A causal inference framework in education diagnostics with a Bayesian network-inspired concept map

Pavan S Holur
UCLA ECE Department, Master's Project

June 2019

Abstract

Provided a well-defined concept map that is designed by an expert in a broad educational field, hybrid educational platforms can be made to infer conceptual weaknesses and model a student's learning patterns over time. Furthermore, provided the ability to detect and establish confounding factors in the existing graph, one may introduce interventions to generate real factors that were initially confounded. This better models a student's progress. A maintained repository of this work is found here [3].

1 Introduction

Current automated educational platforms, while "smart" in detecting and personalizing educational pathways for students to learn a subject, focus entirely on a pre-defined and fixed progress map, that incrementally tracks a student's score in a particular diagnostic [1]. Personalizing education has taken the form of static modelling: one where a student is correlated to how well they fit an existent pattern of learning (generally stipulated by an expert system). While such an educational pathway guarantees learning, it does not truly guarantee customized learning for every student for multiple reasons:

1. There is no dynamic way to enable tutor-to-platform interaction, to include causal factors in a student's learning progress, for example learning disabilities and/or sickness and extraneous factors, such as boredom, lack of foundation to learn the current subject etc..

2. The complete automation is insufficient to model diverse student approaches to learning, and even develop causal inferences that may be beneficial to impacting a student's learning, provided history of similar-featured students.

Indeed the results-focused and autonomous learning systems today have little-to-no instructor to student interaction and customizing power to cater to a student's or set of students' needs [2]. In this space, we seek to create a hybrid educational platform that, provided a dynamic learning graph, can discover confounding factors in students' learning (with inputs from the instructor), and discover group trends to create more robust personalized tutoring systems provided features from a test subject.

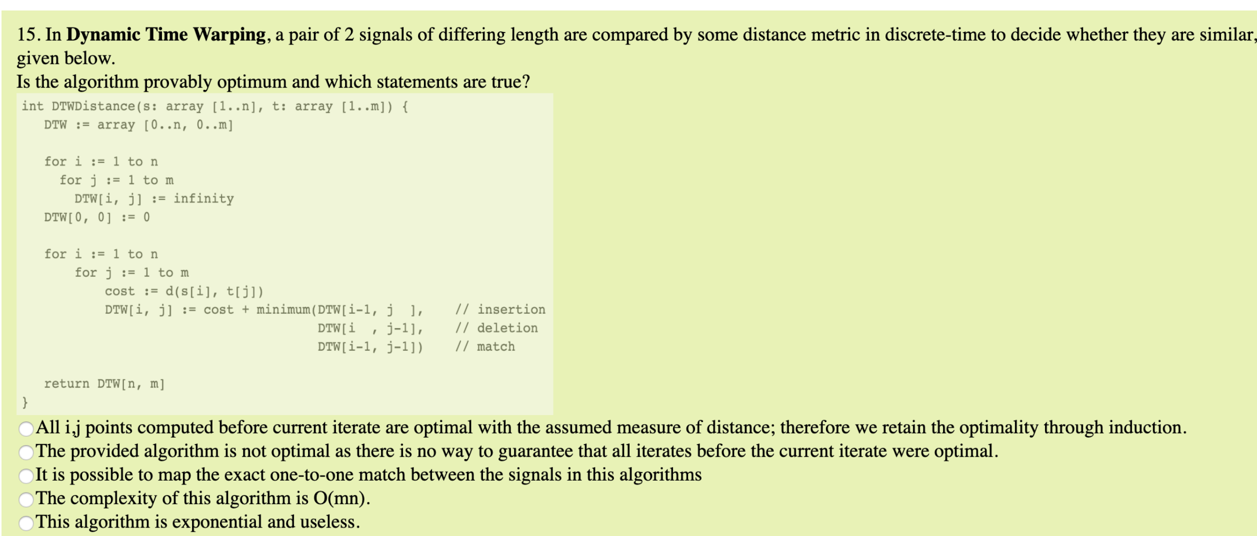
2 Background

In this effort, we propose a small-scale causal inference engine back-end that models a student's progress in learning logic and probability. There is significant literature on the usage of Bayesian networks in creating these inference engines. For instance, for learning proofs, we may design a directed acyclic graph (DAG) that consists of nodes that represent concepts that parent more advanced concepts; i.e each node consists of a sub-topic crucial to understanding a more advanced topic residing in another node. The edges would represent the contribution of these sub-topics in solving the succeeding problems in those topics.

Provided this graph, we can gather diagnostic data from student input on a front-end logic and proofs test that maps a student's responses to this causal graph, and model the areas the student is weak in. Furthermore, by acknowledging confounding factors as nodes in the causal graph, we can attribute some cause to them as well; however more information on the confounding variable must be derived from the student.

3 Problem Formulation

To this effect, we talk about the front-end logic test first. The test itself was designed on a simple HTML website, with multiple-correct multiple choice questions, totalling 80 questions in logic and probability. The questions were designed by extracting thorough information about the requirements of logic in the Engineering curriculum at UCLA, specifically focusing on stochastic logic. Many questions have been sourced from ECE 131 (Probability and Statistics), ECE 231A (Information Theory) and CS M51A (Digital Logic). A sample view of the test front end is given in figure 1.



