Reinforcement Learning for the Adaptive Scheduling of Educational Activities

Jonathan Bassen, Bharathan Balaji, Michael Schaarschmidt, Candace Thille, Jay Painter, Dawn Zimmaro, Alex Games, Ethan Fast, John C Mitchell

Best paper award - CHI 2020

Today's plan

1. Why ?
2. What has been done until now ?
3. What is proposed in this paper ?
4. How is the solution evaluated ?
5. What are the results ?

Motivation

Designing courses is not easy:

- What learning material do you present to students ?
- When ?
- How do you adapt it to each learner?

“Given a set of course materials, how can we assign each learner the smallest number of activities that maximize their learning gains?”

Related work

Adaptive scheduling requires:

Skill map
- Cognitive task analysis (CTA)
- Q-learning

&

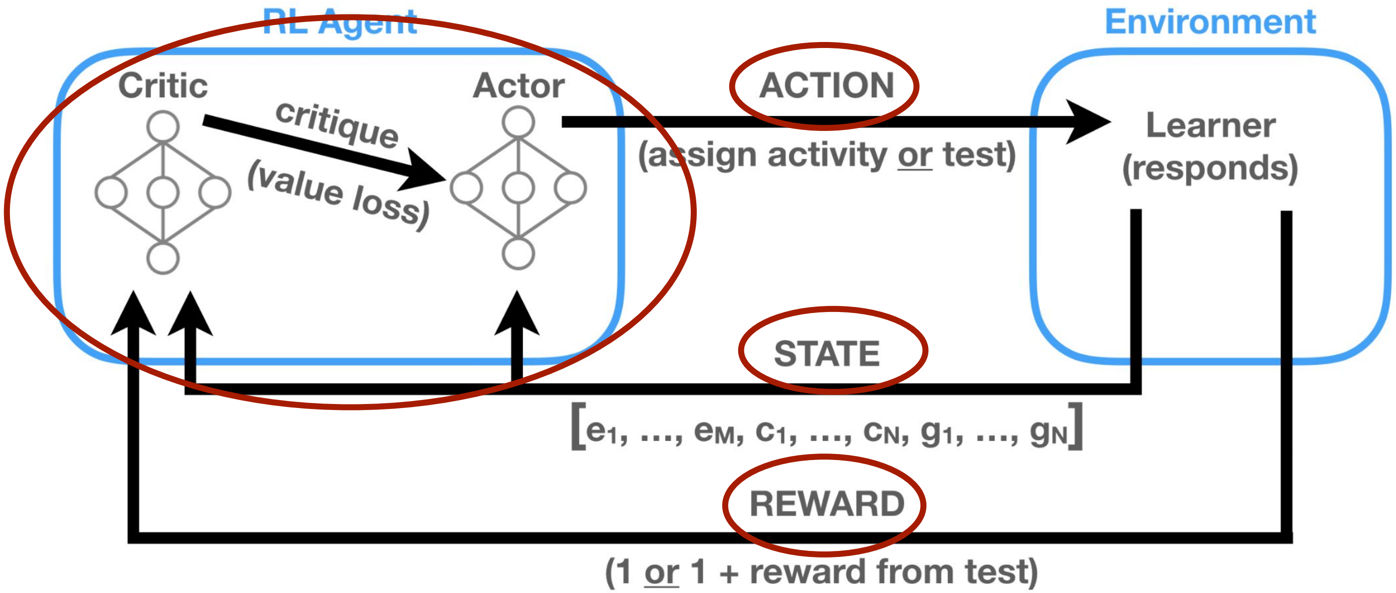
Learners knowledge models
- Bayesian Knowledge Tracing (BKT)
- IRT-Integrated Knowledge Tracing (IIKT)
- Deep Knowledge Tracing (DKT)

- Requires historical data
- Manual re-training
- Requires skills annotations

Reinforcement Learning is only used a little

Solution

Reinforcement Scheduling



Reinforcement Scheduling: *Action Space*

$$A_i = [\underbrace{a_1, \ldots, a_N}, \underbrace{a_{N+1}}]$$

- Educational activities
- Conditions: No repetition & at least one activity
- Always ends with post-test

Reinforcement Scheduling: *State Space*

$$S_i = [\underbrace{e_1, \ldots, e_M}, \underbrace{c_1, \ldots, c_N}, \underbrace{g_1, \ldots, g_N}]$$

- Pre-test scores
- Activities done (no matter the order)
- Scores of activities done (no matter the order)

Reinforcement Scheduling: *Reward Function*

$$R_i = \underline{1} \quad \text{[after educational activity is assigned]}$$

$$= \underline{1} + \sum_{p=1}^{M} \underline{\left[max(0, o_p - e_p) \right]} - \underline{(1+\psi) * H} \quad \text{[after post-test is assigned]}$$

Design choices:

1. Exploration of longer and diverse paths
2. Prevents assigning activities to preferentially
3. Ignore when initial guess or slip at post-test
4. Include only activities that significantly help

