# Exploring Induced Pedagogical Strategies Through a Markov Decision Process Framework: Lessons Learned

Shitian Shen
North Carolina State University
sshen@ncsu.edu

Behrooz Mostafavi
North Carolina State University
bzmostaf@ncsu.edu

Tiffany Barnes
North Carolina State University
tmbarnes@ncsu.edu

Min Chi
North Carolina State University
mchi@ncsu.edu

An important goal in the design and development of Intelligent Tutoring Systems (ITSs) is to have a system that adaptively reacts to students' behavior in the short term and effectively improves their learning performance in the long term. Inducing effective pedagogical strategies that accomplish this goal is an essential challenge. To address this challenge, we explore three aspects of a Markov Decision Process (MDP) framework through four experiments. The three aspects are: 1) *reward function*, detecting the impact of immediate and delayed reward on effectiveness of the policies; 2) *state representation*, exploring ECR-based, correlation-based, and ensemble feature selection approaches for representing the MDP state space; and 3) *policy execution*, investigating the effectiveness of stochastic and deterministic policy executions on learning. The most important result of this work is that there exists an aptitude-treatment interaction (ATI) effect in our experiments: the policies have significantly different impacts on the particular types of students as opposed to the entire population. We refer the students who are sensitive to the policies as the Responsive group. All our following results are based on the Responsive group. First, we find that an immediate reward can facilitate a more effective induced policy than a delayed reward. Second, The MDP policies induced based on low correlation-based and ensemble feature selection approaches are more effective than a Random yet reasonable policy. Third, no significant improvement was found using stochastic policy execution due to a ceiling effect.

**Keywords:** reinforcement learning, intelligent tutoring systems, problem solving, worked example, pedagogical strategy

## 1. Introduction

In general, the effectiveness of an intelligent tutoring system (ITS) can greatly depend on the implemented *pedagogical strategy*, which decides the next action for the tutor to take at each step of a student's learning process among a set of alternatives. Using pedagogical strategies, ITSs are able to adaptively interact with students by taking different actions given various situations in the short term in order to improve their learning performance in the long term. However, inducing pedagogical strategies in an ITS is challenging. On one hand, the relation between a tutor's decisions and eventual outcomes cannot be immediately observed. On the other hand, each decision affects the student's subsequent actions and performance, which also has an impact on the tutor's next decision. Therefore, the effectiveness of the current decision depends upon the effectiveness of subsequent decisions, and this iterative process cannot be easily solved by directly optimizing an objective function. Similar to prior work (Chi et al., 2011; Tetreault et al., 2007), we apply a classic type of reinforcement learning (RL) framework, the Markov Decision Process (MDP), to address this challenge. In this work, we report our results from exploring the MDP framework from three aspects including *reward function*, *state representation*, and *policy execution*.

**Reward Function**. Real-world RL applications often contain two types of rewards: immediate reward, which is the immediate feedback after taking an action, and delayed reward, which is the reward received later after taking more than one action. The longer rewards are delayed, the harder it becomes to assign credit or blame to particular actions or decisions. Moreover, learning short-term performance boosts may not result in long-term learning gains. Thus, in this work we explore both immediate and delayed rewards in our policy induction, and empirically evaluate the impact of the induced policies on student learning. Our results show that using immediate rewards can be more effective than using delayed rewards.

**State Representation**. For RL, as with all machine learning tasks, success depends upon an effective state representation. When a student interacts with an ITS, there are many factors that might determine whether the student learns well from the ITS, yet many other factors are not well understood. To make the RL problem tractable, our approach is to begin with a large set of features to which a series of feature-selection methods are applied to reduce them to a tractable subset. In this work, we apply a series of correlation-based feature selection methods to RL: first we explored the option of selecting the next feature that is the *most correlated (High option)* to the currently selected feature set and then the option of selecting the *least correlated (Low option)*. In general, the features that are most highly correlated with output labels are selected in supervised learning (Yang and Pedersen, 1997; Lee and Lee, 2006; Chandrashekar and Sahin, 2014; Koprinska et al., 2015). Since output labels are not present in reinforcement learning, we use correlation to the current feature set as a best approximation. Section 5.6 shows that the high-correlation option indeed outperformed two baseline methods: the random baseline and also the best feature selection explored in our previous work (Chi et al., 2011). However, for our dataset, the high correlation-selected features tend to be homogeneous. Different from the supervised learning tasks, we hypothesize that it is more important to have heterogeneous features in RL that can grasp different aspects of learning environments. Therefore, we also explore the low correlation-based option for feature selection with a goal to increase the diversity of the selected feature set. To do so, we select the next feature that is the least correlated with the currently selected feature set and extend the feature set. Our results show that the low correlation-based option outperformed not only the high option but also the other two baselines.

**Policy Execution**. In most of the prior work with RL in ITSs, deterministic policy execution is used. That is, when evaluating the effectiveness of RL-induced policies, the system would strictly carry out the actions determined by the policies. In this work, we explore *stochastic* policy execution. We argue that stochastic execution can be more effective than deterministic execution because if the RL induced policy is sub-optimal, under the stochastic policy execution, it would still be possible for the system to carry out the optimal action; whereas if the induced policy is indeed optimal, our approach will make sure that when the decisions are crucial, the stochastic policy execution would behave like deterministic policy execution in that the optimal action will be carried out (see section 6.7 for details). We empirically evaluate the effectiveness of the stochastic policy execution but our results show that there is a ceiling effect.

Generally speaking, ITSs contain different types of tutorial decisions including *what* material to teach and *how* to teach it. In our work, we focus on applying RL to induce policy on the how part. To do so, we controlled the content (the *what* part) to be the same across all students and we mainly focus on one type of tutorial decision: whether to present a given problem as a *problem solving (PS)* or a *worked example (WE)*. When providing a WE, the tutor presents an expert solution to a problem step-by-step so that the student sees both the answer and the solution procedure. During PS, students are required to complete the problem with tutor support. In a series of empirical experiments described below, we compare the RL-induced policies against a policy where the system *randomly* decides whether to present the next problem as WE or as PS. Because both PS and WE are always considered to be *reasonable* educational interventions in our learning context, we refer to such policy as a *random yet reasonable* policy or *random*.

Our results consistently suggest that there is an **aptitude-treatment interaction (ATI)** effect (Cronbach and Snow, 1977; Snow, 1991): certain students are less sensitive to the induced policies in that they achieve a similar learning performance regardless of policies employed, whereas other students are more sensitive in that their learning is highly dependant on the effectiveness of the policies. In the following, we refer to the former group as *Unresponsive* and latter group as *Responsive* respectively.

In short, we extensively explore applying the MDP framework for pedagogical policy induction on WE vs. PS and conduct extensive empirical experiments for investigating the effectiveness of induced RL policies. The effectiveness of the policies is evaluated by using students' in-class exam scores, referred to as *transfer post-test score*. Overall, our main contributions are summarized as follows:

- We found a consistent aptitude-treatment interaction effect across experiments: the performance of the Responsive group is significantly dependent on the implemented policies, whereas the Unresponsive group performs similarly regardless of policies.
- We induce RL policies based on immediate and delayed rewards respectively and detect the impact of reward on the effectiveness of policies.
- We propose correlation-based and ensemble feature selection approaches for state representation in the MDP framework and then empirically evaluate the RL-induced policies.
- We explore executing policies stochastically in contrast to previous research which mainly evaluates the RL-induced policies deterministically.

The rest of the paper is arranged as follows: Section 2 presents an overview of related work. Section 3 describes the reinforcement learning framework and Markov Decision Process. Section 4 describes the tutorial decisions, Deep Thought tutor, our training data, and state representation. Section 5 describes five correlation metrics and then introduces our proposed feature selection methods. Section 6 presents the overview of our four empirical studies and research questions. Section 7 reports experimental results for each of the four experiments. Section 8 presents our post-hoc comparison results. Finally, we summarize our conclusions, limitations and future work in Section 9.

# 2. Related Work

## 2.1. Reinforcement Learning Applications in Education Domains

**Markov Decision Process** (MDP; Littman 1994; Sutton and Barto 1998) is a widely used reinforcement learning framework in educational applications. Beck et al. (2000) investigated temporal difference learning to induce pedagogical policies that would minimize the time students spend on completing problems in AnimalWatch, an ITS that teaches arithmetic to grade school students. They used simulated students in the training phase of their study and used time as an immediate reward given that student's time can be assessed at each step. In the test phase, the new AnimalWatch with induced pedagogical policy was empirically compared with the original version. They found that the policy group spent significantly less time per problem than their non-policy peers.

Iglesias and her colleagues applied online Q-learning with time as the immediate reward to generate a policy in RLATES, an intelligent educational system that teaches students database design (Iglesias et al., 2009a; Iglesias et al., 2009b; Iglesias et al., 2003). The goal of inducing the policy was to provide students with direct navigation support through the system's content and to help them learn more efficiently. They also used simulated students in the training phase and evaluated the induced policy by comparing the performance of both simulated and real students using RLATES with that of other students using IGNATES, which provided indirect navigation support without RL. Their results showed that students using RLATES spent significantly less time than students using IGNATES, but there was no significant difference in students' level of knowledge evaluated by the exam.

Martin and Arroyo (2004) applied a model-based RL method with delayed reward to induce policies that would increase the efficiency of hint sequencing on Wayang Outpost, a web-based ITS. During the training phase, the authors used a student model to generate training data for inducing the policies. In the test phase, the induced RL policies were tested on a simulated student model and students' performance was evaluated by learning level, a customized score function. The results showed that students following RL policies achieved a significantly better learning level than the non-policy group.

Additionally, Chi et al. (2011) applied a model-based RL method with delayed reward to induce pedagogical policies to improve the effectiveness of Cordillera, an intelligent natural language tutoring system that teaches students college physics. They collected an exploratory corpus by training human students on a version of the ITS that made random decisions. Their empirical evaluation showed the induced policies were significantly more effective than the previous policies based on students' normalized learning gain (NLG).

In short, most prior work on the application of MDP to ITSs has found no significant learning differences between the induced RL policies and baseline random policies. One potential explanation for this is that MDP relies on a small set of pre-defined state representations, which may not fully represent the real interactive learning environments.

**Partially observable Markov Decision Process** (POMDP; Jaakkola et al. 1995; Koenig and Simmons 1998) is another widely used framework in educational domains. Different from the MDP framework where the state space is constructed by a set of observable features, the POMDP framework uses a belief state space to model the unobserved factors, such as students' knowledge level and proficiency. Mandel et al. (2014) combined a feature compression approach that can handle a large range of state features with POMDP to induce policies for an educational game. The induced policies with the immediate reward outperformed both random and expert-designed policies in both simulated and empirical evaluations.

Rafferty et al. (2016) applied POMDP to represent students' latent knowledge by combining embedded graphical models for concept learning with interpreted belief states in the domain of alphabet arithmetic. They applied POMDP to induce policies using time as the reward with a goal of reducing the expected time for learners to comprehend concepts. They evaluated policies using simulated and real-world studies and found that the POMDP-based policies significantly outperformed a random policy.

Clement et al. (2016) constructed models to track students' individual mastery of each knowledge component. They combined POMDP with the student models to induce teaching policies using learning gain as the immediate reward. The results of a series of simulated studies showed that the POMDP policies outperformed the learning theory-based policies in terms of students' knowledge levels. Similarly, Whitehill and Movellan (2018) implemented POMDP to induce a teaching policy with the purpose of minimizing the expected time for foreign language learning in their ITS. The belief state of their POMDP was constructed based on a modified student model which hypothesized that students cannot always fully absorb the examples and so only partially update their belief state. They conducted a real-world study and verified that the POMDP policy performed favorably compared to two hand-crafted teaching policies.

**Deep RL Framework** is a subject of growing interest in inducing policies. Deep RL adds deep neural networks to RL frameworks such as POMDP for function approximation or state approximation (Mnih et al., 2013; Mnih et al., 2015). This enhancement makes the agent capable of achieving complicated tasks. Wang et al. (2017) applied a deep RL framework for personalizing interactive narratives in an educational game called CRYSTAL ISLAND. They designed the immediate rewards based on normalized learning gain (NLG) and found that the students with the Deep RL policy achieved a higher NLG score than those following the linear RL model in simulation studies. Furthermore, Narasimhan et al. (2015) implemented a Deep Q-Network (DQN) approach in text-based strategy games, constructed based on Evennia, which is an open-source library and toolkit for building multi-users online text-based games. In the DQN method, the state is represented by a Long Short-Term Memory network, the Q-value is approximated by a multi-layered neural network, and the immediate reward is designed based on the performance in the game. Using simulations, they found that the deep RL policy significantly outperformed the random policy in terms of quest completion.

Table 1 summarizes the related work about the application of RL in the educational domain. While both POMDP and Deep RL have been shown to be highly effective in many real-world applications, they generally require a great deal of training data, especially Deep RL. More importantly, it is often hard to interpret the induced POMDP and Deep RL policies. Therefore, in this paper, we mainly focus on exploring the MDP framework, especially the tabular MDP framework. Compared with previous research, we explore three aspects of the MDP framework and evaluate the effectiveness of induced policies using a series of experiments conducted in real classroom settings.