15. In **Dynamic Time Warping**, a pair of 2 signals of differing length are compared by some distance metric in discrete-time to decide whether they are similar,
given below.
Is the algorithm provably optimum and which statements are true?

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
   DTW := array [0..n, 0..m]

   for i := 1 to n
     for j := 1 to m
       DTW[i, j] := infinity
   DTW[0, 0] := 0

   for i := 1 to n
       for j := 1 to m
           cost := d(s[i], t[j])
           DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                       DTW[i  , j-1],    // deletion
                                       DTW[i-1, j-1])    // match

   return DTW[n, m]
}
```

- All i,j points computed before current iterate are optimal with the assumed measure of distance; therefore we retain the optimality through induction.
- The provided algorithm is not optimal as there is no way to guarantee that all iterates before the current iterate were optimal.
- It is possible to map the exact one-to-one match between the signals in this algorithms
- The complexity of this algorithm is O(mn).
- This algorithm is exponential and useless.

Figure 1: A sample view of the test provided to students. Note: The students do not know of the concept map back-end.

The particular question in Figure 1 tests the student on the subject of Dynamic Time Warping, by providing the student the complete background; however the question asks whether the provided algorithm is optimal. Given that this is a Dynamic Programming-inspired question, the inductive optimality of the algorithm makes the entire algorithm optimal. Therefore, the answers a student provides for this question, are relayed to a large extent to the concepts (or nodes) of "Induction in Optimality Detection" or "Run-time Complexity Computation". Indeed the complete causal graph is shown in figure 2.

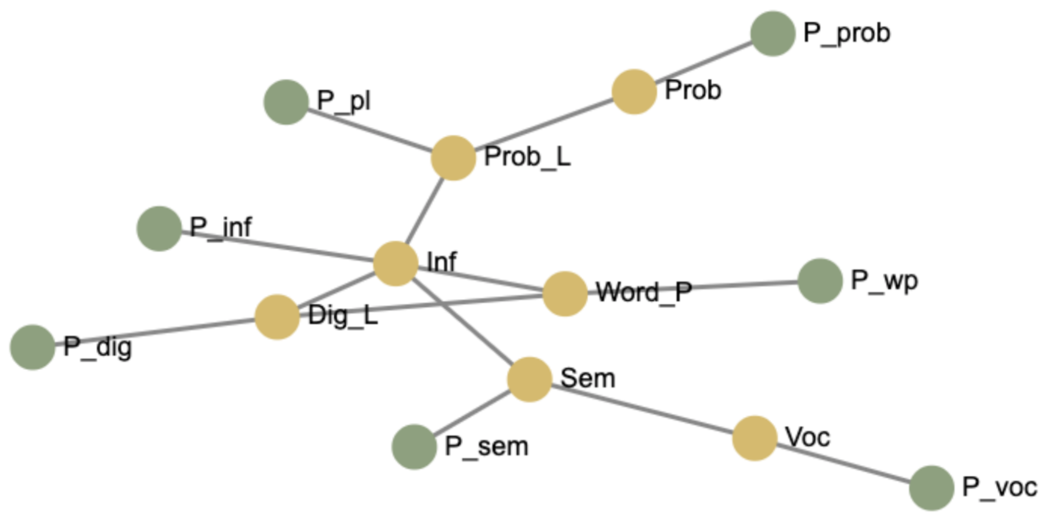


Figure 2: The Back-End Concept Map, inspired by expert system knowledge

In particular, the current implementation of the causal model contains yellow nodes which are termed "Success at a Given Topic" and the green nodes "Practice in a Specific Topic". The edges denote contributions of each topic to a succeeding topic. Right now, the graph has 7 nodes or concepts and edges that result in the production of a DAG. Provided the a-posteriori results from a diagnostic test on a student or multiple students, we model in the following way. Note that $E[X, A]$ implies that X is the set of all nodes whose child is A, and E is the set of edges that capture these relationships.

$$P(A) \triangleq \text{Weight}_{\text{node A}} = \text{Prob}_{\text{empirical fractional success in topic A}} \tag{1}$$

$$P(A) = \sum_{[x,i] \text{ for } i \in E[X,A], x \in X} w_i P(x) \tag{2}$$

4 Analysis and Results

Equation 2 is a Bayesian rule that we can use to determine the conditional parent influence on a child success score. Provided a diagnostic test result, we can then create a set of linear equations (with determinable variables) on the concept map, where the variables are in green. We do not "know" how much practice is needed per concept but we are willing to try to estimate using this expert graph. Running the final analysis tool, we get the following results as shown in figure 3.

```
Net score of user 204403134 is: 80/80
The user is reasonably good at logic! Move to the next module!


Net score of user 987654321 is: 71/80
The user has some ways to improve to be better at logic!
The user would do well to practise on Inference of Simple Logic Expressions and Semantics of Simple Logic Vocab.

