Towards Modeling Chunks in a Knowledge Tracing Framework for Students' Deep Learning

Yun Huang
Intelligent Systems Program
University of Pittsburgh
210 S. Bouquet Street
Pittsburgh, PA, USA
yuh43@pitt.edu

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
135 N. Bellefield Ave.
Pittsburgh, PA, USA
peterb@pitt.edu

ABSTRACT
Traditional Knowledge Tracing, which traces students' knowledge of each decomposed individual skill, has been a popular student model for adaptive tutoring. Unfortunately, such a model fails to model complex skill practices where simple decompositions cannot capture potential additional skills that underlie the context as a whole constituting an interconnected *chunk*. In this work, we propose a data-driven approach to extract and model potential *chunk units* in a Knowledge Tracing framework for tracing deeper knowledge, which is primarily based on Bayesian network techniques. We argue that traditional prediction metrics are unable to provide a "deep" evaluation for such student models, and propose novel data-driven evaluations combined with classroom studies in order to examine our proposed student model's real-world impact on students' learning.

Keywords
complex skill, chunk, robust learning, deep learning, Knowledge Tracing, Bayesian network, regression

1. INTRODUCTION
Knowledge Tracing (KT) [4] has established itself as an efficient approach to model student skill acquisition in intelligent tutoring systems. The essence of this approach is to decompose domain knowledge into elementary skills, map each step's performance into the knowledge level of each single skill and maintain a dynamic knowledge estimation for each skill. However, KT assumes skill independence in problems that involve multiple skills, and it is not always clear how to decompose the overall domain knowledge. Recent research demonstrated that the knowledge about a set of skills can be greater than the "sum" of the knowledge of individual skills [8], some skills must be integrated (or connected) with other skills to produce behavior [11]. For example, students were found to be significantly worse at translating two-step algebra story problems into expressions (e.g., 800-40x) than they were at translating two closely matched one-step problems (with answers 800-y and 40x) [8]. Also, recent research that has applied a difficulty factor assessment [1] demonstrated that some factors underlying the context combined with original skills can cause extra difficulty, and should be included in the skill model representation. Meanwhile, research on computer science education has long argued that knowledge of a programming language cannot be reduced to simply the "sum" of knowledge about different constructs, since there are many stable patterns (schemas, or plans) that have to be taught or practiced [16]. We summarize the above findings and connect them with a long-established concept in cognitive psychology called *chunks*. According to Tulving and Craik [17], a chunk is defined as "a familiar collection of more elementary units that have been inter-associated and stored in memory repeatedly and act as a coherent, integrated group when retrieved". It has been used to define expertise in many domains since Chase and Simon's early research in chess [2]. We argue that modeling chunks is important but it hasn't been well-addressed in the current Knowledge Tracing framework. In order to identify chunks in a modern data-driven manner, we propose starting from automatic extraction of stable combinations between individual skills, or between skills and difficulty factors from huge volumes of data available from digital learning systems. We think that such *chunk units* contain different complexity levels, and more complex chunk units can be constructed from simpler chunk units, so they could and should be arranged hierarchically. So we propose a hierarchical Bayesian network which we consider a natural fit for the skill and student model, rather than alternative frameworks [1, 14, 12].

Meanwhile, complex skill knowledge modeling has been a challenge. Starting from simple variants based on traditional KT [5], more advanced models have been put forward. However, these student models use a "flat" knowledge structure, and research works that consider relationships among skills mostly focus on prerequisite relations [3] or granularity hierarchy [13]. Regarding the data-driven evaluations of student models, problem-solving performance prediction metrics [7, 5] have raised some growing concerns [6, 9]. A recent learner outcome-effort paradigm and a multifaceted evaluation framework [6, 9] offer promising methods that we plan to extend. We also plan to conduct classroom studies that deploy a new adaptive learning system that is based on our proposed student model.

2. PROPOSED CONTRIBUTIONS

The first contribution we expect to achieve is to present a novel perspective and data-driven approach for building (skill and) student models with *chunks*. Second, we aim to present a novel multifaceted data-driven evaluation framework for student models that considers practically important aspects. Third, we aim to demonstrate our proposed model's impact for real-world student learning such as helping differentiating shallow and deep learning, enabling better remediation, and ultimately promoting deep learning.

3. APPROACH AND EVALUATION

3.1 Model Construction

Our proposed student model will conduct performance predictions, dynamic knowledge estimations, and mastery decisions when deployed in a tutoring system. To save space, we only describe the major components here.

