Data-Driven Probabilistic Framework for Student Learning

Ashwin Venkataraman[1], Rishabh Ranawat[1], Shiva Iyer[1], Srikanth Jagabathula[2], Lakshminarayanan Subramanian[1]

[1] Department of Computer Science, NYU Courant
[2] IOMS Department, NYU Stern School of Business
{ashwin, shiva.iyer, lakshmi}@cs.nyu.edu, rdr324@nyu.edu, sjagabat@stern.nyu.edu

Abstract

Understanding the cognitive and behavioral aspects of student learning in a principled manner can enable educators and psychologists to improve the state of education. With this in mind, we propose a novel data-driven probabilistic framework to model student learning over a period of time. Our framework provides a means to quantify and track student learning as a function of critical factors such as student skill level, quality of instruction and the amount of prerequisite understanding. We evaluate our proposed model on a real dataset of student responses and show that it achieves good accuracy in predicting responses for previously unseen questions.

1 Introduction

The ability to understand and quantify characteristics of student learning can have immense benefit in terms of improving the quality of instructional content as well as personalizing instruction to individual students. There has been research focused on tracking and modeling student learning in Intelligent Tutoring Systems [Chang *et al.*, 2006; Corbett and Anderson, 1995; Villano, 1992]. However, many of these models lack a relation to the psychological and cognitive aspects of student learning. In this paper, we propose a novel probabilistic framework to quantify, track and understand student learning along two cognitive dimensions—*grasping power* and *takeaway fraction*. Our model heavily relies on David Ausubel's idea that the single most important factor influencing learning is what the learner already knows [Ausubel *et al.*, 1978]. Specifically, our framework models sequential learning of a set of related concepts over time, wherein students incrementally build on what they have learnt so far. We evaluate our model on a popular real-world dataset for benchmarking student learning models—ASSISTments [ASS,] and show that it achieves an accuracy of $\sim 83\%$ in predicting student responses on a hold-out set of questions.

2 Data-Driven Student Learning Model

Our goal is to develop a simple, general and mathematically tractable model that can capture the most salient aspects of student learning. To that end, we consider the following scenario: There is a universe $\mathcal{T}$ of *concepts* and the students are enrolled in some education program that teaches these concepts. We abstract away the details of the education program and suppose that each student is presented with a sequence of instruction *steps*, each step designed to teach a single concept. The objective of each student is to maximize the learning of all concepts over the duration of the program.

2.1 Terminology

We first introduce some key terms that form the basis of our model:

- **Concept Utility and Prerequisites.** We suppose that each concept $T \in \mathcal{T}$ is associated with some utility Δ_T that quantifies the "importance" of this concept in the domain of study. Further, the relationship (or dependence) between different concepts is described using a *prerequisite graph* such that $\mathcal{P}(T) \subset \mathcal{T}$ denotes the prerequisites for concept T.
- **Instruction Quality**. We assume that the instruction provided to a student in each step has a certain *quality* associated with it, denoted by $Q(\cdot)$. Our notion of quality incorporates the knowledge level of the teacher and her presentation/communication skills, along with the quality of the educational material itself.
- **Knowledge State.** We also associate a knowledge state (or skill level) $KS(\cdot)$ for each student that indicates the general aptitude for the specific topic/field of study, with higher values denoting more skilled students.

2.2 Student Learning Parameters

We focus on quantifying student learning of the different concepts along two dimensions: i) what is the ability of a student to grasp (or understand) new instructional content for a given concept? We term this the *grasping power* of that student for that concept. ii) how much has a student retained whatever has been taught so far for a given concept? We define this as the *takeaway fraction* for that student and that concept. We describe both of these quantities next.

Grasping power: We assume that every student "learns" a certain fraction of each step's content as she progresses through an instruction sequence. For a student s and concept T_i, we say that $0 \leq \alpha_s(T_i, n) \leq 1$ denotes the grasping power for content presented at step n during the learning process, $n \geq 1$, and is defined as the fraction of step n's content that the student is able to understand.

Takeaway fraction: This quantity measures the net retention for a particular concept at any learning step. Again for student s and concept T_i, we say that $0 \leq F_s(T_i, n) \leq 1$ denotes the takeaway fraction before the start of the n^{th} step of instruction, $n \geq 1$.

We define and choose grasping power and takeaway fraction in the above manner because every instruction step aims to improve or reinforce the understanding of a particular concept, and the goal of the student is to (eventually) fully understand all the concepts.

2.3 Parameter Characterizations

Here, we describe how we update the model parameters given a sequence of instruction steps presented to a student.
Updating Grasping Power. The grasping power is characterized as follows:

$$\alpha_s(T_i, n; \boldsymbol{w}_1, \boldsymbol{w}_2) = G_1\left(Q_n(T_i), KS(s,n); \boldsymbol{w}_1\right) \times G_2\left(H_n(\mathcal{P}(T_i)); \boldsymbol{w}_2\right)$$

where $\boldsymbol{w}_1, \boldsymbol{w}_2$ are parameters, $KS(s,n)$ is the knowledge state of student s at the beginning of step n and $H_n(\mathcal{P}(T_i))$ is the collection of takeaway fractions for concept T_i and its prerequisites $\mathcal{P}(T_i)$ at the beginning of step n. G_1 and G_2 take appropriate functional forms based on the context and we specify an example in Section 3. We can think of $\alpha_s(T_i, n)$ as the probability of getting a question, testing concept T_i, correct at the n^{th} step. The intuition behind the above update is that grasping power of a student is dependent on two factors: (a) the instruction quality (independent of the student) and the student skill level KS (which captures long-term learning) and (b) the ability of the student to retain the corresponding prerequisite concepts up to step n (which captures short-term learning).

