Empirically Evaluating the Effectiveness of POMDP vs. MDP Towards the Pedagogical Strategies Induction

Shitian Shen(✉), Behrooz Mostafavi(✉), Collin Lynch(✉), Tiffany Barnes(✉), and Min Chi(✉)

Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
{sshen,bzmostaf,cflynch,tmbarnes,mchi}@ncsu.edu

Abstract. The effectiveness of Intelligent Tutoring Systems (ITSs) often depends upon their *pedagogical strategies*, the policies used to decide what action to take next in the face of alternatives. We induce policies based on two general Reinforcement Learning (RL) frameworks: POMDP &. MDP, given the limited feature space. We conduct an empirical study where the RL-induced policies are compared against a random yet reasonable policy. Results show that when the contents are controlled to be equal, the MDP-based policy can improve students' learning significantly more than the random baseline while the POMDP-based policy cannot outperform the later. The possible reason is that the features selected for the MDP framework may not be the optimal feature space for POMDP.

Keywords: Reinforcement Learning · POMDP · MDP · ITS

1 Introduction

Reinforcement Learning (RL) offers one of the most promising approaches to applying data-driven decision-making to improve student learning in Intelligent Tutoring Systems (ITSs), which facilitates learning by providing step-by-step support and contextualized feedback to individual students [4,12]. These step-by-step behaviors can be viewed as a sequential decision process where at each step the system chooses an action (e.g. give a hint, show an example) from a set of options. *Pedagogical strategies* are policies that are used to decide what action to take next in the face of alternatives.

A number of researchers have applied RL to induce pedagogical policies for ITSs [2,3,5,8]: some apply Markov Decision Processes (MDPs) thus treating the user-system interactions as fully observable processes [6,11] while others utilize partially-observable MDPs (POMDPs) [9,13,14] to account for hidden states. In this work, we focus on comparing POMDPs vs. MDPs directly and induce the policies based upon these two frameworks given a small feature set. Besides, we

© Springer International Publishing AG, part of Springer Nature 2018
C. Penstein Rosé et al. (Eds.): AIED 2018, LNAI 10948, pp. 327–331, 2018.
https://doi.org/10.1007/978-3-319-93846-2_61

employ a simple baseline pedagogical policy where the system *randomly* decides whether to present the next problem as Worked Example (WE) or as Problem Solving (PS). Because both PS and WE are always considered to be *reasonable* educational intervention in our learning context, we refer to such policy as *random yet reasonable* policy or *random* in the following. The empirical result indicates that the RL-induced policies can improve students' learning significantly more than the random baseline for a particular type of students.

2 Methods

MDP is defined as a 4-tuple $\langle S, A, T, R\rangle$, where S denotes the observable state space, defined by a set of features that represent the interactive learning environment; A denotes the space of possible actions for the agent to execute; T represents the transition probability, and R represents expected reward of transiting from a state to another one by taking an action. In our work, the optimal policy π^* of an MDP is generated by Value Iteration algorithm.

POMDP is an extension of MDP, defined by a 7-tuple $\langle S,\ A,\ R,\ P_h,\ P_o,\ B,\ prior\rangle$, where A and R have the same definitions as in MDPs. S represents the *hidden* state space. P_h denotes the transition probability among the hidden state by taking the action, and P_o is the conditional observation probability. $Prior$ denotes the prior probability distribution of hidden states. B denotes the belief state space, which is constructed through Input-Output Hidden Markov Model (IOHMM) [1] in our work.

The POMDP policy induction procedure can be divided into three steps. First, we transform the training corpus into the hidden state space through the Viterbi algorithm. Second, we implement Q-learning to estimate the Q-values for each hidden state and action pair: $(s,\ a)$. Third, we estimate the Q value of belief state b and action a at time step t as:

$$Q_t(b, a) = \sum_s B_t(s) \cdot Q(s, a) \tag{1}$$

Thus, $Q_t(b, a)$ is a linear combination of the $Q(s, a)$ for each hidden state with its corresponding belief $B_t(s)$. When the process converges, π^* is induced by taking the optimal action a at time t associated with the highest $Q_t(b, a)$.

