.

2 Related work and motivations

There are two main approaches for building diagnostic techniques: manually through authoring tools, and automatically though machine learning. Firstly, authoring tools are environments allowing building a TEL system without programming everything.

adfa, p. 1, 2011.

Some include the design of a diagnostic technique: rules in Eon [11], Model Tracing in CTAT [1], and Constraint-based in ASPIRE [9]. These systems often require to design several components of the TEL system (like the interface), and using existing components like complex interfaces is limited. They support only one diagnostic technique. Secondly the goal of machine learning is to instantiate a generic diagnostic technique to a given domain using students' traces. The result is an instantiated or learned diagnostic technique. Some authors discussed this approach for bug libraries [13], Item-to-Item Theory [5], cognitive modeling [7][2]. The main issues of the results of unsupervised algorithms are their interpretability for humans, their plausibility, and thus their utility. These algorithms can learn only one diagnostic technique.

None of these approaches allow building different kinds of diagnostic. This paper addresses this issue. Our proposition is based both on authoring tools and machine learning, aiming to reduce implementation cost but also to keep the interpretability and the utility of the instantiated diagnostic techniques. Motivations include easing the implementation of diagnostic techniques, the comparison of techniques over several datasets, and the choice of one technique for an existing or a new TEL system.

3 Semi-automatic machine learning method

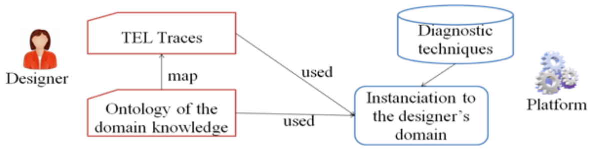

Fig. 1. Schema of the interaction between the user and the platform.

We present in this section our methodology, which we implemented in a first platform. The problem is to assist a designer to instantiate a set of diagnostic techniques for his/her domain, as defined in introduction, thanks to traces. The set of techniques is generic (independent to the learning domains). Instantiate them means to find in traces the different variables required to infer a student model. For instance, what are the skills of the domain, the steps involving each skill?

We addressed two problems. First, the format of traces is unknown and traces may be incomplete. We propose to add semantic to the traces with an ontology of the domain knowledge. Then, design each diagnostic technique independently may be too fastidious for a designer. We propose a machine learning algorithm to instantiate a set of generic techniques using traces. The set of techniques is stored into the platform using a common representation. Currently the techniques are: Knowledge Tracing [4], Additive Factor Model [3], Constraint-based [12], and Control-based [8].

First we propose to the designers to define an ontology of the domain knowledge, and the ontology is mapped to the traces. The ontology does not depend on a diagnostic technique, and does not have to be complete. The goal is to describe variables in the traces and complete the traces. We impose two main classes in the ontology: observable elements and knowledge elements. All new classes inherit of one of these two. The second step is to map the ontology to the traces. Classes or individuals in the ontology are associated by the designer to variables or elements in the traces. Several variables in the traces can be associated to one class in the ontology.

The machine learning algorithm works in three steps. First it associates the variables in the traces to the variables required by each diagnostic technique, using the ontology and the mapping from the ontology to the traces. Thus, each variable of each technique is mapped to the corresponding elements in the traces. Then it extracts all possible values of the variables in the traces. Finally it learns the required parameters such as the probabilities of the Hidden Markov Model used by Knowledge Tracing. The results depend on the ontology. Our assistance is iterative: user shall start with a basic ontology and complete it until results (the learned techniques) are satisfying. We show below how the platform directly helps to evaluate the learned techniques.

4 Experiments and results

We applied our approach in two domains, using students' traces. The first set of traces was collected with TELEOS [8] in orthopedic surgery. We got 2695 correct or incorrect interactions (actions) with the tutor. The second set of traces were collected with the Reading Tutor [10]. We got 240,204 words read fluently or not by a student.

We computed and compared in cross validation how well the instantiated technique fit the traces, by measuring their predictive accuracy, i.e. the percent of good predictions at time t of the student's answer at time $t+1$ (like correct or not). Almost all accuracies beat the majority class (correct actions for TELEOS, words read fluently for Reading Tutor), meaning that the learned diagnostic techniques are more accurate than always predicting the majority class (Table 1).