Reinforcement Scheduling: *Policy Optimization*

Model characteristics:

1. Advantage Actor-Critic architecture (both neural network)

Actor

The actor network learns π_θ with parameters θ, the mapping between a given state *s* and the probability of taking action *a*

Critic

The critic network learns V_ϕ with parameters ϕ, the mapping between the given state *s* and the reward R(s,a)

Advantage: relative benefit of taking action *a* $\longrightarrow$ $A(s_t, a_t) = R(s_t, a_t) + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$

Reinforcement Scheduling: *Policy Optimization*

Model characteristics:

1. Advantage Actor-Critic architecture (both neural network)
2. Proximal Policy Optimization

Actor

$$J_\theta = \mathbb{E}_t \left[min\left(\rho(\theta) A(s,a), \boxed{clip(\rho(\theta), 1-\varepsilon, 1+\varepsilon)} A(s,a)\right)\right]$$

$$\rho(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$$

Critic

$$L_\phi = \frac{1}{2} \sum_t V_\phi(s_t) - \left(R(s_t, a_t) + \gamma V_\phi(s_{t+1})\right)^2$$

The actor lags behind model update. Thus ρ(θ) and clipping reduce training instability and increase sample efficiency.

Evaluation

Course platform

- Supports different types of learning material
- Exposed an API for learner traces
- Can be set to allow different types of behavior (experimental conditions):
  - Reinforcement Scheduling (SD): one activity at a time selected by the model, not possible to choose activity or to go back
  - Linear Scheduling (LS): one activity at a time, all activities sequentially (predefined order)
  - Self-Directed (SD): one activity at a time selected by user, possible to go back and see multiple time same activity

Course overview

- Elementary linear algebra class, decomposed in three basic skills
- For each skill: video explanations, written descriptions, worked examples and assessment questions
- Each test problem and educational activity recorded a binary score of 1 or 0 after the learner responded to it
- 90min or less
- Available to Amazon employees in English-speaking region

Evaluation → pre-test and post-test with same 6 problems (2 per basic skill)

1987 enrollment splitted as follow:

- 95% Reinforcement Scheduling (1830 completed)
- 2.5% Linear Scheduling (91 completed)
- 2.5% Self-directed (66 completed)

Testing

Simulated learners

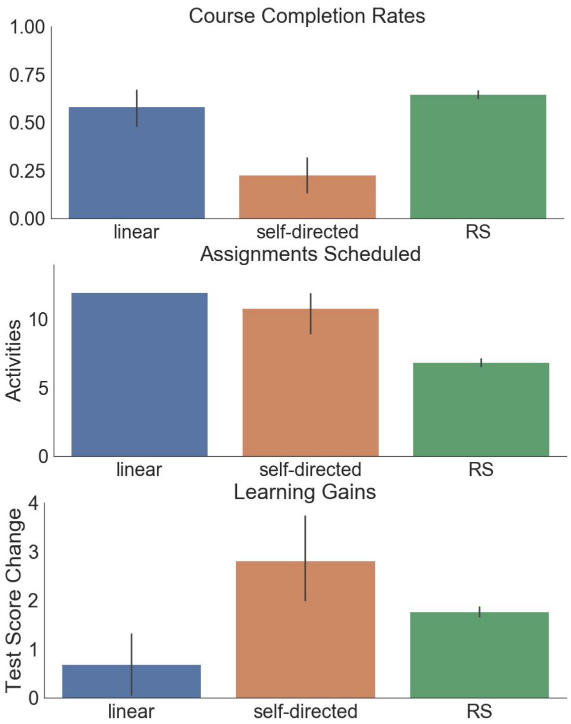
- BKT and IIKT
- Test model designs choices

Pilot study

- 24 learners from target population
- Test content and platform

Results

R1: How does reinforcement scheduling affect learning gains, the number of activities completed, and dropout?