3.1.1 Representing Chunk Units

To start, we plan to use the Bayesian network (BN) framework for the final skill and student model. We call our proposed model *conjunctive knowledge modeling with hierarchical chunk units (CKM-HC)* (Figure 1).

- **The first layer** consists of basic individual skills (e.g., K_1) that capture a student's basic knowledge of a skill.
- **The intermediate layers** consist of *chunk units* (e.g., $K_{1,2}$), which can be derived from smaller units that capture deeper knowledge.
- **The last layer** consists of *Mastery* nodes (e.g., M_1) for each individual skill, which reflects the idea of granting a skill's mastery based on the knowledge levels of relevant chunk units. We now assert mastery of a skill by computing the joint probability of the required chunk units being known.

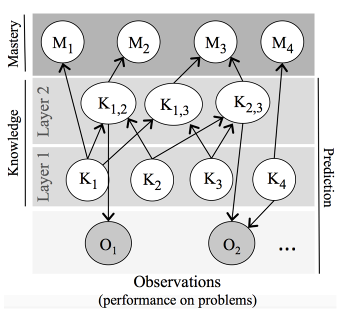

Figure 1: The BN structure of CKM-HC, with pairwise skill combinations as chunk units, in one practice time slice.

3.1.2 Identifying Chunk Units

We consider the following two frameworks to extract chunk units, with Bayesian network as the major framework:

- **Regression-based feature selection or structure learning framework.** Based on regression models, many *efficient* feature selection or structure learning methods already exist. However, the limitations of this approach include: 1) the compensatory relationship among skills is assumed; 2) it's hard to realize the evidence propagation among skills in a probabilistic way; and 3) it doesn't provide the explicit knowledge level of each individual skill. Still, we might be able to use this framework for exploratory analysis or for pre-selection, due to its potential efficiency.
- **BN-based score-and-search framework.** We can employ a search procedure for learning the structure; namely, what chunk units to include. However, if we don't limit the search space, the complexity will grow exponentially. As a result, we propose a greedy search procedural that requires a pre-ranking of the candidates for chunk units. During each iteration, it compares the cost function value of the network with a chunk unit that is newly incorporated with that of the optimal network so far.

To rank chunk units, we use the following general information that should be available across datasets or domains:

- **Frequency information based on skill to problem q-matrix.** Chunk units with higher frequencies, according to the q-matrix, can be considered to be more typical or stable patterns to be modeled.
- **Performance information based on student performance data.** We can employ various strategies, such as giving higher scores to chunk units with larger difference in the estimated difficulty between the current chunk unit and its hardest constituent skill (unit).
- **Natural language processing on the problem (solution) text.** We can consider information such as the textual proximity and semantics that can be obtained by automatic text analysis (or natural language processing).

To further improve the *interpretability*, *robustness* and *generality*, we can also use some domain-specific knowledge to extract more meaningful or typical chunk units. For example, in programming, we can use the abstract syntax tree as in [15]. However, there are still two other challenges:

- **Model run-time complexity.** Since the network involves latent variables, we use Expectation-Maximization, which computes the posteriors of latent variables in each iteration, which can be a time-consuming process.
- **Temporal learning effect.** It is also challenging to consider the temporal learning effect in such a complex network. As a first step, we ignore it during the model learning process, while maintaining the dynamic knowledge estimation during the application phase, as in [3].

We expect to explore some efficient implementations and techniques (such as re-using some posteriors or using approximate inference) to address these two challenges.

3.2 Model Evaluation

We will conduct both data-driven and classroom study evaluations to compare our model with alternatives, including traditional KT-based models [4, 5], and BN-based models with chunk units incorporated in a non-hierarchical way.

3.2.1 Data-driven Evaluation

First, we will conduct data-driven evaluations that consider:

- **Mastery accuracy and effort**. The basic idea of the mastery accuracy metric is that once a student model asserts mastery for an item's required skills, the student should be very unlikely to fail the current item. Meanwhile, the mastery effort metric empirically quantifies the number of practices that are needed to reach mastery of a set of skills. These metrics extend our approach in [6].
- **Parameter plausibility.** This metric investigates how much the fitted parameters can satisfy a model's assumptions and can be interpreted by a human. This is based on our recent Polygon evaluation framework [9].

- **Predictive accuracy of student answers.** This metric evaluates how well the new model predicts the correctness of a student's answer, or the content of a student's solution, based on the problem type.