Table 1. Results for TELEOS and Reading Tutor. 95% confidence intervals in parentheses.

Diagnostic techniques	Knowledge Tracing	Additive Factor Model	Constraint-based	Control-based	Majority class
TELEOS	71% (±2,7%)	70% (±2,8%)	73% (±3,6%)	75% (±3,3%)	54%
Reading Tutor	78% (±3,2%)	78% (±3,9%)	72% (±4,1%)	74% (±3,5%)	74%

5 Conclusion

We proposed a methodology and a first platform for assisting the design and development of different knowledge diagnostic techniques. Our work is independent both from the domain and the diagnostic techniques, allowing building and comparing more than one diagnostic technique. This is new as far as we know. Our method is based on a semi-automatic machine-learning algorithm, driven by an ontology. Results showed accuracies over the majority class in two domains, surgery and reading.

Unlike existing tools, our method is independent from each diagnostic technique, and aims to increase interpretability and utility of learned techniques thanks to the semi-automatic approach. The tradeoff is that a manual work for building the ontology is still required. Choosing a diagnostic technique depends on the goal of the designer, in term of pedagogical strategies implemented in the TEL system, and it is not clear when and why a technique is better than another, as shown in [6]. Our assistance platform can make easier to try, test and compare different techniques.

Future work includes evaluating our platform on more domains, improving the interface of our platform to test its usability, and assisting the design of the ontology.

This work was supported by a PhD scholarship from the Rhone-Alpes Region in France. We thank project LISTEN and TELEOS for the data used in our study.

References

1. Aleven, V., McLaren, B., Sewall, J., Koedinger, K.: The cognitive tutor authoring tools (ctat): Preliminary evaluation of efficiency gains. *8th International Conference Intelligent Tutoring Systems*, 61–70 (2006).
2. Beck, J.E., Woolf, J.E., Beal, C.R..: ADVISOR: a machine-learning architecture for intelligent tutor construction. *17th AAAI Conference on Artificial Intelligence*, 552–557 (2000).
3. Cen, H., Koedinger, K., Junker, B.: Comparing two IRT models for conjunctive skills. *9th International Conference on Intelligent Tutoring Systems*, 796–798 (2008).
4. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modelling and User-Adapted Interaction* 4, 253–278 (1995).
5. Desmarais, M.C., Meshkinfam, P., Gagnon, M.: Learned student models with item to item knowledge structures. *User Modeling and User-Adapted Interaction* 16(5), 403–434 (2006).
6. Gong, Y., Beck, J.E., Heffernan, N.T.: How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis. *International Journal of Artificial Intelligence in Education* 21(1), 27–46 (2011).
7. Gonzales-Brenes, J., Mostow, J.: Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models. *5th International Conference on Educational Data Mining*, 49–56 (2012).
8. Minh Chieu, V., Luengo, V., Vadcard, L., Tonetti, J.: Student modeling in complex domains: Exploiting symbiosis between temporal Bayesian networks and fine-grained didactical analysis. *International Journal of Artificial Intelligence in Education* 20(3), 269–301 (2010).
9. Mitrovic, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., Holland, J.: Authoring constraint-based tutors in ASPIRE. *8th International Conference on Intelligent Tutoring Systems*, 41–50 (2006).
10. Mostow, J., Aist, G.: Evaluating tutors that listen: An overview of Project LISTEN. *Smart Machines in Education*, MIT/AAAI Press, 169–234, (2001).
11. Murray, T.: Eon: Authoring tools for content, instructional strategy, student model, and interface design. *Authoring Tools for Advanced Technology Learning Environments*, 309–340 (2003).
12. Ohlsson, S.: Constraint-based student modeling. *NATO ASI Series F Computer and Systems Sciences*. 125, 167–189 (1994).
13. Sison, R., Shimura, M.: Student modeling and machine learning. *International Journal of Artificial Intelligence in Education* 9(1-2), 128–158 (1998).