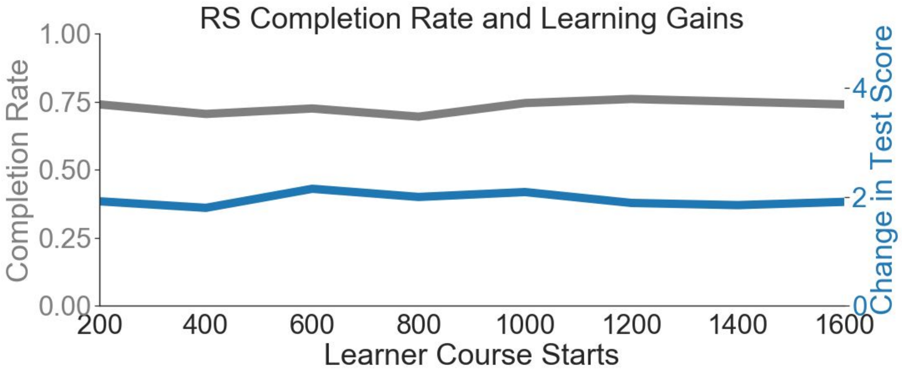


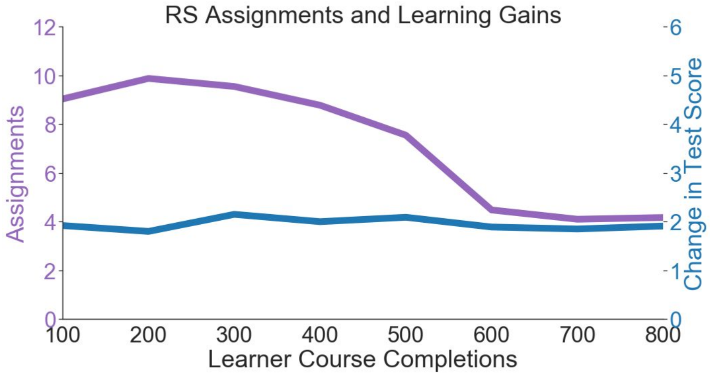
- LS & RS → higher course completion
- RS → less activities
- SD + RL → higher learning gain

  Due to loss of non motivated learners ?

Learners prefer not to choose activities and the number of activities doesn't seem to have a huge influence on learning gains

R2: Do early participants suffer from a worse assignment policy under reinforcement scheduling?

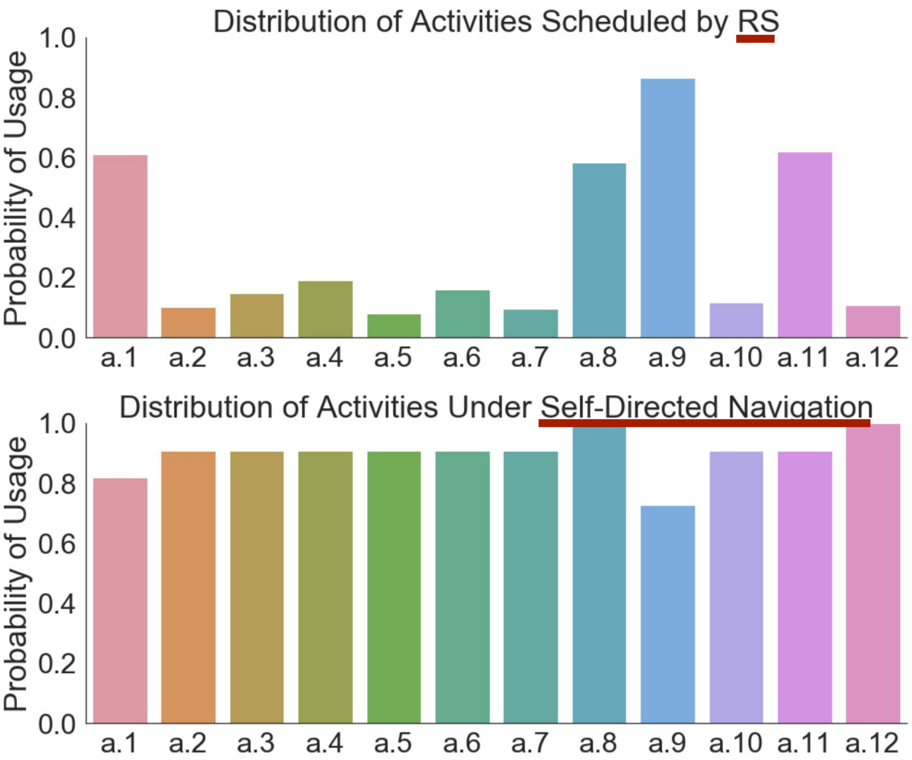




Almost constant completion rate and learning rate

- Delayed penalty for additional educational activities
- Positive immediate reward for assignments, encouraging agent's early exploration towards longer paths

R3: What can instructors and course designers learn from reinforcement scheduling?



Distribution of activities for the last 200 learners:

- mostly 4 items presented (covering the 3 elementary skill)
  → largest impact on score improvement

- less activities depending on user's pre-test score

R4: What are the qualitative experiences of learners under reinforcement scheduling?

Survey answers:

- 80% users satisfied
- Around 3.3/5 for effectiveness of ordering/selection/number of activities
- Overall liked the fact that it adapts to learners’ knowledge state
- Still sometimes some exploration from the model

And now ?

Generalization: complex evaluation ? less students ?

Data analysis: final policy ? study students' learning behavior ?

Reinforcement Scheduling itself: remove penalty for many activities ? cap assignments but allow repetitions ? penalize loss of learners ?

Conclusion

- Address the problem continuous improvement of online course scheduling
- Reinforcement Scheduling: Actor-critic architecture using proximal policy optimization
- Tested on a online learning course that they created, with around 2000 participants
- RS performed better while reducing the number of learning activities presented

Thank you for you attention

Questions ?