### 2.2. APTITUDE-TREATMENT INTERACTION (ATI) EFFECT

Previous work shows that the ATI effect commonly exists in many real-world studies. More formally, the ATI effect states that instructional treatments are more or less effective to individual

Table 1: Reinforcement Learning Applications in Educational Domain

| Framework | Work | Reward | Experiment | Evaluation |
|---|---|---|---|---|
| MDP | Beck et al. (2000) | Immediate | Simulation | Time |
| | Iglesias et al. (2009a) | Immediate | Simulated & Real | Time & Perform |
| | Martin and Arroyo (2004) | Delay | Simulation | Perform |
| | Chi et al. (2011) | Delay | Laboratory | Perform |
| POMDP | Mandel et al. (2014) | Immediate | Simulated & Real | Performance |
| | Rafferty et al. (2016) | Immediate | Simulated & Real | Time |
| | Clement et al. (2016) | Immediate | Simulation | Performance |
| | Whitehill and Movellan (2018) | Immediate | Real | Time |
| Deep RL | Wang et al. (2017) | Immediate | Simulation | Performance |
| | Narasimhan et al. (2015) | Immediate | Real | Performance |

learners depending on their abilities (Cronbach and Snow, 1977). For example, Kalyuga et al. (2003) empirically evaluated the effectiveness of worked example (WE) vs. problem solving (PS) on student learning in programmable logic. Their results show that WE is more effective for inexperienced students while PS is more effective for experienced learners.

Moreover, D'Mello et al. (2010) compared two versions of ITSs: one is an affect-sensitive tutor which selects the next problem based on students' affective and cognitive states combined, while the other is an original tutor which selects the next problem based on students' cognitive states alone. An empirical study shows that there is no significant difference between the two tutors for students with high prior knowledge. However, there is a significant difference for students with low prior knowledge: those who trained on the affect-sensitive tutor had significantly higher learning gain than their peers using the original tutor.

Chi and VanLehn (2010) investigated the ATI effect in the domain of probability and physics, and their results showed that high competence students can learn regardless of instructional interventions, while for students with low competence, those who follow the effective instructional interventions learned significantly more than those who did not. Our prior work consistently finds that for pedagogical decisions on WE vs. PS, certain learners are always less sensitive in that their learning is not affected, while others are more sensitive to variations in different policies. For example, Shen and Chi (2016) trained students in an ITS for logic proofs, then divided students into the Fast and Slow groups based on time, and found that the Slow groups are more sensitive to the pedagogical strategies while the Fast groups are less sensitive.

### 2.3. Pedagogical Decisions: Worked Examples vs. Problem Solving

In this study, we investigate RL-induced pedagogical strategies on one type of tutorial decision: worked examples (WE) vs. problem solving (PS). A great deal of research has investigated the impacts of WE and PS on student learning (McLaren and Isotani, 2011; McLaren et al., 2014; Najar et al., 2014; Salden et al., 2010). During PS, students are given a training problem which they must solve independently or with partial assistance, while during WE, students are shown a detailed solution to the problem.

McLaren et al. (2008) compared WE-PS pairs with PS-only, where every student was given the same 10 training problems. Students in the PS-only condition were required to solve every problem while students in the WE-PS condition were given 5 example-problem pairs. Each pair consists of an initial worked example problem followed by tutored problem solving. They found no significant difference in learning performance between the two conditions; however, the WE-PS group spent significantly less time on task than the PS group.

McLaren and Isotani (2011) found similar results in two subsequent studies, which compared learning gains and time on task for high school chemistry students given 10 identical problems in three conditions: WE, PS, and WE-PS pairs. There were no significant differences among the three groups in terms of learning gains, but the WE group spent significantly less time on task than the other two conditions, and no significant time on task difference was found between the PS and WE-PS conditions. A follow-up 2014 study compared four conditions: WE, tutored PS, untutored PS, and Erroneous Examples (EE) in high school stoichiometry (McLaren et al., 2014). Students in the EE condition were given *incorrect* worked examples containing between 1 and 4 errors and were tasked with correcting them. Again the authors found no significant differences among the conditions in terms of learning gains, and as before the WE students spent significantly less time than the other groups. More specifically, for time on task they found that: $WE < EE < untutored\ PS < tutored\ PS$. WE students took only 30% of the total time of the tutored PS students.

The advantages of WE were also demonstrated in another study in the domain of electrical circuits (Van Gog et al., 2011). In that study, they compared four conditions: WE, WE-PS pairs, PS-WE pairs (problem-solving followed by an example problem), and PS only. Their results showed that the WE and WE-PS students significantly outperformed the other two groups, and no significant difference was found among four conditions in terms of time on task. Additionally, Razzaq and Heffernan (2009) designed an experiment on comparing worked examples vs. problem solving in an ITS that teaches mathematics. They found that more proficient students benefit more from WE when controlling for time, while less proficient students benefit more from PS.

Some existing theories of learning suggest that when deciding whether to present PS or WE, a tutor should take into account several factors, including the students' current knowledge model. Vygotsky (1978) coined the term "zone of proximal development" (ZPD) to describe the space between abilities that students may display independently and abilities that they may display with support. He hypothesized that the most learning occurs when students are assigned tasks within their ZPD. In other words the task should neither be so simple that they can achieve it independently or trivially, nor so difficult that they simply cannot make progress even with assistance. Based upon this theory, we expect that if students are somewhat competent in all the knowledge needed for solving a problem, the tutor should present the problem as a PS, and provide help only if the students fail so that they can practice their knowledge. However, if students are completely unfamiliar with the problem, then the tutor should present the problem as a WE. Brown et al. (1989) describe a progression from WE to PS following their "model, scaffold & fade" rubric. Koedinger and Aleven (2007) by contrast defined an "assistance dimension", which includes PSs and WEs. The level of assistance a tutor should provide may be resolved differently for different students and should be adaptive to the learning environment, the domain materials used, the students' knowledge level, their affect state and so on. Typically, these theories are considerably more general than the specific decisions that ITS designers must make, which makes it difficult to tell if a specific pedagogical strategy is consistent with the theory. This is why we hope to derive pedagogical policy for PS/WE decision making directly from empirical data.

Finally, compared with all previous studies in which the PSs and WEs are generally designed by domain experts or expert-like system developers, in this work both PSs and WEs are constructed through a *data-driven* approach using previous students log files (more details in section 4). In short, prior research on WE and PS has shown that WE can be as or more effective than PS or alternating PS with WE, and the former can take significantly less time than the latter two (McLaren et al., 2014; Renkl et al., 2002; McLaren and Isotani, 2011; Mostafavi et al., 2015). As opposed to previous work, which involves the hard-coded rules for providing PS or WE, we induce a data-driven pedagogical strategy which explicitly indicates how to make decisions given the current state of students and the learning context.

## 3. Reinforcement Learning and the MDP Framework

The Markov Decision Process (MDP) is one of the most widely used RL frameworks. In general, an 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; and $T$ represents the transition probability where $p(s, a, s')$ is the probability of transiting from state $s$ to state $s'$ by taking action $a$. Finally, the reward function $R$ represents the immediate or delayed feedback where $r(s, a, s')$ denotes the expected reward of transitioning from state $s$ to state $s'$ by taking action $a$. Since we apply the tabular MDP framework, the reward function ***R*** and transition probability table ***T*** can be easily estimated from the training corpus. The goal of MDP is to generate the deterministic policy $\pi : s \rightarrow a$ that maps each state onto an action.

Once the tuple $\langle S, A, T, R\rangle$ is set, the optimal policy $\pi^*$ for an MDP can be generated via dynamic programming approaches, such as Value Iteration. This algorithm operates by finding the optimal value for each state $V^*(s)$, which is the expected discounted reward that the agent will gain if it starts in $s$ and follows the optimal policy to the goal. Generally speaking, $V^*(s)$ can be obtained by the optimal value function for each state-action pair $Q^*(s, a)$ which is defined as the expected discounted reward the agent will gain if it takes an action $a$ in a state $s$ and follows the optimal policy to the end. The optimal state value $V^*(s)$ and value function $Q^*(s, a)$ can be obtained by iteratively updating $V(s)$ and $Q(s, a)$ via equations 1 and 2 until they converge:

$$Q(s,a) \quad := \quad \sum_{s'} p(s,a,s')\left[r(s,a,s') + \gamma V_{t-1}(s')\right] \tag{1}$$

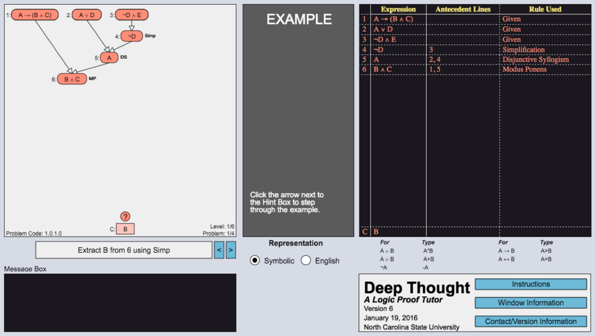
$$V(s) \quad := \quad \max_a Q(s,a) \tag{2}$$

where $0 \leq \gamma < 1$ is a discount factor. When the process converges, the optimal policy $\pi^*$ can be induced corresponding to the optimal Q-value function $Q^*(s, a)$, represented as:

$$\pi^*(s) = argmax_a Q^*(s,a) \tag{3}$$

where $\pi^*$ is the deterministic policy that maps a given state into an action. In the context of an ITS, this induced policy represents the pedagogical strategy by specifying tutorial actions using the current state.

In the present work, the effectiveness of the MDP policy is estimated by Expected Cumulative Reward (ECR; Tetreault and Litman 2008; Chi et al. 2011). The ECR of a policy $\pi$ is



Problem Code: 1.0.1.0

C: B

Level: 1/6
Problem: 1/4

Click the arrow next to the Hint Box to step through the example.

|  | Expression | Antecedent Lines | Rule Used |
|---|---|---|---|
| 1 | A → (B ∧ C) |  | Given |
| 2 | A ∨ D |  | Given |
| 3 | ¬D ∧ E |  | Given |
| 4 | ¬D | 3 | Simplification |
| 5 | A | 2, 4 | Disjunctive Syllogism |
| 6 | B ∧ C | 1, 5 | Modus Ponens |
| C | B |  |  |

Extract B from 6 using Simp

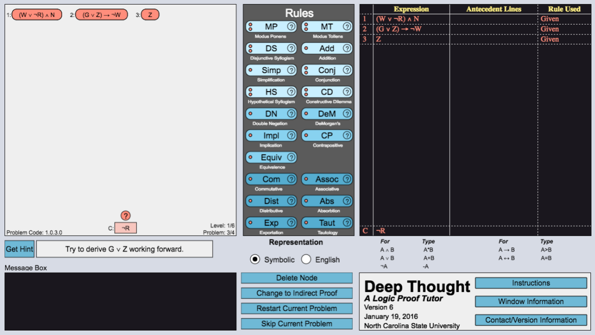
● Symbolic ○ English

| For | Type | For | Type |
|---|---|---|---|
| A ∧ B | A*B | A → B | A>B |
| A ∨ B | A+B | A ↔ B | A=B |
| ¬A | -A |  |  |

Message Box

Deep Thought
*A Logic Proof Tutor*
Version 6
January 19, 2016
North Carolina State University

Figure 1: Interface for Worked Example



Problem Code: 1.0.3.0

C: ¬R

Level: 1/6
Problem: 3/4

|  | Expression | Antecedent Lines | Rule Used |
|---|---|---|---|
| 1 | (W ∨ ¬R) ∧ N |  | Given |
| 2 | (G ∨ Z) → ¬W |  | Given |
| 3 | Z |  | Given |
| C | ¬R |  |  |

Try to derive G ∨ Z working forward.

● Symbolic ○ English

| For | Type | For | Type |
|---|---|---|---|
| A ∧ B | A*B | A → B | A>B |
| A ∨ B | A+B | A ↔ B | A=B |
| ¬A | -A |  |  |

Message Box

Deep Thought
*A Logic Proof Tutor*
Version 6
January 19, 2016
North Carolina State University

Figure 2: Interface for Problem Solving

calculated by the average over the value function of initial states. It's defined as:

$$ECR(\pi) = \sum_i \frac{N_i}{N} \times V^{\pi}(S_i) \tag{4}$$

Where $N$ denotes the number of trajectories in the training corpus, i.e. the total number of the initial states; and $N_i$ denotes the number of states $S_i$ as the initial states in the training corpus. In our case, the trajectories have a finite time horizon. Thus, ECR evaluates the expected reward of the initial states. The higher the ECR value of a policy, the better the policy is expected to perform.

## 4. Pedagogical Decisions in a Logic Tutor: Deep Thought

### 4.1. Overview of Deep Thought

**Deep Thought** (DT) is a data-driven ITS used in the undergraduate-level Discrete Mathematics course at North Carolina State University (Behrooz and Tiffany, 2017). DT provides students with a graph-based representation of logic proofs which allows students to solve problems by applying logic rules to derive new logical statements, represented as nodes. The system automatically verifies proofs and provides immediate feedback on rule application (but not strategy) errors. Every problem in DT can be presented in the form of either a worked example (WE) or problem solving (PS). In WE (shown in Figure 1), students are given a detailed example showing the expert solution for the problem or were shown the best next step to take given their current solution state. In PS (shown in Figure 2), by contrast, students are tasked with solving the same problem using the ITS or completing an individual problem-solving step. Focusing on the pedagogical decisions of choosing WE vs. PS allows us to strictly control the content to be *equivalent* for all students.

