

An Improved Data-Driven Hint Selection Algorithm for Probability Tutors

Thomas W. Price
twprice@ncsu.edu

Collin F. Lynch
cflynch@ncsu.edu

Tiffany Barnes
tmbarnes@ncsu.edu

Min Chi
mchi@ncsu.edu

Abstract

The Hint Factory is designed to generate procedural hints from prior student solutions. We present an algorithm that improves the performance of the existing system in Coherent Derivational Domains. In an experiment, this algorithm improved performance on the cold-start problem and was able to find higher-value hints in previously-unseen states 55% of the time.

Introduction

- The Hint Factory is a **data-driven** hint generation technique
- It operates on an **interaction network**, which represents student states as vertices, actions as edges and solutions as paths
- Hints are provided by matching new students to states in the network and directing them down a known solution path
- We present a novel hint selection algorithm for **Coherent Derivational Domains**, such as probability, where:
 - A student's state is a set of *facts*, derived by applying *rules*
 - No two rule applications are mutually exclusive
 - If S is a solution state, any superset of S is also a solution state

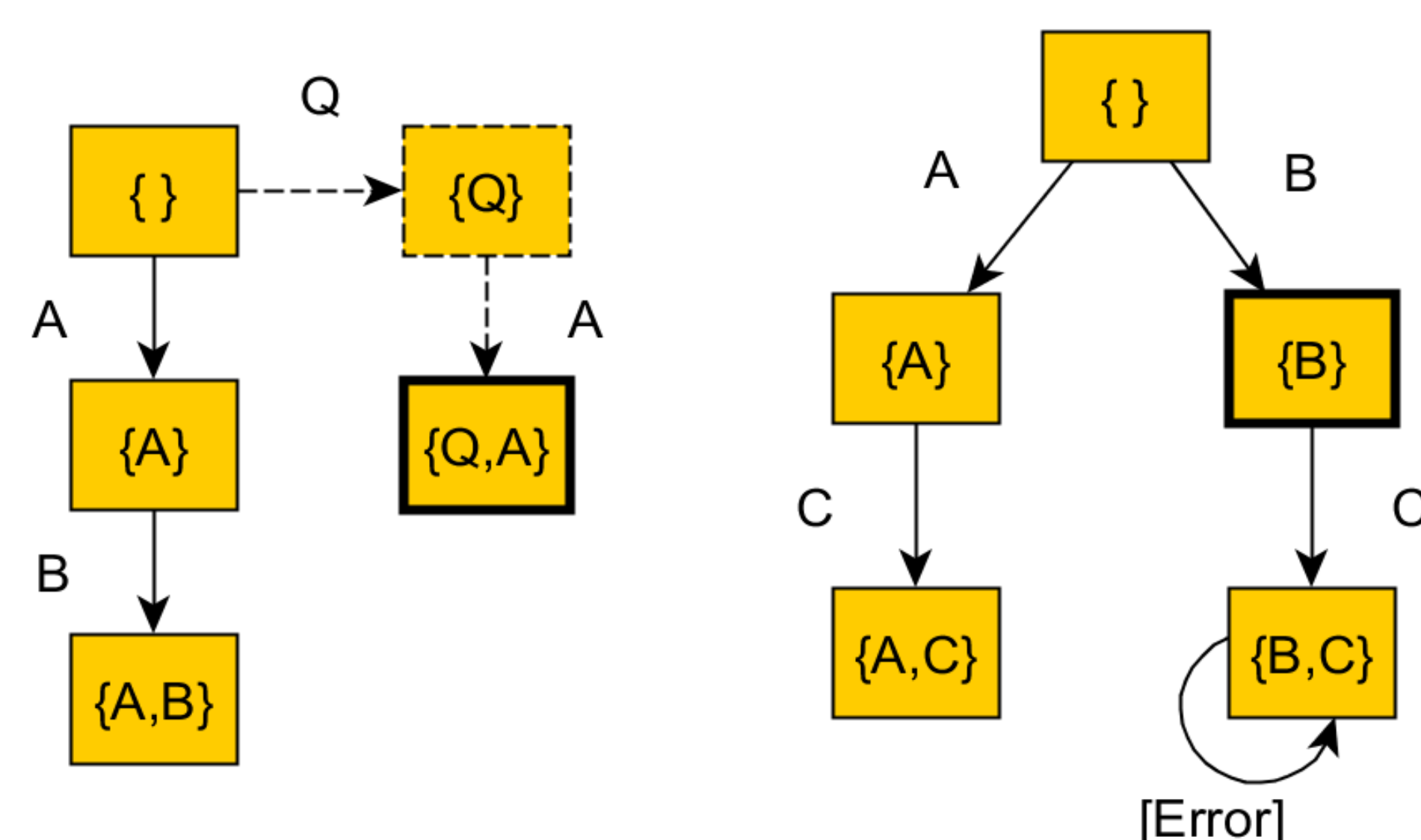


Figure 1 Left, a student derives an unexpected fact (Q), which our algorithm is able to ignore to suggest deriving A . Right, a student starts down an error path, which our algorithm is able to detect, directing the student to an alternate path.