3 Experiment

Participates and Conditions. 124 undergraduate students who enrolled in Fall 2016 were randomly assigned to one of three conditions: MDP ($N = 45$), POMDP ($N = 40$), Random ($N = 39$). We subdivided the conditions into Fast ($n = 61$) and Slow ($n = 63$) groups based upon their average response time on Level 1. Combining conditions with Fast and Slow, we had a total of 6 groups: MDP-Fast ($N = 22$), MDP-Slow ($N = 23$), POMDP-Fast ($N = 18$), POMDP-Slow ($N = 22$), Random-Fast ($N = 21$), Random-Slow ($N = 18$).

The Chi-square test demonstrated that there was no significant difference on distribution of Fast vs. Slow among three conditions: $\chi^2 = 0.03, p = 0.86$.

Procedure. Deep Thought (DT) was a data-driven ITS that teaches logic proofs and it was used as part of an assignment in an undergraduate discrete mathematics course. DT consists of 6 strictly ordered levels of proof problems [7]. Students were required to complete 3–4 problems per level and a total of 18–24 problems overall. Students could skip problem if they encountered an issue to solve this problem. We treat level 1 as the pre-test phase to measure student's incoming confidence since students received the same problems in level 1 where all of the problems were PS. From level 2 to level 6, students were assigned a PS at the end of each level for evaluating student's performance fairly. Implemented policies made other decisions during the training process. ITS made total 10–15 decisions during a complete training process for each student.

Performance Evaluation. To fully evaluate student performance, we modified our in-class exam, referred as Post-test. Students' answers were graded to the scale of 1–100 by the Teaching Assistants of the class (who are not part of the research group). We mainly treated the Post-test score as Students' learning outcome measure in the following.

Training Data was collected in the Fall 2014 and Spring 2015 semesters. All of the students used the same ITS, followed the same general procedure, studied the same training materials, and worked through the same set of training problems. The only substantive difference was the presentation of the materials, WE or PS, randomly decided. The training dataset contained the interaction logs of 306 students and the average number of problems solved by students was 23.7 and the average time that students spent in the tutor was 5.29 h. There are a total of 133 features to represent students' behaviors. We generate the same feature space for both MDP and POMDP through a MDP-based feature selection approach [10], which selects a total of six features, shown as follows:

1. **totalPSTime**: total time that students spend on PS.
2. **easyProbCount**: easy problem that students solved so far.
3. **newLevel**: whether students jump into a new level.
4. **avgStepTime**: average step time so far.
5. **hintRatio**: the ratio between hint count and number of applying rules.
6. **numProbRule**: number of rules in the current problem's solution.

4 Results

Pre-test Score. A two-way ANOVA using condition {MDP, POMDP, Random} and type {Fast, Slow} as factors, shows that there is no significant interaction effect with the students' pre-test scores. Additionally, a one-way ANOVA indicates that there is no significant difference in the pre-test scores among the three conditions, or between the Fast and Slow groups. Therefore, we can conclude that all of the six groups have a similar incoming competence. Table 1 presents the mean and (SD) of pre- and post-test score for each group.

Post-test Score. A two-way ANCOVA, using condition and type as factors and pre-test as the covariate, shows a significant interaction effect on the post-test score: $F(2, 117) = 4.06, p = .019$. Additionally, one-way ANCOVA tests show that there is no significant difference either among conditions or between Fast and Slow. Furthermore, one-way ANCOVA tests on policy using pre-test as the covariate shows no significant difference among the three Fast groups on the post-test score: $F(2, 57) = 0.74, p = 0.48$, but the significant difference among the three Slow groups: $F(2, 59) = 5.03, p = .009$. Specifically, pairwise t-tests indicate that *MDP-Slow* scored significantly higher post-test than both *POMDP-Slow* and *Random-Slow*: $p = .004$ and $p = .015$ respectively, and no significant difference is found between the latter two groups. Therefore, our results exhibited an Aptitude-Treatment Interaction effect: all of three Fast groups learned equally well after training on ITS regardless of the policies employed while the Slow groups were indeed more sensitive to induced policies. For Slow groups, the MDP policy significantly outperformed the POMDP and Random policies while no significant difference existed between the latter two policies.