All of the hints that students receive for PS in DT are data-driven. Specifically, next-step hints for a PS are constructed by using previous successful student solutions which include the current proof state, and by matching current expressions in the proof. The hint presented at the current proof state guides the student to the most frequent next step that had resulted in successful completion of the proof given that proof state (Stamper et al., 2013), and is given to the student below the proof construction window on the left hand side of the tutor (shown in Figure 2). The hints are in the format of "Use expression X and expression Y to derive expression Z using rule R". Students are given the opportunity to request hints on-demand by clicking the "Get Hint" button next to the dialogue box; however, if students stay in the current proof state for longer than the median step time of that problem or a maximum of 30 seconds, DT automatically presents the available hint. The WEs were constructed in a similar manner, where the most efficient (shortest-path) solution of the current proof from previous student solutions was used for a step-by-step presentation of the proof with procedurally constructed instructions given to the student below the proof window (Figure 2). At each step, the instructions for constructing the next step are presented in the same format as the next-step hints until the conclusion is reached.

The problems in DT are organized into six strictly ordered levels with 3–4 problems per level. Level 1 functions as a ***pre-test*** in that all participants receive the same set of PS problems. In the five training levels 2–6, before the students proceed to a new problem, the system follows the corresponding RL-induced or random policies to decide whether to present the problem as PS or WE. The last question on each level is a PS without DT's help and thus functions as a quiz for evaluating students' knowledge of the concepts of that level. After completing the entire training in DT, students take an in-class exam, referred to as the ***transfer post-test***. Given that the ultimate goal of the DT tutor is to improve students' performance on the real classroom exam, in the following the transfer post-test scores were used to evaluate students' learning performance and to investigate the effectiveness of pedagogical policies.

In the following, students' pre-test and transfer post-test scores are used for evaluations. We found that the pre-test scores can reflect students' incoming competence; a Pearson correlation test show that a significant correlation between students' pre-test and transfer post-test scores exists: $r(239) = 0.17, p = .005$. However, It is important to note that due to classroom constraints, the pre-test and transfer post-test covered different concepts and were collected at different times: the pre-test occurred in a single session before the policies were employed, while the transfer post-test scores were collected in the classroom after the entire training section is complete. Thus the two scores cannot be directly aligned. Additionally, the transfer post-test is the in-class written test that the RL policies aimed to improve. Therefore, we did not use learning gain to evaluate students' learning performance but rather compare their transfer post-test scores through the ANCOVA tests using pre-test score as the covariate.

### 4.2. Two Training Datasets: ***DT-Imme*** and ***DT-Delay***

Our training dataset was collected in the Fall 2014 and Spring 2015 semesters, with a total of 306 students involved. All students were trained on DT where whether to present the next problem as a WE or a PS was *randomly* decided. The average number of problems solved by students was 23.7 and the average time that each student spent in the tutor was 5.29 hours. In addition, we calculated students' level scores based on their performance on the last problem in each of levels 1–6. For the sake of simplicity, level scores were normalized to $[0, 100]$. Note that when inducing RL policies using the training data set, reward functions are generated based on level scores because transfer post-test scores were not available for the two training datasets and the last problem on each level is designed to be very similar to problems in the transfer post-test. If the students quit the tutor during the training, we assigned a strong negative reward, -300 in this case, on the last problem they attempted. Furthermore, the **immediate reward** was defined as the difference between the current and previous level scores, and the **delayed reward** was defined as the difference of the level scores between level 1 and 6. From the interaction logs, we represent each observation using a high-dimensional feature space introduced in the following section. Combing observation with two types of rewards, we construct two different types of training datasets named ***DT-Imme*** and ***DT-Delay*** respectively.

### 4.3. State Representation

A total of 133 state features, referred as to $\Omega$, were extracted from the DT log files. More specifically, $\Omega$ includes 45 discrete or categorical features and 78 continuous features that can be grouped into five categories listed as follows:

1. **Autonomy (AM).** This category relates to the amount of student work done. For example, *interaction* denotes the cumulative number of student clickstream interactions and *hintCount* denotes the number of times a student clicked the hint button during problem solving. There are a total of 12 features in the AM category, including 8 categorical and 4 continuous features.

2. **Temporal Situation (TS).** This category encodes the time-related information about the work process. For example, *avgTime* denotes the average time taken per problem, and *TotalPSTime* denotes the total time for solving a particular problem. There are a total of 13 continuous features in the TS category.

3. **Problem Solving (PS).** This category encodes information about the current problem-solving context. For example, *probDiff* is the difficulty of the current solved problem; *NewLevel* indicates whether the current solved problem is in a new level in the tutor. There are a total of 30 features in the PS category, including 13 categorical and 17 continuous features.

4. **Performance (PM).** This category describes information about the student's performance during problem solving. For example, *RightApp* denotes the number of correct rule applications. There are a total of 36 features in the PM category, including 24 categorical and 12 continuous features.

5. **Student Action (SA).** This category is a tutor-specific category for DT. It evaluates the statistical measurement of a student's behavior. For instance, *actionCount* denotes the number of non-empty-click actions that students take; *AppCount* denotes the number of clicks for the derivation of a logical expression. There are a total of 32 continuous features in the SA category.

Before feature selection and policy induction, we discretized all continuous features by exploring k-means clustering first and then a simple median split. The latter is conducted only if k-means failed to generate balanced bins. More specifically, the general discretization process is 1) for a given continuous feature, we start by using k-means with $k = 5$ to generate 5 bins; 2) if the sizes of the bins are not balanced, we reduce the value of $k$ by 1 and repeat k-means until balanced bins are achieved; 3) otherwise, if $k = 1$, we use median split to discretize the feature.

In this work, we focus on applying different feature selection approaches to generate a small set of features to construct the state space in a tabular MDP framework. By doing so, we can shed some light on what the most important features are for deciding to apply PS vs. WE. Moreover, when applying RL in real-world scenarios, we may not always have the full computation power to track all of the features at once. Next, we describe the feature selection approaches in Section 5.

## 5. Feature Selection on the MDP Framework

One of the biggest challenges of applying the tabular MDP framework into DT is the high-dimensional feature space. Each state is a vector representation composed of a number of state features and thus the state space grows exponentially in the number of state features, which would cause a data sparsity problem (the available data is not enough to cover each state in the state space) and would exponentially increase the computational complexity. On the other hand, with respect to only including a small set of features, while existing learning literature and theories give helpful guidance on state representation, we argue that such guidance is often considerably more general than the specific state features chosen. For example, to describe a student's knowledge level, we can use "Percentage Correct" defined as the number of the correct student entries divided by the total number of the student entries, or "Number of Correct" defined as the number of the correct student entries, or "Number of Incorrect" defined as the number of the incorrect student entries and so on. When making specific decisions about including a feature of student knowledge level in the state, for example, it is often not clear which of these features should be included. Therefore a more general state representation approach is needed. To this end, this project began with a large set of features to which a series of feature-selection methods were applied to reduce them to a tractable subset.

### 5.1. Related Work For Feature Selection in RL

Much previous work on feature selection for RL mainly focused on model-free RL. Model-free algorithms learn a value function or policy directly from the experience while interacting with the agent. Kolter and Ng (2009) applied Least-Squares Temporal Difference (LSTD) with *Lasso* regularized items to approximate the value function as well as to select an effective feature subset. Similarly, Keller et al. (2006) applied LSTD to approximate a value function and select a feature subset by implementing *Neighborhood Component Analysis* to decompose approximation error, which can be used to evaluate the efficacy of the feature subset. Bach (2009) explored the penalization of an approximation function by using *multiple kernel learning*. Additionally, Wright et al. (2012) proposed the feature selection embedded in a neuro-evolutionary function which approximates the value function, and they selected each feature based on its contribution to the evolution of network topology.

For model-based RL, Chi et al. (2011) previously investigated 10 feature selection methods, called *RLpre-FS* (Sec. 5.4). These methods were implemented to derive a set of various policies, where features are mostly selected based on the single feature's performance or covariance in training data. The results showed there was no consistent winner and in some particular cases these methods perform no better than the random baseline method.

Different from prior work, our features are selected based on the correlations through two steps: 1) a new feature is selected based on its correlation with the current "optimal" subset of features; 2) for different sets of state features, the same $A$, $R$ and training data are used for estimating $T$ when applying MDP to induce policies, and ECR is used to evaluate the induced policies.

### 5.2. Five Correlation Metrics

Our feature selection methods involve five correlation metrics. The first four are commonly used in supervised learning, and here we will investigate whether they can be effectively applied for feature selection in RL. We propose the fifth metric, called Weighted Information Gain (WIG), by combining the first four metrics and adapting them based on the characteristics of our data sets. More specifically, we have:

1. Chi-squared (CHI; McHugh 2013): a statistical test used to identify the independence between the two variables: whether the distribution of a categorical variable differs significantly from another categorical variable.

2. Information gain (IG; Lee and Lee 2006) measures how much information we would gain about a variable $Y$ if knowing another variable $X$. It is calculated as:

   $$IG(Y, X) = H(Y) - H(Y|X) \tag{5}$$

   where $H(\cdot)$ is the entropy function – measuring the uncertainty of a variable. IG(Y, X) evaluates how the uncertainty of a variable $Y$ would change from knowing the variable $X$. To some extent, it can also be treated as a type of correlation between $X$ and $Y$. Note that IG has the bias towards variables with a large number of distinct values.

3. Symmetrical uncertainty (SU; Yu and Liu 2003) is defined as:

   $$SU(Y, X) = \frac{H(Y) - H(Y|X)}{H(X) + H(Y)} \tag{6}$$

   SU evaluates the correlation between two variables $Y$ and $X$ by normalizing $IG(Y, X)$. SU compensates for the weakness of IG by considering the uncertainty of both variables $X$ and $Y$ in the denominator.

4. Information gain ratio (IGR; Kent 1983) is the ratio of information gain to the intrinsic information, which is the entropy of conditional information. IGR can be represented as:

   $$IGR(Y, X) = \frac{H(Y) - H(Y|X)}{H(X)} \tag{7}$$

   Compared with SU, IGR only considers the uncertainty of variable $X$ in the denominator.

5. Weighted Information gain (WIG) is proposed as:

   $$WIG(Y, X) = \frac{H(Y) - H(Y|X)}{(H(Y) + H(X)) \cdot H(X)} \tag{8}$$

   WIG can be seen as a combination of IG, SU and IGR. Compared to SU, WIG sets more weight on $X$ by multiplying $H(X)$ in the denominator; while compared to IGR, WIG normalizes IG by considering the uncertainty of both variables $X$ and $Y$.

In our application, each of the five correlation metrics is used for evaluating the correlation between the current selected feature set $Y$ with a new feature $X$. For each metric we explore two options: The High option is to select the next feature that is ***most correlated*** to the currently selected feature set whereas the Low option is to select the ***least correlated*** feature. As described above, the high correlation-based option is commonly used for supervised learning where the features that are most highly correlated with the output labels are often selected (Yang and

Pedersen, 1997; Lee and Lee, 2006; Chandrashekar and Sahin, 2014; Koprinska et al., 2015). However, for RL, the high option-selected features tend to be homogeneous. Different from the supervised learning tasks, we hypothesize that it is more important to have heterogeneous features in RL that can grasp different aspects of learning environments. Therefore, we also explore the low correlation-based option for feature selection with a goal to increase the diversity of the selected feature set. As a result, we have 10 correlation-based methods named: CHI-high, IG-high, SU-high, IGR-high, WIG-high, CHI-low, IG-low, SU-low, IGR-low, and WIG-low. Our goal is to investigate which option is better: high vs. low, and which of the five correlation metric performs the best.

### 5.3. Correlation-based Feature Selection Approaches

Algorithm 1 shows the process of our correlation-based feature selection method. It contains three major parts. In the first part (lines 1–4), the algorithm constructs MDPs for every single feature in $\Omega$, induces a single-feature policy and calculates its $ECR$ (defined in Sect. 4). Then the feature with highest $ECR$ is added to the current optimal feature set $\mathcal{S}^*$. In the second part (lines 6–9), the algorithm follows a forward step-wise feature selection procedure in that, given the currently selected feature set $\mathcal{S}^*$, it selects the next feature based on the five correlation metrics described above. More specifically, it first calculates the correlations between $\mathcal{S}^*$ with each feature $f_i \in \Omega - \mathcal{S}^*$ using a specific correlation metric $m$, ranks the results, and then selects the top 5 features with the highest correlations for high-option or the bottom 5 lowest features for low options, decided by the Boolean variable *reverse* in line 9. These features are selected to form a feature pool $\mathcal{F}$. In the third part (lines 10–13), the current $\mathcal{S}^*$ is combined with each feature $f_i \in \mathcal{F}$ to induce a policy, and the $Calculate\text{-}ECR$ function calculates the $ECR$ of the induced policy. Then $\mathcal{S}^* + f_k$, the combination that produces the policy with the highest $ECR$, will be the new $\mathcal{S}^*$ for the next round. The algorithm will terminate when the size of an optimal feature set reaches maximum number $\mathcal{N}$.