The Algorithm

Input: an interaction network $N=(V,E)$, an ordering function, $f: V \rightarrow \mathbb{R}$, that assigns desirability to states, and a student's current state, s .

Output: a state that will be the objective of the student's hint.

The current **Backup** algorithm looks for an *exact* match to s in the network, and looks through previous states until one is found.

Our **Frontier** algorithm searches for all *applicable* states:

- Let T be the set of all states $t \in V$ where t is a subset of s
- Let the Frontier, F , be the set of all states reachable from some state in T by a single action
- Select the state $v \in F$ such that $f(v)$ is maximized

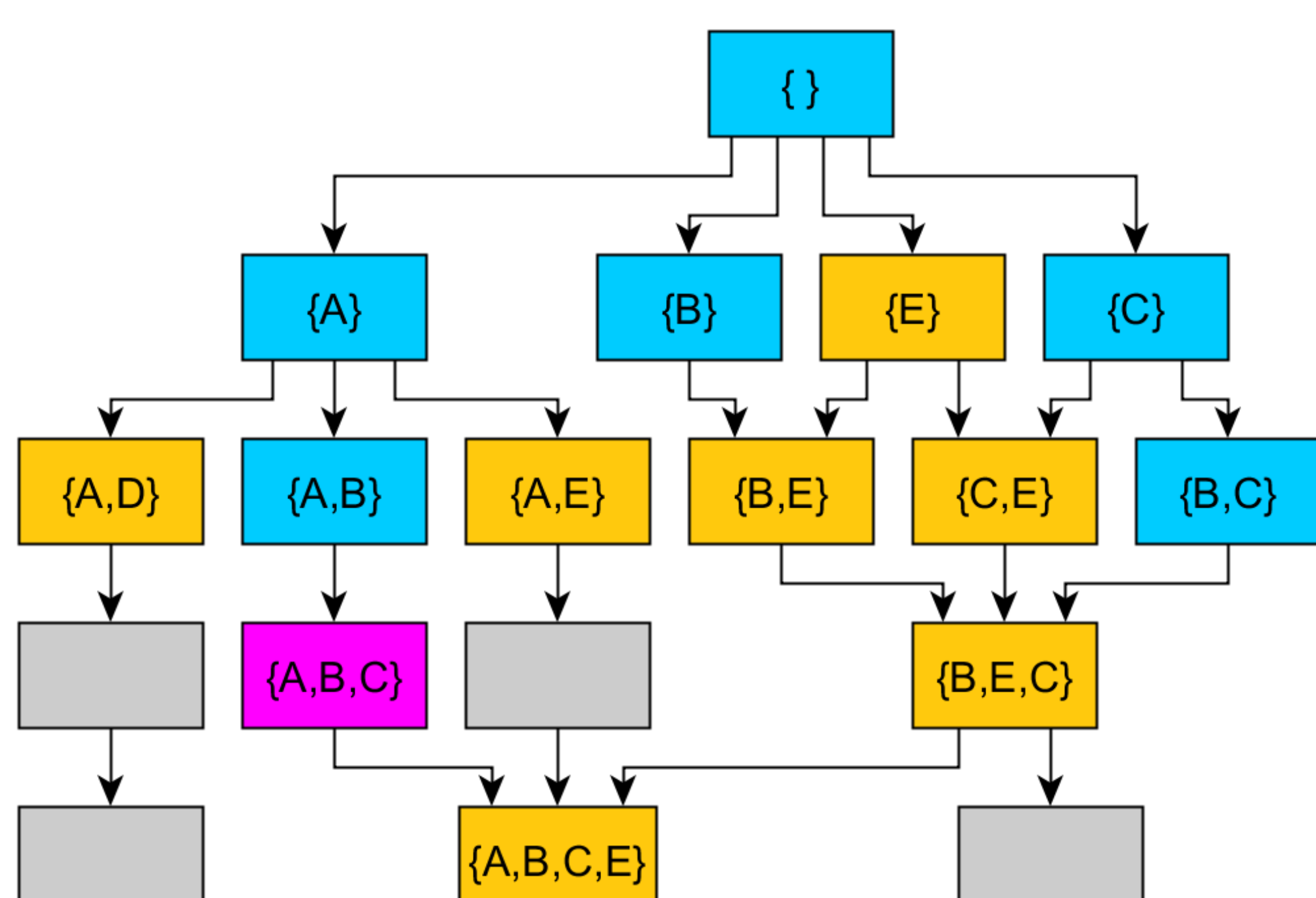


Figure 2 For a student in the magenta state, the set T is shown in blue and the Frontier is shown in yellow; these states can be effectively reached by the student.

