Dynamically Optimizing Learning Plan in Adaptive Instructional System

Ziang Chu, Department of Mathematical Sciences
Carnegie Mellon University

Overview

Introduction:

- An ***Adaptive Instructional System*** *(AIS)* aims at providing an **individualized optimal plan** to achieve learners' learning goals, given their various profiles.
- In our study, we focused on the process of **dynamically** refurbishing and improving the optimal plan based on learners' current status.

Contributions:

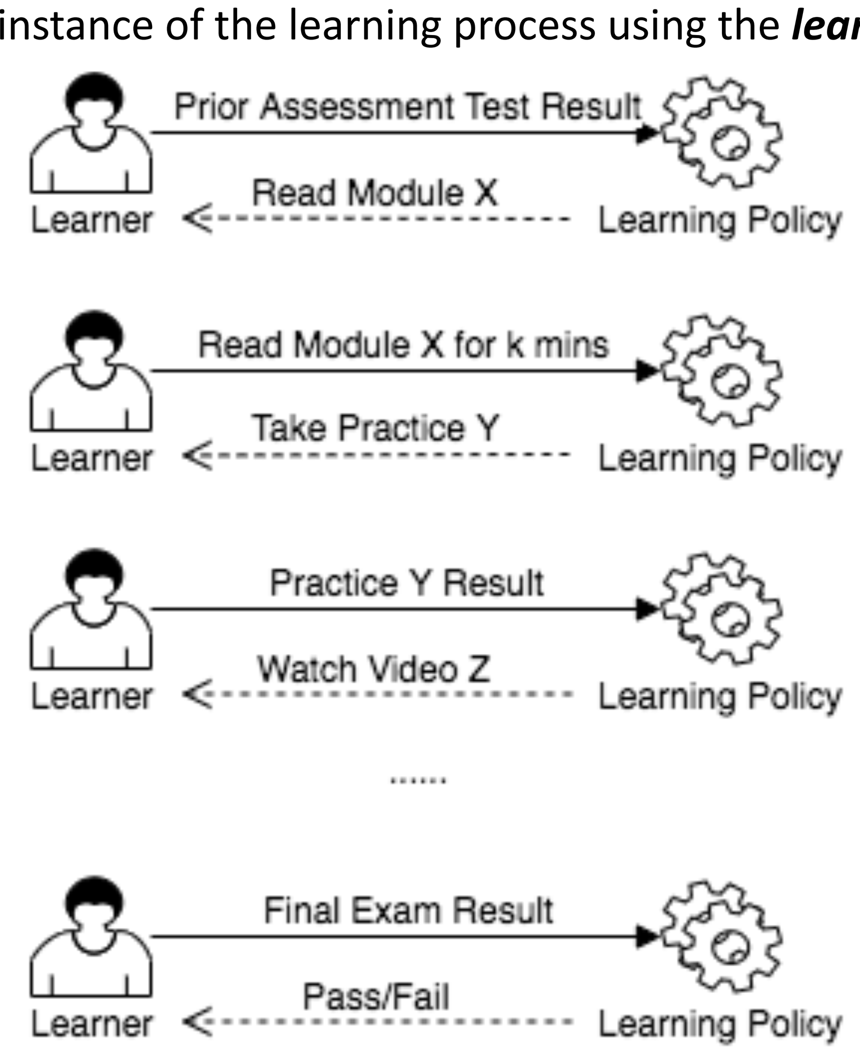
- Formally defined the **Dynamic Learning Plan problem**
- Proposed **Estimated Markov Decision Process** (MDP) approach and **Deep Q-learning** approach to solve the problem.
- Tested and compare the solution qualities of the above algorithms.

Dynamic Learning Plan Problem

Definitions:

- ***Knowledge Components*** (KC): unit of knowledge that can be acquired.
- ***Proficiency Levels***: students' knowledge in one KC[1,3].
- ***Learning Modules***: available learning resource students can take
- ***Learning Goal***: target proficiency levels for a set of KCs.
- ***Learning Policy***: a policy that can dynamically recommend students on next learning module to take based on their learning history records.