**Algorithm 1** Correlation-based Feature Selection Algorithm

```
Require: $\Omega$: Feature space; $\mathcal{D}$: Training data; $\mathcal{N}$: Maximum number of selected features
Ensure: $\mathcal{S}^*$: Optimal feature set
 1: for $f_i$ in $\Omega$ do
 2:     $ECR_i \leftarrow$ CALCULATE-ECR($\mathcal{D}$, $f_i$)
 3: end for
 4: Add $f^*$ with highest $ECR$ to $\mathcal{S}^*$
 5: while SIZE($\mathcal{S}^*$) $< \mathcal{N}$ do
 6:     for $f_i$ in $\Omega - \mathcal{S}^*$ do
 7:         $C_i \leftarrow$ CALCULATE-CORRELATION($\mathcal{S}^*$, $f_i$, $m$)    ▷ $m$ refers to a correlation metrics
 8:     end for
 9:     $\mathcal{F} \leftarrow$ SELECTTOP(C, 5, reverse)   ▷ Select top or bottom 5 features based on metrics $m$
10:     for $f_i$ in $\mathcal{F}$ do
11:         $ECR_i \leftarrow$ CALCULATE-ECR($\mathcal{D}$, $\mathcal{S}^* + f_i$)
12:     end for
13:     Replace $\mathcal{S}^*$ by $\mathcal{S}^* + f_k$ with highest $ECR$
14: end while
```

### 5.4. PreRL-FS Approach

Chi et al. (2011) developed a series of feature selection approaches, referred to as *PreRL-FS* in the following. They can be grouped into three categories: 1) four ECR-based methods, which use ECR, Upper-Bound of ECR, Lower-Bound of ECR, or Hedge value of the single-feature policy as the feature selection criteria where the Upper-Bound and Lower-Bound of ECR refer to the 95% confidence interval for ECR, and Hedge is defined as $Hedge = ECR/(UpperBound - LowerBound)$; 2) two PCA-based methods, which select features that are highly correlated with principal components; and 3) four ECR & PCA-based methods, the combination of the former two approaches. The results indicated that the *four ECR-based* methods outperformed the other two types of approaches in terms of ECR.

**Algorithm 2** Ensemble Feature Selection Algorithm

```
Require:
    $\Omega$: Feature space; $\mathcal{D}$: Training data; $\mathcal{N}$: Maximum number of selected features;
    $\mathcal{M}$: A set of feature selection approaches.
Ensure: $\mathcal{S}^*$: Optimal feature set
 1: for $f_i$ in $\Omega$ do
 2:     $ECR_i \leftarrow$ CALCULATE-ECR($\mathcal{D}$, $f_i$)
 3: end for
 4: Add $f^*$ with highest $ECR$ to $\mathcal{S}^*$
 5: while SIZE($\mathcal{S}^*$) $< \mathcal{N}$ do
 6:     $\mathcal{F} \leftarrow \emptyset$
 7:     for $Method_k$ in $\mathcal{M}$ do
 8:         $\mathcal{F}_k \leftarrow$ SELECT-FEATURE($\mathcal{D}$, $\Omega - \mathcal{S}^*$, $Method_k$)
 9:         $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_k$
10:     end for
11:     for $f_i$ in $\mathcal{F}$ do
12:         $ECR_i \leftarrow$ CALCULATE-ECR($\mathcal{D}$, $\mathcal{S}^* + f_i$)
13:     end for
14:     Replace $\mathcal{S}^*$ by $\mathcal{S}^* + f_k$ with highest $ECR$
15: end while
```

### 5.5. Ensemble Approach

Algorithm 2 shows the basic process of our ensemble feature selection procedure, which is similar to that of correlation-based methods. The major difference is in the second part (lines 6–10). Our ensemble approach explored a total of 12 feature selection methods that are referred to as $\mathcal{M}$ in Algorithm 2: the four *ECR-based* methods which are the better methods among the *PreRL-FS* approaches and the eight out of the 10 proposed correlation-based methods (WIG-high and WIG-low were excluded here because they were not explored when we first explored the ensemble approach). More specifically, the ensemble approach integrates the features $\mathcal{F}_k$ generated from each of feature selection method $Method_k$ in $\mathcal{M}$ and generates a relatively large feature pool $\mathcal{F}$. The maximum size of $\mathcal{F}$ can be up to 70, but often much smaller because of the overlapping of feature sets generated from different methods. Note that the feature pool is still much larger than any of our 10 correlation-based methods, which is 5. After generating the feature pool, the ensemble method carries out the same procedure, the third part (lines 11–13), as the correlation-based methods described above. Although the ensemble method has a relatively high computational complexity, it has a wider exploration of the feature space by integrating different types of feature selection methods.

## 5.6. Comparison Results for Feature Selection Approaches

We explore three categories of feature selection approaches: *PreRL-FS*, ensemble, and high- and low- correlation-based approaches and compare them against a random feature selection baseline. We use ECR to theoretically evaluate the effectiveness of the MDP policies, which indirectly verify the effectiveness of feature selection approaches. Note that ECR is calculated based on the induced MDP policies and the two training datasets: DT-Immed and DT-Delay (Section 4.2).

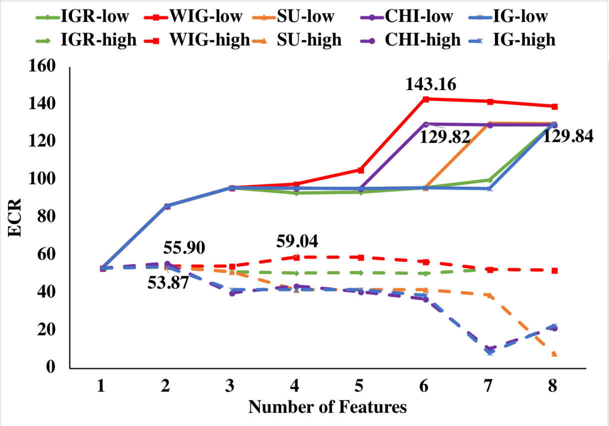


Figure 3: DT-Immed

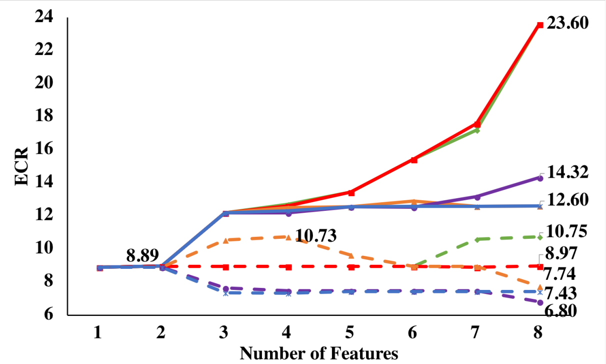


Figure 4: DT-Delay

**High VS Low Correlation-based Approaches.** Figure 3 and Figure 4 show the ECR values of 10 correlation-based methods on DT-Immed and DT-Delay respectively, where the y-axis represents the value of the ECR of the induced policy given the selected features, and the x-axis denotes the number of features (maximum is 8). Note that all feature selection methods start in the same place at $x = 1$ except the random method. This is because all methods will initially select the feature with the best ECR of single-feature policy. However, ECR values vary dramatically as the number of selected features increases. The solid line indicates the performance of the low correlation-based approaches and the dotted line denotes the performance of the high correlation-based version. In addition, the ECR value of policies using immediate reward is much higher than that of policies using the delayed reward.

The results show that for each of the five correlation metrics, the low correlation-based option significantly outperforms the high correlation-based option. For the *DT-Immed* dataset, the $ECR$ of WIG-low is 143.16, while $ECR$ of WIG-High is only 59.04. Similarly, the $ECRs$ of CHI-low and CHI-High are 129.82 vs. 55.90. The average percent increase for the low correlation methods over the high correlation methods is 75.35%, the maximum percent increase is 142.48%, and the minimum percent increase is 17.24%. To summarize, our results show that low correlation is more suitable for the MDP framework than high correlation, and indirectly illustrate that the high variety of the feature space had a positive impact on the effectiveness of the induced policies. The same pattern was found in the *DT-Delay* dataset.

**Five Correlation Metrics.** Figures 3 and 4 show that WIG is the consistently highest performer in that it has the best ECR for both DT-Immed and DT-Delay datasets. CHI performed well in DT-Immed dataset while IGR performed well in DT-Delay dataset. In short, our proposed WIG performed best among all the five correlation metrics.

**Overall Comparison.** Figures 5 and 6 show the overall comparison among all methods on DT-Immed and DT-Delay data respectively. Particularly, with the purpose of simplicity, for both low and high correlation-based methods and the PreRL-FS methods, we selected the best method from each category. In other words, the figures present a comparison among the five methods including the best of five Low-correlations, the best of five High-correlations, ensemble, the best of PreRL-FS, and the random approach. Results show that, as expected, the random method performs worst across the two datasets. In addition, the best of the high correlation-based methods outperforms random and Best-RLPreviousFS approaches when the number of features is above 5. The best of the low correlation-based methods outperforms other methods. In general, the best low correlation-based method outperforms the best of *PreRL-FS* by an average of 43.87% and outperforms the ensemble method by an average of 9.05%. In addition, the ensemble method improves over the best of *PreRL-FS* by an average of 36.46%. The value of ECR does not always rise as the the number of features increases. The ECR of the low-correlation approach decreases a lot when increasing the number of features from 6 to 8. The ECR of the ensemble method seems to converge when the number of features is more than 6 for both two training datasets. The ECR of the best of *PreRL-FS* decreases when the number of features is more than 4.

In summary, based on ECR results we can rank five categories of methods as Low correlation-based $>$ Ensemble $>$ High correlation-based $\approx$ PreRL-FS $\gg$ Random. In particular, the WIG-Low approach performs best among all implemented approaches.

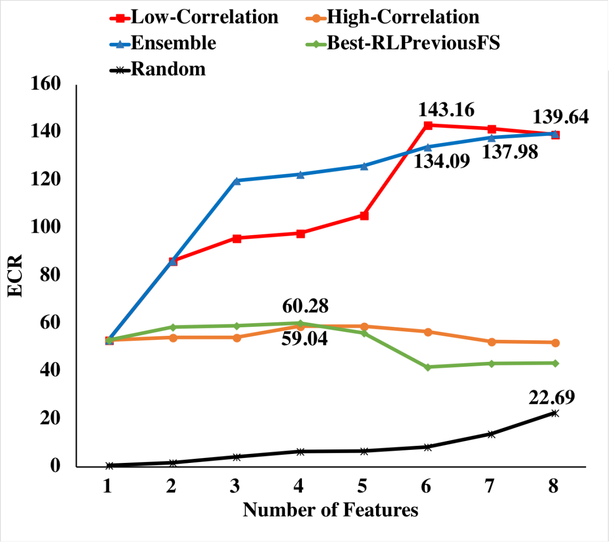


Figure 5: DT-Immed

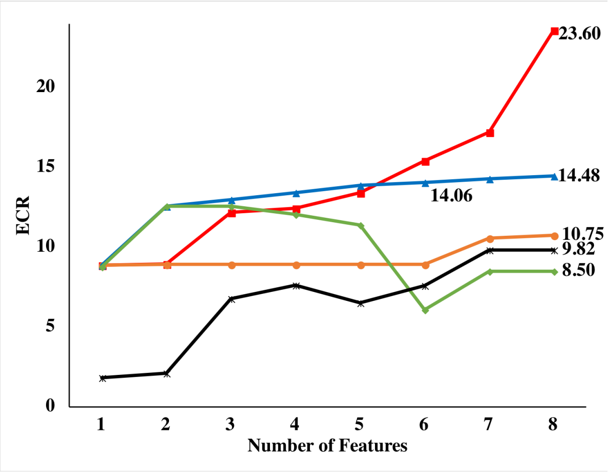


Figure 6: DT-Delay

# 6. Three Research Questions & Four Experiments Overview

## 6.1. Three Research Questions

In this work, we investigate the effectiveness of RL-induced policies using the MDP framework from three aspects: state representation using different feature selections, reward function, and policy execution options. For each aspect, we have a corresponding research question and thus our three research questions are listed as follows:

- **Q1 (State):** Can effective feature selection methods empirically improve the effectiveness of the induced policy?
- **Q2 (Reward):** Does immediate reward facilitate the MDP framework to induce a more effective pedagogical policy than delayed reward?
- **Q3 (Execution):** Can stochastic policy execution be more effective than deterministic policy execution?

## 6.2. Five Reinforcement Learning Policies

Table 2 lists the five RL policies induced for investigating the three research questions above. All five policies were induced using the MDP framework but involved different types of feature selection methods (the second column), reward function (the third column), and/or policy execution (the fourth column). The last column shows that the ECR of the RL-induced policies. More specifically, *MDP-ECR* is induced by using MDP with the best PreRL-FS feature selection approach; *Ensemble-Imme* and *Ensemble-Delay* are two policies induced with the ensemble feature selection approach using immediate and delayed reward respectively; and *WIG-det* and *WIG-sto* were both induced using WIG with the low-correlation option for feature selection, and the main difference is that the former is executed deterministically while the latter is executed stochastic. Note that because *WIG-sto* is a stochastic policy and because ECR can only be calculated for a deterministic policy, the ECR of *WIG-sto* is listed as “NA”.