Evaluation

- We can measure the quality of data-driven hints via a cold-start curve, plotted as data is added to the network
- We compared our method to the existing Backup algorithm on logs from two probability tutors: Andes and Pyrenees

Procedure:

- Select one student, S , to be the "hint requester"
- Randomly add other students to the network, one at a time
- After each addition, request a hint from each state S reached
- Generate hints with both algorithms and compare
- Repeat 500 times to account for ordering effects

Results

- Ran procedure for 1393 problem attempts over 11 problems
- In *unknown states*, our algorithm gave improved hints 33.5-69.8% of the time, averaging 55.0% over problems
 - These states get very rare as more students are added
- In *known states*, for *some problems*, our algorithm gave improved hints 3.6-49.7% of the the time, averaging 17.8%
 - This effect improved as more students were added
- We get improved hints in data-sparse and data-rich settings

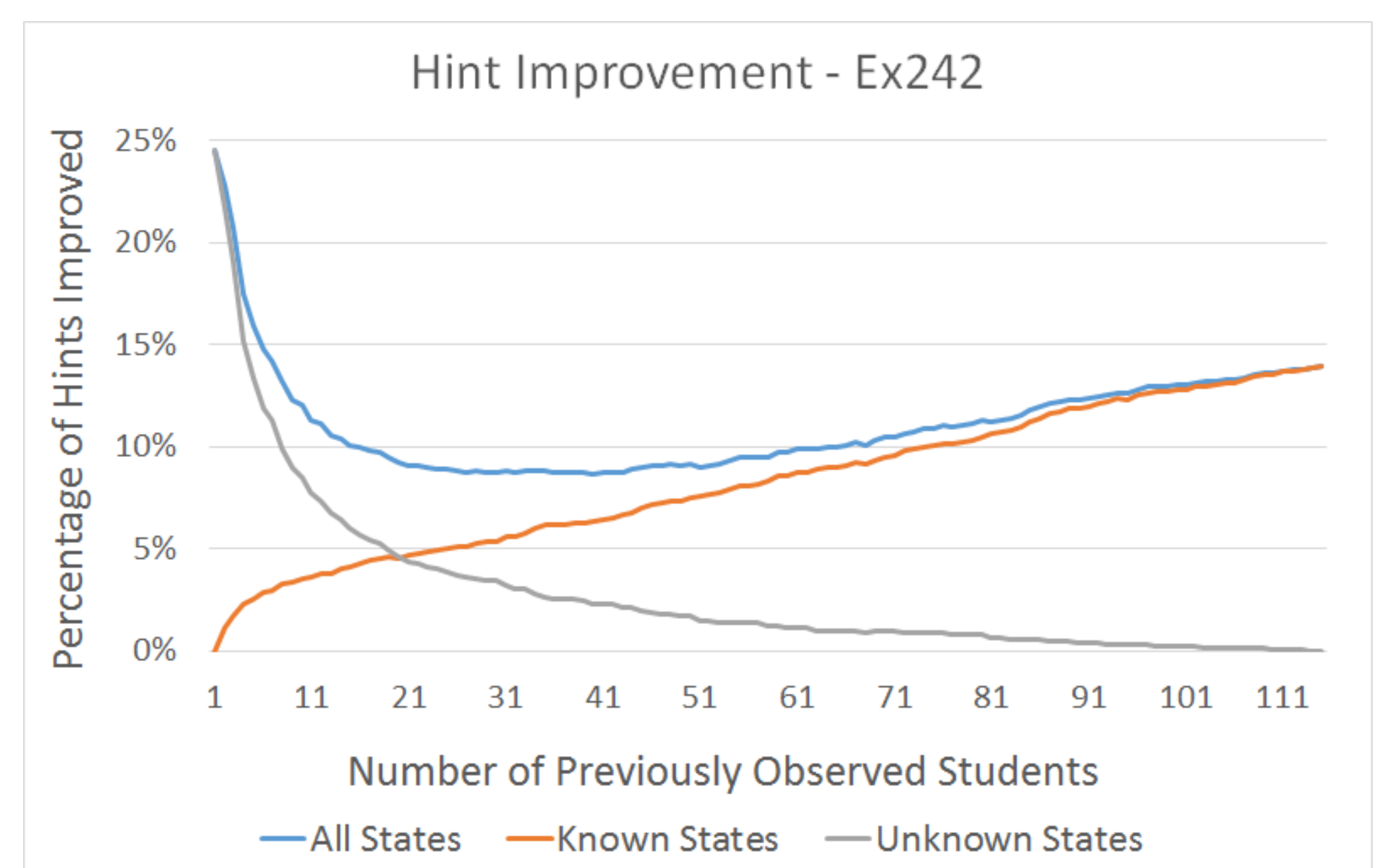


Figure 3 A Cold Start graph for one problem, showing initial improvement for unknown states, and later improvement for known states.

Conclusion

- Our algorithm selects improved hints most of the time when students are in unknown states
 - This is most useful when we have little data and much of the problem's state space is unknown
- For some problems, our algorithm also gives improved hints when students are in known states
 - Results seem very problem dependent, with smaller problems having little to no improvement
- Limitation: the algorithm requires an ordering function, which is only an approximation of state desirability
- Future work will attempt to expand this algorithm to non-derivational domains, and to incorporate other pedagogical considerations (e.g. preferencing a student's current trajectory)