Can reinforcement learning agents be e-teachers? Application of machine learning in education

Marcin Andrzejewski[1], Aleksander Iżemski[1], Piotr Linkowski[1], Kamil Pluciński[1], Eryk Szpotański[1], Mateusz Lango[1], and Krzysztof Jedrzejewski[2]

[1] Institute of Computer Science, Poznan University of Technology, Poznan, Poland
{marcin.m.andrzejewski,aleksander.izemski, piotr.m.linkowski,eryk.szpotanski}@student.put.poznan.pl
[2] AI Learning Capabilities, Efficacy & Learning, Pearson
krzysztof.jedrzejewski@pearson.com

Abstract. E-learning systems are ubiquitous across the whole world. These systems depend on teachers who manually assign tasks to be solved by students in order to optimally develop their skills. However, this is possible only when a teacher runs a course with a small number of students. In large groups teachers are not able to track learning needs of each student. Supporting human teachers with machine learning-based recommendations may allow providing fully personalized learning experience regardless of course size. This paper investigates the possibility of using reinforcement learning algorithms to provide such support. We prepared an environment simulating an e-learning platform. Then we performed an experiment, in which a reinforcement learning agent sets tasks to students to develop their skills. Our experiments show that machine learning algorithms are able to efficiently teach students to increase their level of skill proficiency.

Keywords: reinforcement learning · education · e-learning

1 Introduction

Education is a key element of both personal and economic growth. Better education correlates with higher weekly earnings, a lower unemployment rate, and with the reduction of child mortality in developing countries. Therefore, research towards more effective, personalized learning methods is of particular interest.

Recently, e-learning systems become increasingly popular as they offer multiple benefits such as access from any place with an Internet connection and significantly lower costs compared to traditional learning. Furthermore, e-learning opens possibilities of improving the learning experience by constructing systems adaptively selecting content tailored for the needs of a particular student. Especially, the use of reinforcement learning (RL) to discover better teaching policies can result in much more effective usage of students' time, enabling them to make faster progress and focus their efforts on the most needed learning content.

The idea of individualizing the education process with RL dates back to 1960 [1]. Since then, many different solutions have been proposed. For instance, many works explore the framework of multi-armed bandits to learn the most suitable task for the student. This also includes techniques based on contextual multi-armed bandits exploiting learning data from previous similar students. Another popular approach is to train policies that can provide hints as feedback and scaffolding. The next group of methods applies reinforcement learning for modeling students (i.e. agents) to, e.g., diagnosing students' mistakes based on their attempted solutions. Recently, many works have been conducted on NeurIPS 2020 Education Challenge task, where the algorithms have to estimate the student's knowledge based on a limited number of multiple-choice questions.

This paper investigates the possibility of applying reinforcement learning methods to find a teacher policy for an interactive e-learning platform that maximizes learning outcomes. The solved task is not limited to estimating the students' current knowledge but includes simultaneously estimating and improving students' knowledge. The paper describes the design of an e-learning platform simulator based on a psychometric model with a proper interface for training RL algorithms. The design of other such RL simulators like OpenAI Gym or ViZDoom resulted in bigger research interest and novel algorithms. Another contribution of this paper is the experimental evaluation of several modern reinforcement learning methods on the proposed simulated e-learning platform.

2 Our approach

In this part, we cast the learning process at an e-learning platform to the reinforcement learning framework and describe our implementation of it.

An episode in the environment mimics a student's individual learning process that ends with an exam with previously unknown questions. The agent, which can be understood as a teacher, interacts with the environment by selecting the tasks that the student will attempt to solve. The task is characterized[3] by particular skill needed to solve it (learning objective) and the difficulty level on each skill. The student can solve the task correctly or incorrectly (which also includes the lack of answer), which is reflected in the environment's state, producing feedback for the agent. In both cases, i.e. incorrect/correct answer, the student's proficiency in a skill increases by a certain degree, as even unsuccessful attempts require some cognitive effort and reflection on the use of a given skill. A task can be potentially solved by a student multiple times in the whole learning process. However, in such a case, every attempt is assumed to deal with another task from a group of tasks having same characteristics.

The simulation time is discrete and divided into episodes. Each episode has two phases: learning and evaluation (exam). During the learning phase, the student solves tasks iteratively selected by the teacher. Then, in the exam phase, the student receives a set of tasks to solve. The reward is always 0, excluding the exam phase, when the reward is equal to the number of correctly solved tasks.

[3] The characteristics of tasks are used for underlying computations, and are unknown both to the student and teacher.

The agent's goal is to maximize the number of correctly solved tasks during the exam, which is correlated with the student's skill development.