Table 2: Reinforcement Learning Policies in Four Experiments

| Policy | Feature Selection | Reward | Execution | ECR |
|---|---|---|---|---:|
| *MDP-ECR* | ECR-based | Immediate | Deterministic | 60.28 |
| *Ensemble-Imme* | Ensemble | Immediate | Deterministic | 137.98 |
| *Ensemble-Delay* | Ensemble | Delay | Deterministic | 14.06 |
| *WIG-det* | Low Corre-based | Immediate | Deterministic | 143.16 |
| *WIG-sto* | Low Corre-based | Immediate | Stochastic | *NA* |

Note: ECR is only used for evaluating the deterministic policies

## 6.3. Four Experiments

Four experiments, one per semester from the Spring of 2015 to the Fall of 2017, were conducted to empirically evaluate the impact of the three aspects on the effectiveness of the five RL-induced policies described above. In each experiment, we compared one or two RL policies against the Random yet reasonable baseline policy. Table 3 shows the overview of the four experiments and the corresponding research questions.

Table 3: Overview of Experiments

| Experiment | Policies | Research Question | | |
|---|---|---|---|---|
| | | Q1(State) | Q2(Reward) | Q3(Execution) |
| Experiment 1 | *MDP-ECR* vs. *Random* | ✓ | | |
| Experiment 2 | *Ensemble-Imme* vs.<br>*Ensemble-Delay* vs. *Random* | ✓ | ✓ | |
| Experiment 3 | *WIG-det* vs. *Random* | ✓ | | |
| Experiment 4 | *WIG-sto* vs. *Random* | ✓ | | ✓ |

## 6.4. ATI Effect: Splitting Students Based on Response Time

Overall, results across the four experiments consistently exhibit an ATI effect. That is, rather than all students, only certain students' learning is significantly affected by the pedagogical decisions on PS vs. WE. In the following, they are referred to as the *Responsive* group and by contrast, we refer to other students as the *Unresponsive* group. It is often not clear which students are more sensitive to the induced policy due in part to the fact that we do not fully understand why such differences exist. In this work, we split *Responsive* and *Unresponsive* groups based upon some measurement of incoming competence.

One common way to measure students' incoming competence is to use their pre-test scores. Across the four experiments, all of the students received the same initial training at Level 1 and our results showed that students' pre-test scores indeed reflect their incoming competence in that a significant positive correlation between students' pre-test scores and transfer post-test scores: $r = 0.17, n = 241, p = .005$. However, applying a median split on pre-test *for all participants* results in unbalanced splits *within treatment groups*. For example, in Experiment 1, a split using the median value of student's pre-test scores would divide the Random group into 16 High pre-test group vs. 6 in the Low pre-test group. Similarly, in Experiment 3, the WIG-det group would divide into 31 in the High pre-test group and 14 in the Low pre-test group.

On the other hand, ever since the mid-1950s, response time has been used as a preferred dependent variable in cognitive psychology (Luce et al., 1986). It has often been used to assess student learning because response time can indicate how active and accessible student knowledge is. For example, it has been shown that response time reveals student proficiency (Schnipke and Scrams, 2002) and that students' average response time and their final exam scores are negatively correlated (González-Espada and Bullock, 2007). With the advent of computerized testing, more and more researchers have begun to use response-time as a learning performance measurement (Schnipke and Scrams, 2002). Inspired by this prior work, we use the average time in Level 1 (*avgTime*) to split students which consistently generated more balanced groups across all four experiments. Therefore, in the following studies, students were split using *avgTime*.

To summarize, in each of the following experiments, students are divided into *Responsive* and *Unresponsive* groups by a median split on their response time at Level 1. Since each experiment has a slightly different median value and criteria for splitting, there is no general definition for the *Responsive* and *Unresponsive* groups. In the post-hoc comparison, we combined all of the experiments and used a global median split to check whether our results would still hold.

### 6.5. Statistical Analysis

In the following analyses, we run several different types of statistical tests to evaluate student performance with a focus on their transfer post-test scores. Although students' pre-test scores were not used to split students into *Responsive* and *Unresponsive* groups, they are used as the covariate in ANCOVA when comparing students' transfer post-test scores.

To confirm that the assumptions of ANCOVA were met, for each experiment ANOVA tests were performed and indicated that there is no significant difference on pre-test score among different treatment groups. In addition, two-way ANOVA tests for each experiment using group and pre-test as factors show that there is no significant interaction effect on transfer post-test score. These results indicate that the pre-test covariate and treatment group variable are independent and that the relationship between the covariate and treatment group variable is the same across groups. Thus, the assumptions of ANCOVA are met, and we report ANCOVA results for the transfer post-test scores.

## 7. Four Experiments

### 7.1. Experiment 1: Preliminary Feature Selection

Experiment 1 was conducted in the Spring of 2015. We compared two policies: an MDP policy and a Random baseline policy. Our research question in Experiment 1 is Q1 (State): Can effective feature selection methods empirically improve the effectiveness of the induced policy?

For Experiment 1, we only explored the PreRL-FS feature selection approaches, and among them, the ECR-based approach using the lower bound of ECR as the selection criteria performed the best. In the following, we refer to the induced policy as the *MDP-ECR* policy. Table 4 shows the definition of the four selected features (left) and the corresponding policy (right). The row denotes the value of the first two features while the column denotes the value of the last two features. For example, when the four features $f_1$:$f_2$:$f_3$:$f_4$ is 0:0:0:0 (the top-left cell), the decision is a PS (black cell). Overall, the *MDP-ECR* policy contains 11 pedagogical rules that propose a PS (black cells) and 5 rules that propose a WE (white cells).

#### 7.1.1. Experiment 1: Participants & Conditions

DT was assigned to 67 undergraduate students as one of their regular homework assignments. Completion of the tutor was required for full credit. Students were randomly assigned to the two conditions: Random ($N = 22$) and *MDP-ECR* ($N = 45$). Because all students followed the random policy when collecting our training data for RL in previous years, we assign more students to the *MDP-ECR* condition to evaluate the effectiveness of RL-induced policies.

Results of Experiment 1 show that there is no significant difference between the *MDP-ECR* and Random on either pre-test ($F(1, 65) = 1.81$, $p = 0.18$) or transfer post-test ($F(1, 65) = 0.46$, $p = 0.50$). However, once we did a median split on students based on the students' "average response time on level 1", our results show that students whose *level1-avgstepTime* $<$ 7.1 sec are more sensitive to the effectiveness of pedagogical strategies than their peers whose

Table 4: *MDP-ECR* Policy

**NextClickWE** ($f_1$): Number of next step clicks in a Worked Example

**TotalWETime** ($f_2$): Total time for solving a Worked Example

**symbolicRepnCount** ($f_3$): Number of problems using symbolic representation

**difficultProbCount** ($f_4$): Number of solved difficult problems

Note: Black: PS, White: WE

| First two features $f_{I1}$:$f_{I2}$ \ Last two features $f_{I3}$:$f_{I4}$ | 0:0 | 0:1 | 1:0 | 1:1 |
|---|---|---|---|---|
| 0:0 | ■ | ■ | □ | □ |
| 0:1 | ■ | ■ | □ | □ |
| 1:0 | ■ | ■ | ■ | ■ |
| 1:1 | ■ | □ | ■ | ■ |

*level1-avgstepTime* $\geq$ 7.1 sec. In the following, we refer the former as the Responsive group and the latter as the Unresponsive group. By combining Policy {MDP-ECR, Random} with Type {Responsive, Unresponsive}, we have a total of four groups including: Random-Resp ($N$ = 9), Random-Unres ($N$ = 13), *MDP-ECR-Resp* ($N$ = 23) and *MDP-ECR-Unres* ($N$ = 22). Pearson's Chi-squared test showed that there was no significant difference on the distribution of Unresponsive vs. Responsive between the two policies, $\chi^2(1, N = 67) = 0.27, p = 0.59$.

Table 5: Pre-test and Transfer Post-test in Experiment 1

| Type | Pre-Test Score | | Transfer Post-Test Score | |
|---|---|---|---|---|
| | MDP-ECR | Random | MDP-ECR | Random |
| Resp | 58.06(29.92) | 48.59(35.64) | **87.50(16.38)** | 69.88(34.43) |
| Unres | 53.87(33.17) | **86.15(20.42)** | 69.94(28.54) | 79.54(23.73) |
| Total | 56.11(31.19) | 67.37(34.24) | 79.31(24.27) | 74.71(29.28) |

### 7.1.2. Experiment 1: Results

Table 5 presents the mean and SD for students' corresponding learning performance in Experiment 1. Despite the fact that students are randomly assigned, *Random-Unres* significantly outperforms all other groups on the pre-test according to results of ANOVA tests: $F(1, 20) = 9.01$, $p = .007$ for *Random-Resp*, $F(1, 33) = 8.82$, $p = .006$ for *MDP-ECR-Unres*, $F(1, 34) = 6.37$, $p = .016$ for *MDP-ECR-Resp*, probably due to the small sample size in the random groups. Despite *Random-Unres* out-performance, no significant difference is found on the pre-test score either between *MDP-ECR* and *Random* (two columns): $F(1, 65) = 1.81$, $p = 0.18$, or between Responsive and Unresponsive (two rows): $F(1, 65) = 1.26$, $p = 0.27$. Furthermore, a possible explanation for a high pre-test score of *Random-Unres* is that *Random-Unres*, considered as the *high proficiency* students, can always learn regardless of teaching policies and are less sensitive to the learning environment (Cronbach and Snow, 1977; Chi et al., 2011).

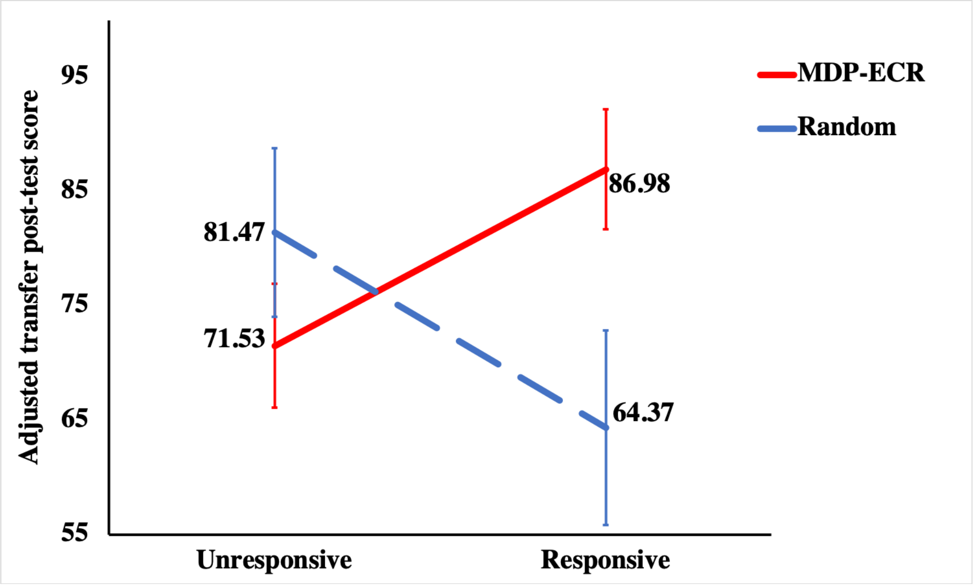


Figure 7: Interaction effect for the adjusted transfer post-test score in Experiment 1

**Transfer Post-Test Score.** A two-way ANCOVA test, using Policy and Type as the two factors and the pre-test score as covariate shows that there is a significant interaction effect on their transfer post-test scores: $F(1, 62) = 5.39, p = .023$, but no significant main effect of either Policy or Type. Figure 7 depicts the cross-over interaction between Policy and Type on the *adjusted transfer post-test score*, which is the transfer post-test score adjusted by the linear regression (ANCOVA) model built to describe the relation between the pre- and transfer post-test score.

Planned contrasts using Tukey's adjustment reveal a significant difference between the two Responsive groups in that *MDP-ECR-Resp* scored significantly higher adjusted transfer post-test than *Random-Resp*, $t(62) = 2.26$, $p = .027$, while there is no significant difference between two Unresponsive groups.

### 7.1.3. Experiment 1: Conclusion

In summary, we find a significant interaction effect in that MDP-ECR benefits the Responsive students significantly more than the Unresponsive students, while no such difference was found between the Responsive and Unresponsive groups under the Random policy. However, one important limitation of Experiment 1 is that the *Random-Unres* group has significant higher pre-test score than all other groups. Therefore, in Experiment 2, we repeat the general procedure of Experiment 1 but explored correlation based and ensemble-based feature selection methods and also explore both Immediate and Delayed rewards.

## 7.2. Experiment 2: Ensemble Feature Selection & Immediate vs. Delayed Rewards

Experiment 2 was conducted in the Fall of 2016 and investigated two research questions:

- **Q1 (State):** Can effective feature selection methods empirically improve the effectiveness of the induced policy?

- **Q2 (Reward):** Does immediate reward facilitate the MDP framework to induce a more effective pedagogical policy than delayed reward?