- Solve the problem by providing ***a learning policy*** to **minimize the time cost**[2] to reach the learning goal.
- Below is an instance of the learning process using the ***learning policy***.

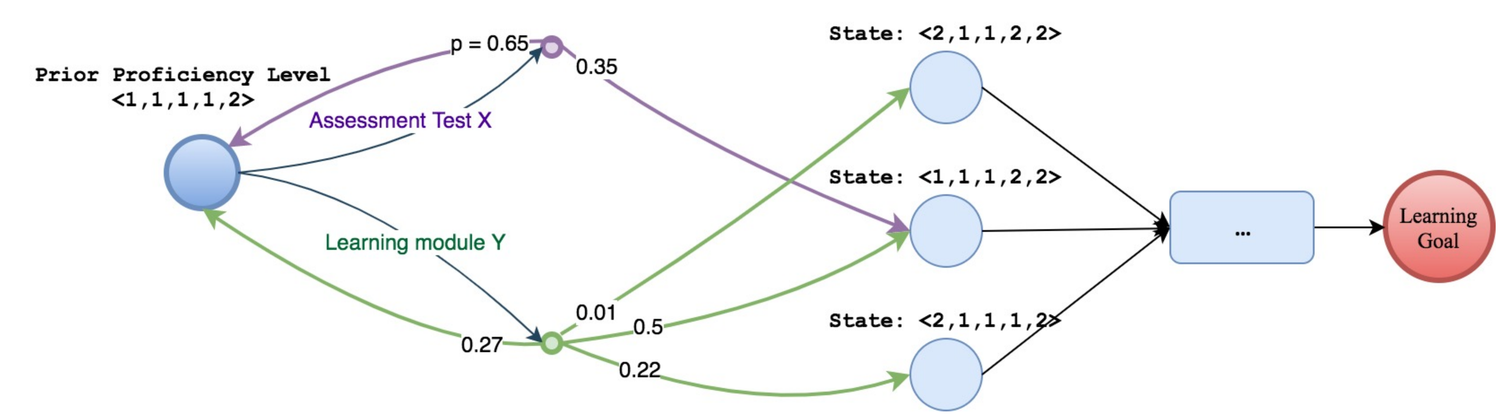


[1] Assume that proficiency levels are discrete and finite. i.e. 0,1,2,…,p.
[2] Assume that learners spend constant time in each learning modules.
[3] Assume that learners' proficiency level never decrease in the learning process.

Estimated Markov Decision Process

- Learner **state**: learner's proficiency levels in each KC; the states is from a finite discrete space with $|S| = \#\, of\, Proficiency\, Levels^{\#\, of\, KCs}$.
- Assume the learning process as a Markov Decision Process and learners have some probabilities from moving to one state to others states by taking each **learning module**.

- **Problem**: we cannot accurately evaluate leaners' states during the learning process.
- **Solution**: summarize the transition probabilities using **interpolation** [1] and construct an **Estimated MDP** model for the problem.
- For unobservable or partially observable state, assume that **all possibilities will happen in equal probability.**