Updating Takeaway Fractions. The takeaway fraction is updated as follows:

$$F_s(T_i, n{+}1) = \begin{cases} 0 \text{ if } n = 0 \\ F_s(T_i, n) \text{ if step n does not teach concept } T_i \\ F_s(T_i, n) + (1 - F_s(T_i, n)) \times \alpha_s(T_i, n) \text{ else} \end{cases} \quad (1)$$

The equation above says that after n learning steps, the total retention for concept T_i increases by a fraction of the amount that was not learned earlier, $1 - F_s(T_i, n)$. That fraction, $\alpha_s(T_i, n)$, is precisely the grasping power of the student at step n.

Updating Knowledge States. After a sufficient number of instruction steps (such as at the end of every course), the Knowledge State of the student is updated as follows:

$$KS_{new}(s) = KS_{old}(s) + \sum_{T_i \in \mathcal{T}} F_s(T_i, N) \times \Delta_{T_i} \quad (2)$$

where N is the number of steps since the last update. This update rule is motivated by the intuition that certain concepts are more fundamental or important (as measured by the utility parameter Δ_{T_i}) in capturing the skill level in a particular domain.

2.4 Estimating Model Parameters

In order to estimate the parameters $\boldsymbol{w}_1, \boldsymbol{w}_2$ we assume access to some observed feedback (in this case binary but this could be extended to handle non-binary responses too) from each student. Consider the set of sequences of student responses as follows:

$$\begin{aligned} seq(s_1) &= \{score(T_{11}), score(T_{12}), \ldots, score(T_{1n_1})\} \\ seq(s_2) &= \{score(T_{21}), score(T_{22}), \ldots, score(T_{2n_2})\} \\ &\cdots \\ seq(s_k) &= \{score(T_{k1}), score(T_{k2}), \ldots, score(T_{kn_k})\} \end{aligned}$$

where $score(T_i)$ is a binary value: 0 (incorrect) or 1 (correct).

For each of these sequences, we can compute the probability of observing them under our student model and we use maximum likelihood estimation to estimate the parameters $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. We use an off-the-shelf method like BFGS to optimize the likelihood function.

3 Early Results

To evaluate the performance of our student learning model, we conducted experiments on a real world dataset - ASSISTments. This dataset is a problem-logs table i.e., each row is a record of a student's response to a particular question and consists of fields such as correct, response time, number of attempts etc. We randomly sample 100 students and 10 concepts. We manually build a prerequisite graph for the chosen concepts. 80% of the questions that each of these students respond to are used for parameter estimation and the remaining 20% are used for testing the model. In all, we have 4710 assessment results across a total of 10 concepts. Since we do not have any information about the instruction quality and the knowledge states of students, we assume that $G_1(\cdot, \cdot) = 1$. For the function G_2, we choose a sigmoid form:

$$\alpha_s\left(T_i, n; \boldsymbol{w}\right) = \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) \quad (3)$$

where $\sigma(\cdot)$ is the sigmoid function and the feature vector $\boldsymbol{x}_n$ of length $|\mathcal{T}|{+}1$ contains the takeaway fractions concept T_i as well as its prerequisites—we put a 0 in entries of the vector $\boldsymbol{x}_n$ corresponding to the remaining concepts (the additional entry corresponds to a bias term).

We use the sigmoid function so that we can represent the grasping power as a probability ($0 \leq \alpha_s(T_i, n) \leq 1$). Furthermore, we update the takeaway fraction (eq 1) if and only if the student's response for the question related to the concept T_i at the n^{th} step is correct.

Here, the task is to predict whether a student's response to the unseen 20% of questions will be correct or not. Here, if α_s for a question is greater than or equal to 0.5, the model predicts the student response to be correct and otherwise incorrect. When we evaluate the probabilistic framework for student learning, we get an average 82.85% test accuracy across the 100 students. We observed that there exists a direct relationship between the magnitude of the weight corresponding to a particular concept and the number of concepts it is a direct or indirect prerequisite of. This suggests that the model is able to capture the hierarchical nature of student learning. Further, we observe that some of the weights are negative. For instance, the weight corresponding to the bias term which suggests that the probability of getting a question correct when a student starts learning is less than 0.5. Similarly, for concepts to which a negative weight was assigned, one can make the claim that these concepts are being newly tested/learned.

References

[ASS,] Assistments dataset. `https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data`. Accessed: 2018-05-01.

[Ausubel *et al.*, 1978] David Paul Ausubel, Joseph Donald Novak, Helen Hanesian, et al. Educational psychology: A cognitive view. 1978.

[Chang *et al.*, 2006] Kai Chang, Joseph Beck, Jack Mostow, and Albert Corbett. A bayes net toolkit for student modeling in intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*, pages 104–113. Springer, 2006.

[Corbett and Anderson, 1995] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1995.

[Villano, 1992] Michael Villano. Probabilistic student models: Bayesian belief networks and knowledge space theory. In *International Conference on Intelligent Tutoring Systems*, pages 491–498. Springer, 1992.