3.2.2 Classroom study evaluation

We will conduct classroom studies, based on an adaptive learning system that applies our new student model. This system will contain a new open student model interface and a new recommendation engine that will be enabled by our new student model. We will focus on following questions:

1. Do students agree more with the knowledge and mastery inference obtained from the new student model?
2. Does the new student model increase students' awareness of pursuing true mastery?
3. Does the new student model enable more helpful recommendation or remediation?
4. Do students using the new adaptive learning system enabled by the new student model achieve deeper learning which is measured by specifically designed tests?

4. CURRENT WORK

We have conducted preliminary studies with skill chunk units extracted from pairwise skill combinations on a Java programming comprehension dataset and a SQL generation dataset collected across two years from University of Pittsburgh classes. Due to the runtime limitation, we employed a heuristic approach to choose skill combinations (without a complete search procedural), and conducted data-driven evaluations (by 10-fold cross validation). We found that incorporating pairwise skill combinations can significantly increase mastery accuracy and more reasonably direct students' practice efforts, compared to traditional Knowledge Tracing models and its non-hierarchical counterparts. The details of this study are reported in [10].

5. ADVICE FOR FUTURE WORK

I am seeking advice on any of the following aspects:

1. Is this idea both significant and valuable? For example, can it be connected or applied in a broad range of tutoring systems or domains?
2. Are there any datasets, domains or tutoring systems suitable for exploring this idea? What should be the desirable characteristics of the datasets?
3. Are there better representations for skill chunks within or beyond Bayesian networks (e.g., Markov random field, case-base reasoning)? Are there better techniques to identify such units?
4. Are there any suggestions for the overall procedures of this research? For example, should we do a user study to investigate this phenomenon before data mining? If so, how should we design such a study, since we can only test limited chunk units? Should we construct ideal datasets where chunk units are expected to be significant, rather than focusing on existing datasets?
5. How should we situate our definition of chunk units in a broader context considering different domains, problem (task) types and cognitive psychology theories? Is *chunk* the right word? What's its connection with production rules, declarative and procedural knowledge, Bloom's taxonomy?

6. REFERENCES

[1] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems*, volume 4053 of *Lecture Notes in Computer Science*, pages 164–175. Springer Berlin / Heidelberg, 2006.

[2] W. G. Chase and H. A. Simon. Perception in chess. *Cognitive psychology*, 4(1):55–81, 1973.

[3] C. Conati, A. Gertner, and K. Vanlehn. Using bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417, 2002.

[4] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[5] Y. Gong, J. E. Beck, and N. T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proc. 10th Int. Conf. Intelligent Tutoring Systems*, pages 35–44. Springer, 2010.

[6] J. P. González-Brenes and Y. Huang. Your model is predictive but is it useful? theoretical and empirical considerations of a new paradigm for adaptive tutoring evaluation. In *Proc. 8th Intl. Conf. Educational Data Mining*, pages 187–194, 2015.

[7] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proc. 7th Int. Conf. Educational Data Mining*, pages 84–91, 2014.

[8] N. T. Heffernan and K. R. Koedinger. The composition effect in symbolizing: The role of symbol production vs. text comprehension. In *Proc. 19th Annual Conf. Cognitive Science Society*, pages 307–312.

[9] Y. Huang, J. P. González-Brenes, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In *Proc. 8th Int. Conf. Educational Data Mining*, pages 203–210, 2015.

[10] Y. Huang, J. Guerra, and P. Brusilovsky. Modeling skill combination patterns for deeper knowledge tracing. In *the 6th Int. Workshop on Personalization Approaches in Learning Environments (In Submission)*, 2016.

[11] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.

[12] B. Mostafavi and T. Barnes. Evolution of an intelligent deductive logic tutor using data-driven elements. *International Journal of Artificial Intelligence in Education*, pages 1–32, 2016.

[13] Z. A. Pardos, N. T. Heffernan, B. Anderson, and C. L. Heffernan. The effect of model granularity on student performance prediction using bayesian networks. In *Proc. 11th Int. Conf. User Modeling*, pages 435–439. Springer, 2007.

[14] R. Pelánek et al. Application of time decay functions and the elo system in student modeling. *Proc. 7th Int. Conf. Educational Data Mining*, pages 21–27, 2014.

[15] K. Rivers and K. R. Koedinger. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education*, pages 1–28, 2015.

[16] E. Soloway and K. Ehrlich. Empirical studies of programming knowledge. *IEEE Trans. Software Engineering*, SE-10(5):595–609, 1984.

[17] E. Tulving and F. I. Craik. *The Oxford handbook of memory*. Oxford: Oxford University Press, 2000.