Table 1. Pre- and Post-test scores for each group

Policy	Pre-test score			Post-test score		
	Total	Fast	Slow	Total	Fast	Slow
MDP	74.90 (26.3)	75.34 (27.6)	74.48 (25.5)	88.26 (15.2)	84.23 (17.7)	**92.12 (11.3)**
POMDP	75.18 (25.9)	74.01 (29.1)	76.15 (23.2)	79.53 (24.4)	86.47 (23.6)	73.86 (24.1)
Random	65.99 (28.1)	67.69 (28.8)	64.02 (27.8)	82.85 (22.3)	88.98 (17.9)	75.69 (25.3)

Furthermore, we compared the Fast and Slow groups within each condition. Two-sample t-tests shows no significant difference between Fast and Slow under the POMDP condition: $t(38) = 1.67$, $p = 0.11$, but the marginal significant difference between Fast and Slow under either MDP or Random condition: $t(43) = -1.78, p = .081$ and $t(37) = 1.91, p = .063$ respectively.

5 Conclusions and Future Work

In this study, we induced two types of RL policies using MDP and POMDP framework respectively and compared their effectiveness against the random baseline in the context of ITS. Besides, we split students into Fast and Slow groups based on their average step time in the initial tutorial level. The empirical results exhibited an Aptitude-Treatment interaction effect: Fast groups were less sensitive to the policies in that they learned equally well regardless of the policies while the Slow groups were more sensitive in that the MDP policy could help slow groups score significantly higher post-test than the POMDP and Random policies. This suggested that the MDP policy is more effective than either POMDP or Random policy for Slow groups. One of the possible reasons for the

ineffectiveness of the POMDP policy is that the feature selection and discretization limit the full power of the POMDP framework. In future work, we plan to maintain the continuous features and design effective feature extraction method for POMDP in order to show the full power of POMDP.

Acknowledgements. This research was supported by the NSF Grants #1726550, #1651909, and #1432156.

References

1. Bengio, Y., Frasconi, P.: An input output HMM architecture. In: Advances in Neural Information Processing Systems, pp. 427–434 (1995)
2. Chi, M., VanLehn, K., Litman, D., Jordan, P.: Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. User Model. User Adap. Inter. **21**(1–2), 137–180 (2011)
3. Doroudi, S., Holstein, K., Aleven, V., Brunskill, E.: Towards understanding how to leverage sense-making, induction and refinement, and fluency to improve robust learning. In: International Educational Data Mining Society (2015)
4. Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A.: Intelligent tutoring goes to school in the big city (1997)
5. Koedinger, K.R., Brunskill, E., Baker, R.S., McLaughlin, E.A., Stamper, J.: New potentials for data-driven intelligent tutoring system development and optimization. AI Mag. **34**(3), 27–41 (2013)
6. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. IEEE Trans. Speech Audio Process. **8**(1), 11–23 (2000)
7. Mostafavi Behrooz, Z.L., Barnes, T.: Data-driven proficiency profiling. In: Proceedings of the 8th International Conference on Educational Data Mining (2015)
8. Rowe, J.P., Lester, J.C.: Improving student problem solving in narrative-centered learning environments: a modular reinforcement learning framework. In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS (LNAI), vol. 9112, pp. 419–428. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19773-9_42
9. Roy, N., Pineau, J., Thrun, S.: Spoken dialogue management using probabilistic reasoning. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp. 93–100. Association for Computational Linguistics (2000)
10. Shen, S., Chi, M.: Aim low: correlation-based feature selection for model-based reinforcement learning. In: EDM, pp. 507–512 (2016)
11. Singh, S., Litman, D., Kearns, M., Walker, M.: Optimizing dialogue management with reinforcement learning: experiments with the NJFun system. J. Artif. Intell. Res. **16**, 105–133 (2002)
12. Vanlehn, K.: The behavior of tutoring systems. Int. J. Artif. Intell. Educ. **16**(3), 227–265 (2006)
13. Williams, J.D., Young, S.: Partially observable Markov decision processes for spoken dialog systems. Comput. Speech Lang. **21**(2), 393–422 (2007)
14. Zhang, B., Cai, Q., Mao, J., Chang, E., Guo, B.: Spoken dialogue management as planning and acting under uncertainty. In: INTERSPEECH, pp. 2169–2172 (2001)