In Experiment 2, we applied ensemble feature selection (Section 5.5) to select a small subset of features from the original 133 features for inducing two policies, named *Ensemble-Imme* and *Ensemble-Delay*, from the two training datasets ***DT-Imme*** and ***DT-Delay*** respectively (Section 4.2). More specifically, *Ensemble-Imme* involves seven features and *Ensemble-Delay* policy involves six features. Table 6 and 7 display the selected features as well as the corresponding policies. In the tables, black cells denote **PS** actions, white cells denote **WE** actions, and gray cells denote that no policy is induced due to the absence of the state in the training data. Generally speaking, the *Ensemble-Imme* policy prefers WE over PS as it contains 65 rules for WE vs. 21 rules for PS; while *Ensemble-Delay* policy prefers PS over WE as it has 48 rules for PS and 18 for WE. Additionally, while Figure 5 shows that the ensemble feature selection with eight features would result in a higher ECR policy than the policy with seven features, we still used the latter here because 1) the ECRs of the two policies are actually very close; and 2) the seven-feature policy is less complicated and has less "none-mapping" from state to action (the gray color cells) compared with the eight-feature policy. For similar reasons, we determined the *Ensemble-Delay* policy to be six features.

### 7.2.1. Experiment 2: Participants and Conditions

A total of 106 students participated in Experiment 2 and were randomly assigned into three conditions: *Random* ($N = 30$), *Ensemble-Imme* ($N = 38$) and *Ensemble-Delay* ($N = 38$). 94

Table 6: Ensemble-Imme Policy

**TotalPSTime** ($f_1$): Total time for solving a problem
**NewLevel** ($f_2$): Whether the current solved problem is in a new level
**WrongApp** ($f_3$): Number of incorrect application of rules
**TotalWETime** ($f_4$): Total time for working on a worked example
**UseCount** ($f_5$): Number of different types of applied rules
**AppCount** ($f_6$): Number of clicks for derivation
**NumProbRule** ($f_7$): Number of expected distinct rules for a solved problem

| First four features $f_{I1}:f_{I2}:f_{I3}:f_{I4}$ | Last three features $f_{I5}:f_{I6}:f_{I7}$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0:0:0 | 0:0:1 | 0:1:0 | 0:1:1 | 1:0:0 | 1:0:1 | 1:1:0 | 1:1:1 |
| 0:0:0:0 | □ | □ | □ | □ | □ | ■ | □ | □ |
| 0:0:0:1 | □ | □ | □ | □ | □ | ■ | □ | ■ |
| 0:0:1:0 | ■ | ■ | ■ | ■ | ■ | ▨ | □ | ■ |
| 0:0:1:1 | □ | □ | ▨ | ▨ | □ | ■ | □ | ■ |
| 0:1:0:0 | □ | □ | ▨ | ▨ | □ | ▨ | ▨ | ▨ |
| 0:1:0:1 | □ | □ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |
| 0:1:1:0 | □ | □ | □ | □ | ▨ | ▨ | ▨ | ▨ |
| 0:1:1:1 | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |
| 1:0:0:0 | □ | □ | ■ | □ | □ | ▨ | □ | ■ |
| 1:0:0:1 | □ | □ | □ | □ | □ | ▨ | □ | □ |
| 1:0:1:0 | ■ | ■ | □ | □ | ■ | □ | □ | □ |
| 1:0:1:1 | □ | □ | □ | □ | ■ | ■ | ■ | □ |
| 1:1:0:0 | □ | □ | □ | □ | ▨ | ▨ | ▨ | ▨ |
| 1:1:0:1 | ■ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |
| 1:1:1:0 | □ | □ | □ | □ | ■ | □ | □ | □ |
| 1:1:1:1 | □ | ▨ | ▨ | ▨ | □ | □ | □ | ▨ |

Table 7: Ensemble-Delay Policy

**stepTimeDev** ($f_1$): Step time deviation
**probDiff** ($f_2$): Difficulty of the current solved problem
**symbolicRCount** ($f_3$): Number of whole problems for symbolic representation
**actionCount** ($f_4$): Number of non-empty-click actions taken by students
**SInfoHintCount** ($f_5$): Number of System Information Hint requests
**NSClickCountWE** ($f_6$): Number of next step clicks in Worked Examples

| | Last three features $f_{D4}$:$f_{D5}$:$f_{D6}$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| First three features $f_{D1}$:$f_{D2}$:$f_{D3}$ | 0:0:0 | 0:0:1 | 0:1:0 | 0:1:1 | 1:0:0 | 1:0:1 | 1:1:0 | 1:1:1 |
| 0:0:0 | ■ | ■ | ▨ | ▨ | □ | □ | ▨ | ▨ |
| 0:0:1 | □ | ■ | □ | ▨ | ▨ | ▨ | ▨ | ▨ |
| 0:1:0 | ■ | □ | ■ | ■ | ■ | □ | ■ | ■ |
| 0:1:1 | ■ | □ | ■ | ▨ | ■ | □ | ■ | ■ |
| 0:2:0 | ■ | ■ | ▨ | ▨ | ■ | ■ | ■ | ■ |
| 0:2:1 | ■ | ■ | ▨ | ▨ | ■ | ■ | ■ | ▨ |
| 1:0:0 | □ | ■ | ▨ | ▨ | □ | □ | ■ | ▨ |
| 1:0:1 | □ | □ | ▨ | ▨ | □ | □ | □ | ▨ |
| 1:1:0 | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| 1:1:1 | ■ | ■ | ■ | ▨ | ■ | ■ | □ | ■ |
| 1:2:0 | ■ | ■ | ▨ | ▨ | ■ | □ | ■ | ▨ |
| 1:2:1 | ■ | ▨ | ▨ | ▨ | ■ | ■ | ▨ | ▨ |

students completed the assignment, distributed as *Random* ($N = 27$), *Ensemble-Imme* ($N = 34$) and *Ensemble-Delay* ($N = 33$). Pearson's chi-squared test yielded no significant relation between completion rate and condition, $\chi^2(2, N = 106) = .012, p = .994$.

The last row in Table 8 (a) presents the mean and SD for students' corresponding learning performance in Experiment 2. No significant difference was found among the three policies on either pre-test ($F(2, 91) = 0.04$, $p = 0.96$) or transfer post-test ($F(2, 91) = 1.33$, $p = 0.27$). Furthermore, similar as Experiment 1, we use the median of "average response time on level 1" (median(*level1-avgstepTime*) = 8.01 sec) to split students in Experiment 2 . Different from Experiment 1, it was shown that students whose *level1-avgstepTime* $<$ 8.01 sec are less sensitive to the effectiveness of pedagogical strategies than those whose *level1-avgstepTime* $\geq$ 8.01 sec, and thus we refer the former as the Unresponsive group and the latter as the Responsive group.

By combining Policy with Type {Responsive, Unresponsive}, we have a total of six groups including three Unresponsive groups: *Random-Unres* ($N = 15$), *Ensemble-Imme-Unres* ($N = 16$), *Ensemble-Delay-Unres* ($N = 15$); and three Responsive groups: *Random-Resp* ($N = 12$), *Ensemble-Imme-Resp* ($N = 18$), and *Ensemble-Delay-Resp* ($N = 18$). Pearson's chi-squared test shows that there is no significant difference in the distribution of Unresponsive vs. Responsive among the three conditions, $\chi^2(1, N = 94) = .681, p = .711$.

### 7.2.2. Experiment 2: Results

Table 8 presents the mean and SD for students' corresponding learning performance. One-way ANOVA tests show that there is no significant difference on the pre-test score either among the three policies {*Ensemble-Imme*, *Ensemble-Delay*, *Random*}, $F(2, 91) = 0.04$, $p = 0.96$, or among the three Unresponsive groups, $F(2, 43) = 0.14$, $p = 0.87$, or among the three Responsive groups, $F(2, 45) = 0.65$, $p = 0.53$. Additionally, there is a significant difference between Responsive and Unresponsive: the former scores significantly higher than the latter on the pre-test score, $F(1, 92) = 7.33$, $p = .008$.

Table 8: Pre-test and Transfer Post-test in Experiment 2

| Type | Pre-Test Score | | | Transfer Post-Test Score | | |
|---|---|---|---|---|---|---|
| | Ensemble-Imme | Ensemble-Delay | Random | Ensemble-Imme | Ensemble-Delay | Random |
| Resp | 62.20(31.84) | 68.17(30.81) | 74.76(23.90) | **90.97(24.36)** | 81.25(31.43) | 62.24(40.16) |
| Unres | 54.11(37.27) | 48.27(27.71) | 49.56(30.47) | 83.33(22.36) | **92.38(10.64)** | 88.75(22.43) |
| Total | 58.52(34.11) | 58.81(30.65) | 60.76(30.07) | 87.50(23.43) | 86.49(24.34) | 76.96(33.67) |

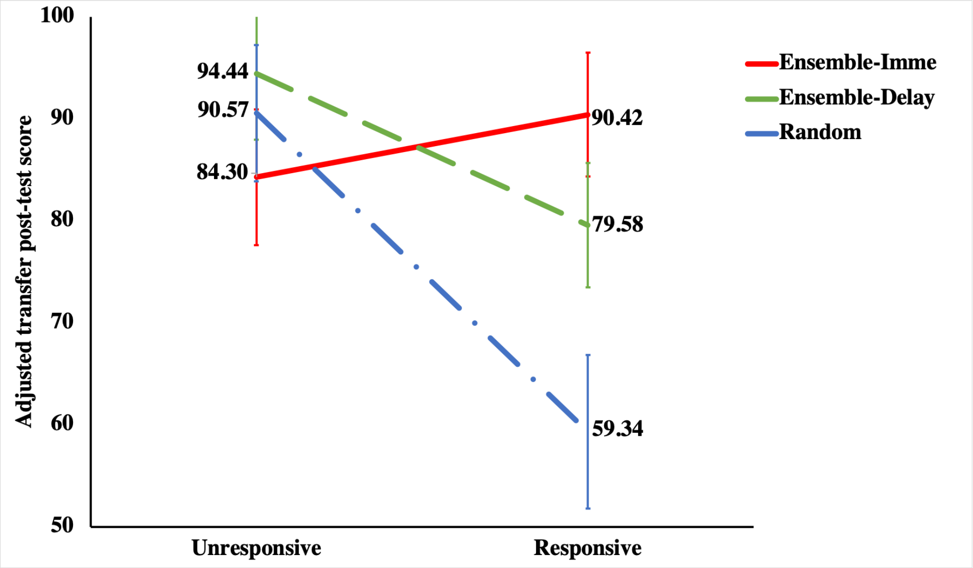


Figure 8: Interaction effect for the adjusted transfer post-test score in Experiment 2

Table 9: Pairwise Contrasts on Adjusted Transfer Post-test in Experiment 2

| Pairwise Policy Comparison | | | $t(87)$ | $p$-value |
|---|---|---|---|---|
| Ensemble-Imme-**Resp** | vs. | Ensemble-Delay-**Resp** | $-1.26$ | 0.75 |
| Ensemble-Imme-**Resp** | vs. | Random-**Resp** | 3.22 | 0.01 * |
| Ensemble-Delay-**Resp** | vs. | Random-**Resp** | 2.11 | 0.21 |
| Ensemble-Imme-**Unres** | vs. | Ensemble-Delay-**Unres** | 1.09 | 0.85 |
| Ensemble-Imme-**Unres** | vs. | Random-**Unres** | $-0.67$ | 0.98 |
| Ensemble-Delay-**Unres** | vs. | Random-**Unres** | 0.42 | 0.99 |

· marginal significant at $p < 0.1$; * significant at $p < 0.05$.

**Transfer Post-Test Score.** A two-way ANCOVA test, using Policy {*Ensemble-Imme*, *Ensemble-Delay*, *Random*} and Type {*Responsive*, *Unresponsive*} as two factors and the pre-test score as covariate, shows that there is a significant main effect of Type, $F(1, 87) = 4.45$, $p = .037$, and a significant interaction effect on transfer post-test scores, $F(2, 87) = 3.90$, $p = .024$. Figure 8 presents the cross-over interaction between Policy and Type on the *adjusted transfer post-test score*, which is the transfer post-test score adjusted by the linear regression model built to describe the relation between the pre- and transfer post-test score.

Table 9 presents the results of contrast tests using Tukey's adjustment for multiple comparisons. Results indicate that while there is no significant difference among three Unresponsive groups, *Ensemble-Imme-Resp* achieved significantly higher adjusted transfer post-test score than *Random-Resp*: $p = 0.01$.

### 7.2.3. Experiment 2: Conclusion

Our empirical results suggest that the ATI effect exists in Experiment 2: while no significant difference is found among the three Unresponsive groups, a significant difference is found among the three Responsive groups in that students following the *Ensemble-Imme* policy score significantly higher on the transfer post-test than their peers following the *Random* policy. This suggests that immediate reward can facilitate the MDP framework to induce an effective policy and that the ensemble feature selection approach is able to extract a good subset of features for MDP to induce a more effective policy compared with the Random policy. Finally, since it was shown that the immediate reward is more effective than the delayed reward for policy induction in the MDP framework in Experiment 2, we will only use the immediate reward to induce policy in the following two experiments.

## 7.3. Experiment 3: Low Correlation-based Feature Selection

Experiment 3 was conducted in the Spring of 2017, and the goal was to further investigate the effectiveness of our feature selection methods. So the research question for Experiment 3 is Q1 (State): can effective feature selection methods empirically improve the effectiveness of the induced policy?