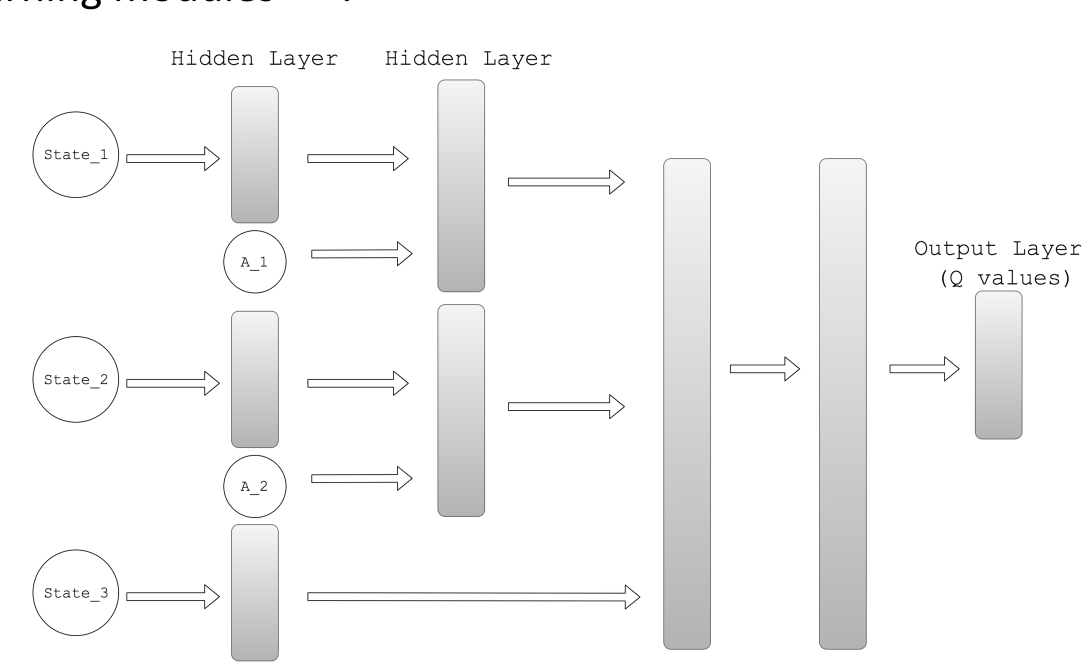


- Using policy iteration to find the optimum policy[1] for the **Estimated MDP**.
- Obtain the **Learning Policy** based on the optimal policy for MDP:
  i. Use the MDP model to maintain a **probability distribution over all states** in each step and do prunes based on learners' practice test results.
  ii. Determine the learning module to take in each step by MDP optimal policy.

[1] MDP Policy π: $S \rightarrow A$, actions include all learning modules and the Final Exam.

Deep Q-learning

- Learning states are unobservable or partially observable. However, based on the assumption that students' proficiency levels never decrease, we can find the **lower bound of the proficiency level** for each knowledge component in each step and we call it **Minimum Proficiency Level State**.
- The Deep Q-learning model takes in the previous 3 **Minimum Proficiency Level States** and previous 2 learning modules and outputs the Q-values of taking any of the learning modules [2,3].
- The Network consists of several fully connected layers, here provides a general view of the **Network structure:**