To model the student's interactions with tasks belonging to each skill, the popular Rasch psychometric model [3] was used. This model describes a student with a continuous proficiency. This value is on a logit scale and usually ranges from -3 to 3. It determines the probability of correctly solving an avg. difficulty task. We used a separate proficiency value for each skill.

In our implementation, the observations returned by the environment mimic a student's grade book and include how many times a given task was solved correctly and incorrectly. More concretely, it is represented as a list with a length equal to the number of possible tasks to choose from, and at position i, it contains tuples of two numbers representing correct and incorrect solutions of i-th task.

3 Experiments

The purpose of the experiments was to investigate the performance of selected RL algorithms, namely A2C (Advantage Actor Critic) [2] and PPO (Proximal Policy Optimization) [4], on the implemented e-learning platform and to compare them to a baseline selecting actions randomly.

Each agent was run for 25,000 episodes. During each episode, the new student was learning s skills by solving tasks (actions) selected by the teacher (agent). For each skill and each of the seven difficulty levels, one task associated with them is available on the e-learning platform. In the case of 7 skills, it means 49 possible tasks to be chosen by the agent. Student proficiencies are sampled from a normal distribution with a mean of 0 and a standard deviation of 1/3, clipped to a range from -1 to 1. At the end of each episode, the student took an exam that consisted of 2 difficult tasks (at the level 6 out of 7) for each skill developed. Experiments with s equal to 1, 3, 5 and 7 have been performed. The length of learning phase was set to $10s$. For example, an episode of teaching 5 skills allows for 50 interactions with tasks.

The goal of the system is to improve the student's proficiency at each skill, so the system was evaluated using the mean of proficiencies. Each parameter configuration was run 5 times with different random seeds[4]. The results of the experiments are presented in Figure 1. The plot on the left shows the mean of student's proficiencies after the exam, when the number of developing skills was set to 5. One can notice here that reinforcement learning approaches are considerably better than the baseline. This means that the agent is able to identify which task support the development of which skill, and based on the observation, estimate which skills the student is deficient in and should improve.

On the other hand, the plot on the right presents mean proficiency of the student after the 25000 episode depending on the number of developing skills and used RL algorithm. As the number of skills increases, the number of possible tasks to choose from increases too. We can observe, that in the result, it becomes

[4] The source code for the experiments can be found in `https://github.com/er713/Schooling-RL`, commit hash: `3f1db56`

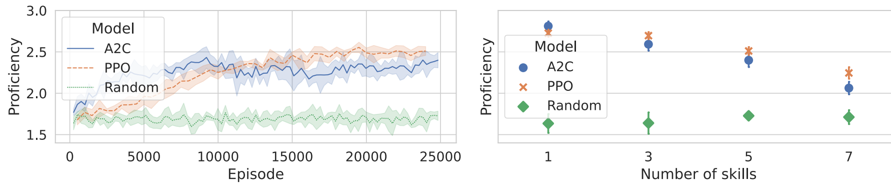

Fig. 1: The figure presents the experimental results. On the left, the difference in performance between algorithms is shown, when the number of skills to learn by the student was set to 5. On the right, the performance of the algorithms after 25000 episodes is compared against the increasing difficulty of the environment (the number of skills and the action space increases). Both plots show the mean values of the 5 runs with different seed values, along with the standard deviation.

noticeably more difficult for the agent to properly map tasks to skills development of which they support. It negatively affects the proficiency the student is able to achieve in the end. Furthermore, we can see that in this limited time, the PPO is able to find the appropriate strategy to teach the student faster.

4 Summary

This paper investigated whether it is possible to apply RL methods to teach students on an e-learning platform using personalized, adaptive study plans. We have shown that algorithms we experimented on can learn development of which skill is supported by each task and how to select tasks appropriate for each student needs. Following that, reinforcement learning teacher successfully improved the student's proficiency. In reference to these results we demonstrated that reinforcement learning agents can support learning in e-learning platforms. As part of our future work, we plan to extend our approach using other algorithms (e.g. Deep-Q Network) or methods based on Monte Carlo Tree search. Moreover, we plan to extend the prepared environment by adding new student psychometric models like a Cognitive Diagnostic Models or Bayesian Knowledge Tracing.

Ack. Authors were supported by AI Tech project sponsored from POPC programme POPC.03.02.00-00-0001/20

References

1. Doroudi, S., Aleven, V., Brunskill, E.: Where's the reward? International Journal of Artificial Intelligence in Education **29** (2019)
2. Mnih, V., Badia, A.P., Mirza, M., et al.: Asynchronous methods for deep reinforcement learning. In: ICML. p. 1928–1937 (2016)
3. Rasch, G.: Probabilistic models for some intelligence and attainment tests. Danish Institute for Educational Research (1960)
4. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017), `http://arxiv.org/abs/1707.06347`