Results of feature selection showed that the policy with the highest ECR is induced when WIG-Low is applied and the number of selected features is six (see Figure 5 in Section 5), so in Experiment 3 we implemented and empirically evaluated the induced *WIG-det* policy. Table 10 shows the selected features and *WIG-det* policy, which contains only 9 rules associated with PS but 46 rules for WE.

### 7.3.1. Experiment 3: Participants and Conditions

A total of 92 students were randomly assigned into two different groups: *Random* $(N = 45)$ and *WIG-det* $(N = 47)$. In the end, a total of 82 students completed the assignment and were distributed as follows: *Random* $(N = 38)$ and *WIG-det* $(N = 44)$. Pearson's chi-squared test revealed no significant relationship between completion rate and condition $\chi^2(1, N = 92) = .034, p = .852$.

The last row in Table 11 (a) shows the mean and SD for either condition's corresponding learning performance. No significant difference was found between *WIG-det* and *Random* on either pre-test ($F(1, 80) = 2.02$, $p = 0.16$) or transfer post-test ($F(1, 80) = 1.74$, $p = 0.19$). Furthermore, as in Experiments 1 and 2, we perform a median split using the "average response time on level 1" (*level1-avgstepTime*) to split students and find that students whose *level1-avgstepTime* $< 8.34$ sec are less sensitive to the effectiveness of pedagogical strategies

Table 10: WIG-Low Policy

**TotalPSTime** ($f_1$): Total time for solving a problem
**easyProSolved** ($f_2$): Number of easy problems solved
**NewLevel** ($f_3$): Whether current solved problem is in a new level
**avgstepTime** ($f_4$): Average time per step
**hintRatio** ($f_5$): Ratio between hint count and action count
**NumProbRule** ($f_6$): Number of expected rules for the next problem

Note: Black: PS, White: WE, Gray: No mapping from state to action

| First three features $f_{I1}$:$f_{I2}$:$f_{I3}$ \ Last three features $f_{I4}$:$f_{I5}$:$f_{I6}$ | 0:0:0 | 0:0:1 | 0:1:0 | 0:1:1 | 1:0:0 | 1:0:1 | 1:1:0 | 1:1:1 |
|---|---|---|---|---|---|---|---|---|
| 0:0:0 | □ | □ | □ | ■ | □ | □ | □ | ■ |
| 0:0:1 | □ | □ | ■ | □ | □ | □ | ■ | □ |
| 0:1:0 | □ | ■ | □ | □ | □ | □ | □ | ■ |
| 0:1:1 | □ | ▨ | □ | ▨ | □ | ▨ | ▨ | ▨ |
| 1:0:0 | □ | □ | □ | □ | □ | □ | □ | □ |
| 1:0:1 | □ | □ | ■ | □ | □ | □ | □ | □ |
| 1:1:0 | □ | ■ | □ | □ | □ | □ | □ | ■ |
| 1:1:1 | □ | ▨ | □ | ▨ | □ | ▨ | □ | ▨ |

while their peers whose *level1-avgstepTime* $\geq$ 8.34 sec are more sensitive to the effectiveness of pedagogical strategies in Experiment 3. In the following section, we refer the former as the Unresponsive group and the latter as the Responsive group. Combining Policy with Type {Responsive, Unresponsive}, we have a total of four groups including two Responsive groups, *Random-Resp* ($N = 18$) and *WIG-det-Resp* ($N = 22$) and two Unresponsive groups, *Random-Unres* ($N = 20$) and *WIG-det-Unres* ($N = 22$). Pearson's chi-squared test revealed no significant difference in the distribution of Unresponsive vs. Responsive between *Random* and *WIG-det*, $\chi^2(1, N = 82) = 0, p = .987$.

### 7.3.2. Experiment 3: Results

Table 11 presents the mean and SD for students' corresponding learning performance in Experiment 3. One-way ANOVA tests show that no significant difference is found on the pre-test score either between *WIG-det* and *Random*, $F(1, 80) = 2.03$, $p = 0.16$, or between Responsive and Unresponsive groups, $F(1, 80) = 0.67$, $p = 0.42$. Additionally, no significant difference is found either between the two Responsive groups, $F(1, 40) = 0.87$, $p = 0.36$, or between the two Unresponsive groups, $F(1, 38) = 1.21$, $p = 0.28$.

Table 11: Pre-test and Transfer Post-test in Experiment 3

| Type | Pre-Test Score | | Transfer Post-Test Score | |
|---|---|---|---|---|
| | WIG-det | Random | WIG-det | Random |
| Resp | 73.13(25.71) | 63.76(28.02) | **91.76(11.47)** | 75.69(25.26) |
| Unres | 77.63(27.52) | 69.65(27.91) | 85.93(17.35) | 90.31(17.26) |
| Total | 75.38(26.41) | 66.85(27.74) | 88.84(14.83) | 83.38(22.38) |

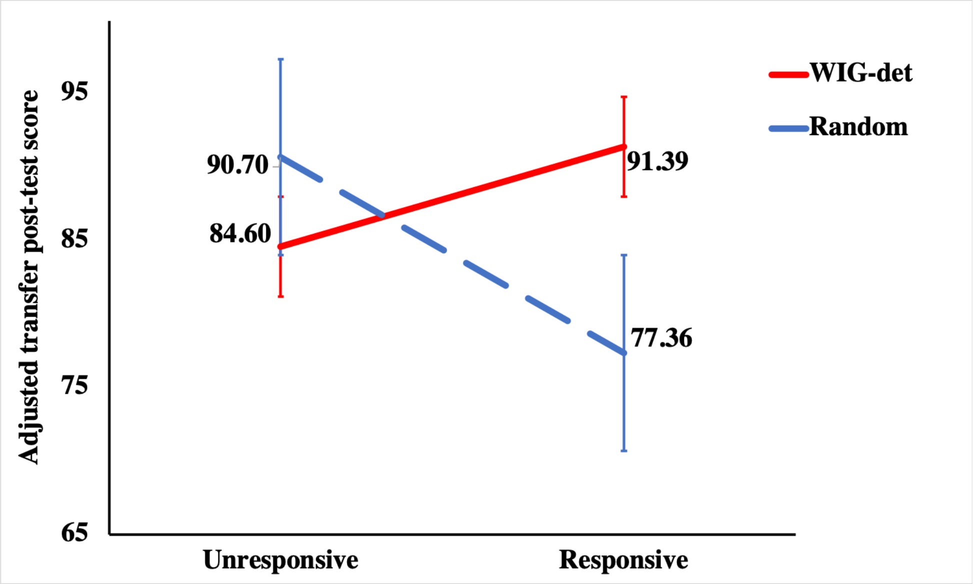


Figure 9: Interaction effect for the adjusted transfer post-test score in Experiment 3

**Transfer Post-Test Score.** A two-way ANCOVA test using Policy and Type as two factors and the pre-test score as covariate shows that there is no significant main effect of either Policy or Type, but there is a significant interaction effect on post-test score, $F(1, 77) = 6.94$,$p = .010$. Figure 9 depicts the cross-over interaction between Policy and Type on the adjusted transfer post-test score.

Furthermore, planned contrasts using Tukey's adjustment indicate a significant difference between the two Responsive groups in that *WIG-Resp* achieved the significantly higher adjusted transfer post-test score than *Random-Resp*, $t(77) = 2.54$, $p = .013$, while there is no significant difference between two Unresponsive groups.

### 7.3.3. Experiment 3: Conclusion

Again results from Experiment 3 shows that there is an ATI effect. The Unresponsive groups are less sensitive to the policies in that they achieve a similar performance on the transfer post-test scores, while the Responsive groups are more sensitive in that their performances are strongly dependent on the effectiveness of the policy. Specifically, the *WIG-det* policy is more effective than the *Random* policy for the Responsive groups.

## 7.4. Experiment 4: Stochastic Policy Execution

In Experiments 1–3, all RL policies were executed deterministically, that is, the action was fully carried out given a state according to the induced RL-policies. However, one classic problem in RL is finding a balance between exploration (discovering more about the world) and exploitation (using what we already know to maximize performance). One approach to improving deterministic policies is to execute them stochastically, where each action is associated with a probability and has a chance to be selected. Therefore, we converted the *WIG-det* policy in Experiment 3 into a stochastic policy, called *WIG-sto*, and conducted Experiment 4 in the Fall of 2017. Our purpose here is to investigate two research questions:

- **Q1 (State):** Can effective feature selection methods empirically improve the effectiveness of the induced policy?
- **Q3 (Execution):** Can stochastic policy execution be more effective than deterministic policy execution?

The crucial part of stochastic policy execution is to assign a probability to each action. Note that in a policy $\pi$, each action $a$ for a particular state $s$ is associated with a Q-value, called $Q^\pi(s,a)$ calculated by using Equation 1 in Section 3. Thus, we transform $Q^\pi(s,a)$ into probability $p^\pi(s,a)$ by the *softmax function* (Sutton and Barto, 1998), shown as follows:

$$p^\pi(s,a) = \frac{e^{\tau \cdot Q^\pi(s,a)}}{\sum_{a'} e^{\tau \cdot Q^\pi(s,a')}} \tag{9}$$

Here $\tau$ is a positive parameter, which controls the variance of probabilities for the state and action pair. Generally speaking, when $\tau \rightarrow 0$, the stochastic policy execution is close to random decision-making. When $\tau \rightarrow +\infty$, the stochastic policy execution becomes deterministic. In order to determine the optimal $\tau$, we use Importance Sampling (Peshkin and Shelton, 2002) which can mathematically evaluate the effectiveness of policies with different $\tau$ values. Specifically, Importance Sampling ($IS$) of a policy $\pi$ is formulated as follows:

$$IS(\pi|\mathcal{D}) = \frac{1}{N_\mathcal{D}} \sum_{i=1}^{N_\mathcal{D}} \left[ \prod_{t=1}^{L^i} \frac{p^\pi(s_t^i, a_t^i)}{p^d(s_t^i, a_t^i)} \cdot (\sum_{t=1}^{L^i} \gamma^{t-1} r_t^i) \right] \tag{10}$$

Where $N_\mathcal{D}$ denotes the number of trajectories in the training corpus $\mathcal{D}$; $L^i$ is the length of the $i$th trajectory; $s_t^i$, $a_t^i$ and $r_t^i$ are the state, action and reward at the $t$th time step of the $i$th trajectory respectively; and $p^d(s_t^i, a_t^i)$ is the probability of taking the action $a_t^i$ for the state $s_t^i$, calculated based on the other policy $d$, which generates the training corpus $\mathcal{D}$. In our case, the decision in the training corpus is randomly decided, thus $p^d(s_t^i, a_t^i)$ always equal to 0.5. In general, the higher value of $IS(\pi|\mathcal{D})$, the better policy $\pi$ is supposed to be.

We explored a wide range of $\tau$ and found that the optimal value of $\tau$ is 0.06 for the MDP-based policies. Moreover, it is important to note that based on Equation 9, for a given state $s$, that if the Q-value of the optimal action $a^*$ is much higher than the Q-values of other alternative suboptimal actions, then the stochastic policy execution becomes deterministic in that the probability of carrying out the optimal action would be closer to 1; if the difference between them is small, then the stochastic policy execution becomes closer to random.

Table 12: Pre-test and Transfer Post-test in Experiment 4

| Type | Pre-Test Score | | Transfer Post-Test Score | |
|---|---|---|---|---|
| | WIG-sto | Random | WIG-sto | Random |
| Resp | 67.43(30.75) | 75.25(26.37) | 95.24(12.49) | 92.79(16.63) |
| Unres | 70.96(25.89) | 68.44(33.02) | 91.07(14.60) | 94.04(12.26) |
| Total | 69.11(28.26) | 72.01(29.58) | 93.25(13.54) | 93.39(14.55) |

### 7.4.1. Experiment 4: Participants and Results

A total 101 of students were randomly split into two distinct groups, *Random* ($N = 51$) and *WIG-sto* ($N = 50$). In the end, a total of 88 students completed the experiment, distributed as *Random* ($N = 44$) and *WIG-sto* ($N = 44$). Pearson's chi-squared test shows that no significant relationship exists between completion rate and condition, $\chi^2(1, N = 101) = 0, p = 1$.

Table 12 presents the mean and SD for students' corresponding learning performance in Experiment 4. There is no significant difference between *WIG-det* and *Random* on either pre-test, $F(1, 86) = 0.22$,$p = 0.64$, or transfer post-test, $F(1, 86) = 0.02$,$p = 0.96$, due to a ceiling effect: about 72.8% of students receive a transfer post-test score of 100. As a result, the *WIG-sto* group scores as high on the transfer post-test as the *Random* group.

Furthermore, as in Experiments 2 and 3, we conduct a median split using the "average response time on level 1" (*level1-avgstepTime*, median = 5.29 sec). Note that this median time is much lower than those used in Experiments 1–3. After splitting, the ceiling effect was found among all four groups of students.

### 7.4.2. Experiment 4: Conclusion

Despite the fact that we used the same DT version, had similar test items in the transfer post-test, and had balanced assignment of students involved in Experiment 4, we found a ceiling effect on the transfer post-test score, which is a significant limitation of Experiment 4. While it is not clear whether the stochastic policy execution would indeed have an effective impact on students' learning performance, it did show that when conducting empirical studies in this domain, we still face many challenges that need to be addressed, especially how to effectively evaluate the induced policies.