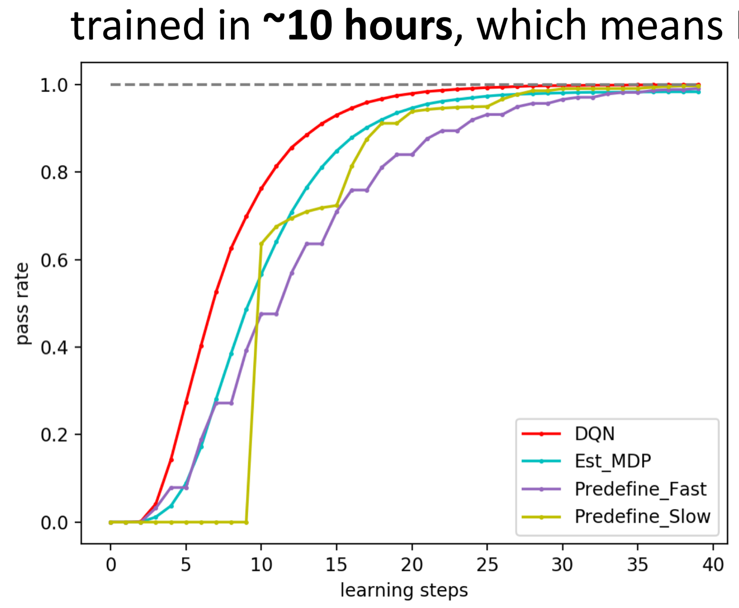
Experiment Set Up

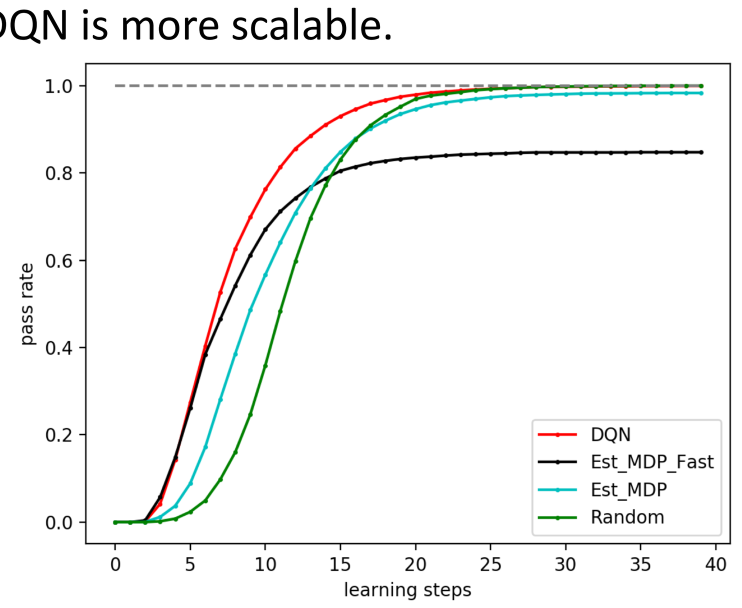
Experiment Scale	Small	Standard
# of Knowledge Components	5	11
# of Proficiency Levels	3	5
# of Assessment Test/ # of Learning Modules	10/20	20/50

- Generate random **student profiles** (prior proficiencies, learning ability. Etc) and **Learning Modules properties** including their knowledge coverage.
- Build **simulators** to simulate students' learning process.
- Train **learning policies** with both algorithms and test them together with some baseline algorithms on 10000 students.
- Record the number of students achieved their learning goals at each step.
- Baseline Algorithms:
  - **Greedy Random**: learners randomly choose learning modules related to their learning currently unreached part of learning goals.
  - **Predefined Policies**: predefined sequences of learning modules that *do not depend* on learners' current states.

Result on Small Scale Problem

- For Estimated MDP Fast, we did interpolation only on top learners.
- Predefine Fast and Predefine Slow are predefined policies, one for top learners and the other for average learners.
- Overall Performance: **DQN > Est MDP ≈ Est MDP Fast > Random ≈ Predefine Fast ≈ Predefine Slow**
- **DQN** performs as good as **Est MDP Fast** for top learners and as good as **Est MDP** for average learners.
- DQN is trained in **~4.5 hours** while MDP is trained in **35 minutes**. However., in the standard scale problem, DQN is trained in ~**5 hours** while MDP is trained in **~10 hours**, which means DQN is more scalable.





Future Work

With fewer assumptions, we want to develop algorithms to work on continues state/action spaces where proficiency levels are continues and learners spend various time on each learning modules.

Acknowledgement

With special thanks to my instructor, Professor Fang Fei, for detailed guidance during my research; Arvind for his contributions on building simulators and providing creative ideas; Richard and KP from **Squirrel AI Learning** for the dataset and valuable feedbacks; and Xiangting for his previous work on the topic.

References
[1] Geo rey J. Gordon, (June 1999), *Approximate Solutions to Markov Decision Processes*, National Science Foundation (NSF) under Grant No. BES-940243
[2] Matthew Hausknecht, Peter Stone, (2015), *Deep Recurrent Q-Learning for Partially Observable MDPs*, arXiv:1507.06527
[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, (2013), *Playing Atari with Deep Reinforcement Learning*, NIPS Deep Learning Workshop