Some suitable links to help you get better at your weaknesses are listed here:
Likely confounding cause undiscovered in Studying Probability Logic in Engineering Contexts
>> http://sites.millersville.edu/bikenaga/math-proof/rules-of-inference/rules-of-inference.html
>> http://www.bu.edu/linguistics/UG/course/lx502/_docs/lx502-propositional%20logic.pdf
```

Figure 3: The analysis results based on the posteriori scores by the user. Note that our program handles multiple students and provides detailed analysis of each student along with links to suggested courses.

Furthermore, the individual student's progress can be tracked by a custom tool built to draw plots on a student's concept-based learning pattern with time. In figure 4, is an example with some test data.

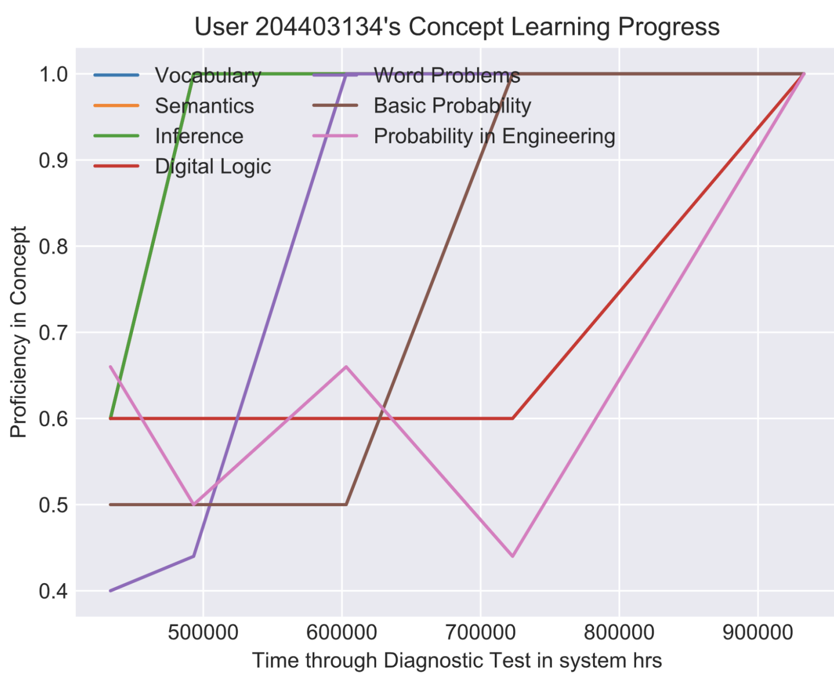


Figure 4: Over time (given in raw server time) the student learns the topics that get them proficient in the larger scope.

Inferences may be drawn based on the properties on this graph. Indeed some of the most glaring issues could help us indicate sure confounding factors in the expert causal graph defined in figure 2. Note that confounding factors may exist even when the probabilities are feasible. This is in fact one of the difficulties in creating such a map in the first place.

1. For example, if the calculated probabilities of a node are over 1 or under 0, it implies that our causal model is not sufficient to model a student's response. Specifically, if a practice node finds a value above 1, it indicates that the drop in abilities between two

successive nodes in a causal graph is too large; i.e there must be a confounding variable (missing a prerequisite class for transfer students for example) that takes some load off the practice node.

2. On the other hand, a sub-zero value in practice indicates the opposite effect. Students who failed miserably at the pre-requisites did exceedingly well at the more advanced topic, indicating perhaps that the student cheated in that section, or just felt bored to answer the simple questions correctly.

3. In figure 4 we can observe that sometimes a student's understanding of a topic can decay, due to the process of "melting"; the student forgets short-term information learned simply to mass the diagnostic test. This negative gradient in the graph can be modelled explicitly to provide more questions in the melting sub-topic in an array of fields, that better solidify the concepts.

Lastly, provided we do receive some confounding variables through time regarding students and their difficulties during taking a course; missing pre-requisites, absent for a critical class, boredom, laziness, mental blocks etc., we can model a renewed Bayesian network with these latent variables added as definitive nodes with weights, that can now contribute to the causal graph.

5 Future Work

Designing diagnostic tools is only a very small area of educational research. Provided an intervention method that may perhaps mitigate some of the nodes' detrimental effects in the causal graph, we still need to formally model the effect of such an intervention on the success of a student in a particular node. In order to achieve this, we may use do-calculus as derived by Pearl [4], as provided a well-formed graph, the rules of do-calculus guarantee to find actionable and conditional probability of improvements in a student's performance.

References

[1] Tara Fenwick and Richard Edwards. Exploring the impact of digital technologies on professional responsibilities and education. *European Educational Research Journal*, 15(1):117–131, 2016.

[2] Jan-Eric Gustafsson. Causal inference in educational effectiveness research: a comparison of three methods to investigate effects of homework on student achievement. *School Effectiveness and School Improvement*, 24(3):275–295, 2013.

[3] Pavan S Holur. Causal-inf-model, Mar 2019. Available at url=https://github.com/pholur/Causal-Inf-Model.

[4] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.