## 7.5. Conclusions of Experiments

We investigated the impact of reward function, state representation, and policy execution on the effectiveness of RL-induced policies using the MDP framework. Four experiments were conducted to compare a series of RL-induced policies with that of a Random policy. With the exception of a ceiling effect found in Experiment 4, an ATI effect is consistently observed across Experiments 1–3 after splitting students into the Responsive and Unresponsive groups using their *level1-avgstepTime*. Specifically, the Unresponsive groups are less sensitive to the effectiveness of policies in that they perform similarly to their random peers regardless of the policies, while the Responsive groups are more sensitive to the RL-induced policies.

For the reward function, we found that using Immediate rewards works more effectively than using Delayed rewards in Experiment 2, while no significant difference is found between *Ensemble-Delay-Resp* and *Random-Resp*. For policy execution, unfortunately, we can not determine the effectiveness of the stochastic policy execution due to a ceiling effect on transfer post-test scores.

For the state representation, we find that by combining effective feature selection methods with RL, our MDP-induced policies can be more effective than the random policy for Responsive students: for Experiment 1, while no significant difference was found between the *Random-Res* and *Random-Unre* groups, the *MDP-ECR-Resp* group scores significantly higher than the *MDP-ECR-Unres* group. For Experiment 2, while no significant difference is found among the three Unresponsive groups on the transfer post-test scores, the *Ensemble-Imme-Resp* group scores significantly higher than the *Random-Resp* group. For Experiment 3, again while no significant difference is found among the three Unresponsive groups on the transfer post-test scores, the *WIG-det-Res* group scores significantly higher than the *Random-Res* group.

Despite all these findings, for different experiments students are split into Responsive vs. Unresponsive students using different median split values and criteria: for Experiment 1, we have *level1-avgstepTime* $<$ 7.6 sec as Responsive group vs. *level1-avgstepTime* $\geq$ 7.6 sec as Unresponsive group; for Experiment 2, we have *level1-avgstepTime* $\geq$ 8.01 sec as the Responsive group vs. *level1-avgstepTime* $<$ 8.01 sec as Unresponsive group; and for Experiment 3, we have *level1-avgstepTime* $\geq$ 8.34 sec as Responsive group vs. *level1-avgstepTime* $<$ 8.34 sec as Unresponsive group. Therefore, it is not clear whether the same results will hold if we split them using one global median value and criteria. Additionally, different feature selection methods are applied for inducing different MDP policies in Experiments 1–3. Thus we conduct a post-hoc comparison to explore the impact of different feature selection methods on the effectiveness of the induced policies.

## 8. Post-hoc Comparisons

In Experiments 1–4, students were drawn from the same target population and all of them were enrolled in experiments with the same method but in different semesters. By assigning students to each condition randomly, it provides the most rigorous test of our hypotheses. In this section, we conduct a post-hoc comparison across the four experiments in the hope that this wider view will shed some light on our main results.

Since all Random students followed the Random policy and trained on the same DT tutor across all four experiments, we expect their performance on both pre-test and transfer post-test will reflect whether our students are indeed similar and whether our transfer post-tests are equivalent from semester to semester. A one-way ANOVA test shows that there is no significant difference on the pre-test score among the four Random groups, however a one-way ANCOVA test on Experiment using the pre-test score as covariate shows there is a significant difference among the four Random groups on the transfer post-test scores, $F(3, 127) = 3.60$, $p = .015$. Specifically, post-hoc Tukey HSD tests show that while no significant difference is found among Random groups across Experiments 1–3, the Random group in Experiment 4 scores significantly higher than Random in Experiment 1, $t(127) = -2.89, p = .024$. This suggests that Experiment 4 is significantly different from the first three experiments. Indeed, once we combine the three Random groups across Experiments 1–3 into a large Random group, referred as *Com-Random*, and referring to the Random group in Experiment 4 as *Random4*, a one-way ANCOVA test using the pre-test score as covariate indicates that there is a significant difference, $F(1, 128) = 8.58$, $p = .004$, such that *Random4* ($M = 93.39$, $SD = 14.55$) scores higher on the transfer post-test than *Com-Random* ($M = 79.21$, $SD = 27.96$). Therefore, our post-hoc comparison will only involve Experiments 1–3 and involve five groups: ranking from most recent to the oldest, *WIG-det*, *Ensemble-Imme*, *Ensemble-Delay*, *MDP-ECR*, and *Com-Random* groups.

### 8.1. Global Median Split

While we find that the ATI effect exists in Experiments 1–3, the Responsive and Unresponsive groups are split in different ways for different experiments. In post-hoc comparisons, we explore consistent splitting criteria and investigate whether the same results will hold. For the global median split, we combine all the students in all policy groups who were in our analysis across Experiments 1–3. Particularly, we find that students whose *level1-avgstepTime* $< 8.01$ sec are less sensitive to the effectiveness of pedagogical strategies than their peers whose *level1-avgstepTime* $\geq 8.01$ sec. In the following section, we refer the former as the Unresponsive group and the latter as the Responsive group.

Combining Policy {*WIG-det*, *Ensemble-Imme*, *Ensemble-Delay*, *MDP-ECR*, *Com-Random*} with Type factor {Responsive, Unresponsive}, we have a total of 10 groups. Table 13 shows the number of the students in each group, and a Pearson's chi-squared test indicates that there is no significant difference in the distribution of Responsive vs. Unresponsive among the five policies, $\chi^2(4, N = 239) = 3.10, p = 0.54$.

Table 13: Group Sizes for Post-hoc Comparisons

| Type | Experiment 1<br>*MDP-ECR* | Experiment 2<br>*Ensemble-Imme* | Experiment 2<br>*Ensemble-Delay* | Experiment 3<br>*WIG-det* | Experiments 1,2,3<br>*Com-Random* |
|---|---|---|---|---|---|
| Ures | 27 | 15 | 16 | 20 | 46 |
| Resp | 18 | 18 | 18 | 24 | 41 |

In the post-hoc analysis, we compare the three MDP policies against the *Com-Random* policy to determine the impact of the feature selection methods. All three MDP policies (*WIG-det* *Ensemble-Imme*, and *MDP-ECR*) are induced by applying different feature selection methods with RL using *immediate* rewards, DT-Imme. Additionally, to determine the impact of the reward function we compared the *Ensemble-Imme* and *Ensemble-Delay* against *Com-Random* since the former two use the same feature selection method. For the impact of the reward function, the same patterns are found in the post-hoc comparison as in Experiment 2: while no significant difference is found among the three Unresponsive groups, the *Ensemble-Imme-Resp* significantly out-performs the *Random-Resp* and no significant difference is found between the *Ensemble-Delay-Resp* and *Random-Resp*. Therefore, in the following, we will focus on exploring the impact of the feature selection on RL-induced policies.

## 8.2. The Impact of Feature Selection on RL Policies

Table 14 presents the mean and SD for students' pre-test and transfer post-test scores for eight groups of students: four Policies {*WIG-det*, *Ensemble-Imme*, *MDP-ECR*, *Com-Random*} × 2 Types {Responsive, Unresponsive}. It is important to note that since all students are split using the new global median value, the pre-test and transfer post-test scores are different from those listed in the tables for the individual experiments.

**Pre-test scores.** A two-way ANOVA test using Policy and Type as two factors show that there is no significant main effect of Type, no significant interaction effect of Policy and Type, but a significant main effect of Policy on pre-test score: $F(3, 201) = 3.54, p = .016$. Specifically, planned contrasts using Tukey's adjustment indicate a significant difference between *WIG-det* and *Com-Random* in that the former achieved the significantly higher pre-test score than the later, $t(201) = 3.22$, $p = .009$, while there is no significant difference for other pair of policies.

**Transfer Post-Test Score.** To take the differences among the eight groups on the pretest into account, we run a two-way ANCOVA test, using Policy and Type as the two factors and

Table 14: Pre-test and Transfer Post-test Score across Experiment 1-3

| Policy | Pre-Test Score | | Transfer Post-Test Score | |
|---|---|---|---|---|
| | Unres | Resp | Unres | Resp |
| WIG-det | 80.23(22.64) | 71.34(29.04) | 84.53(17.60) | **92.45(11.21)** |
| Ensemble-Imme | 54.11(37.27) | 62.20(31.84) | 83.33(22.37) | 90.97(24.37) |
| Com-Random | 59.37(32.18) | 71.51(26.25) | 84.10(24.98) | 73.70(30.34) |
| MDP-ECR | 60.48(30.56) | 49.53(31.82) | 87.96(15.97) | 66.32(28.93) |
| Total | 60.91(31.55) | 66.24(29.79) | 85.98(20.24) | 80.12(27.88) |

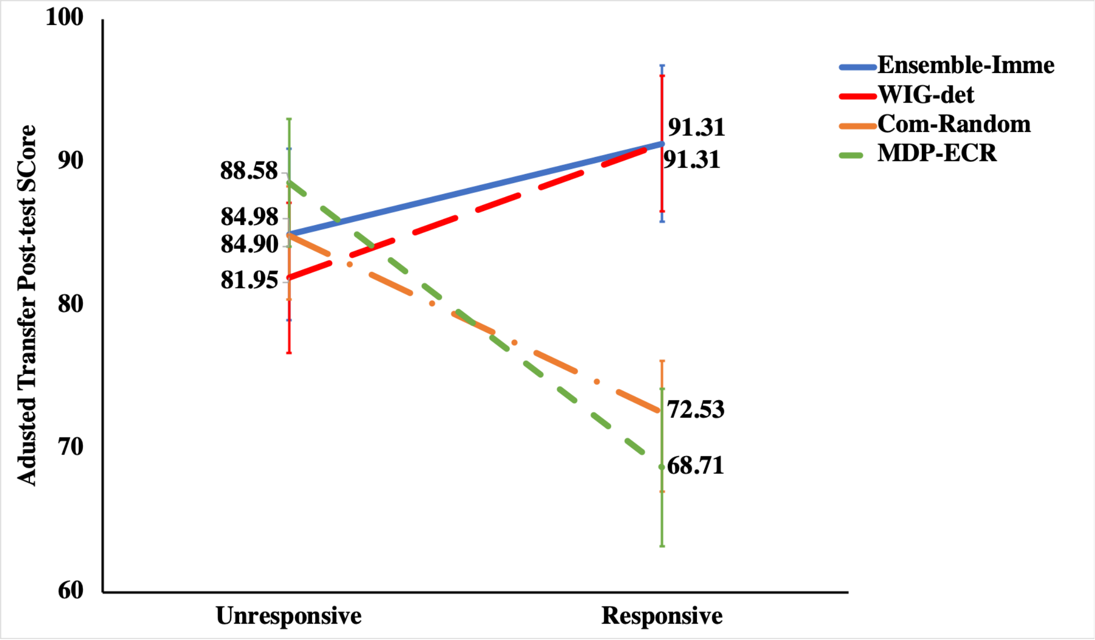


Figure 10: Interaction effect for adjusted transfer post-test score across Experiment 1-3

the pre-test score as covariate. Results show that there is a significant main effect of Type, $F(1,200) = 3.91$, $p = .049$, and a significant interaction effect of Policy × Type on transfer post-test scores, $F(3,200) = 4.22$, $p = .006$. The interaction is shown in Figure 10, which presents the mean and standard error of adjusted transfer post-test score for each group, which is the transfer post-test score adjusted by the linear regression model built to describe the relation between the pre- and transfer post-test scores.

Table 15 presents the results of contrast tests using Tukey's adjustment for multiple comparisons. Results indicate that *WIG-det-Resp* scored significantly higher than *MDP-ECR-Resp* and *Random-Resp*: $p = .022$ and $p = .027$ respectively; *Ensemble-Imme-Resp* achieved higher score than *MDP-ECR-Resp* and *Random-Resp*, where the difference is significant $p = .045$ and marginally significant $p = .054$ respectively. No significant difference is found between other pairs.

Table 15: Pairwise Contrasts on Adjusted Transfer Post-test in Post-hoc Comparison

| Pairwise Policy Comparison | | | $t(200)$ | $p$-value |
|---|---|---|---|---|
| WIG-det-**Resp** | vs. | Random-**Resp** | 3.16 | 0.022 * |
| WIG-det-**Resp** | vs. | MDP-ECR-**Resp** | 3.09 | 0.027 * |
| Ensemble-Imme-**Resp** | vs. | Random-**Resp** | 2.86 | 0.054 · |
| Ensemble-Imme-**Resp** | vs. | MDP-ECR-**Resp** | 2.92 | 0.045 * |
| WIG-det-**Resp** | vs. | Ensemble-Imme-**Resp** | 0.00 | 1.00 |
| MDP-ECR-**Resp** | vs. | Random-**Resp** | 0.57 | 1.00 |
| WIG-det-**Unres** | vs. | Random-**Unres** | 0.47 | 1.00 |
| WIG-det-**Unres** | vs. | MDP-ECR-**Unres** | 0.96 | 0.99 |
| Ensemble-Imme-**Unres** | vs. | Random-**Unres** | 0.01 | 1.00 |
| Ensemble-Imme-**Unres** | vs. | MDP-ECR-**Unres** | 0.48 | 1.00 |
| WIG-det-**Unres** | vs. | Ensemble-Imme-**Unres** | 0.38 | 1.00 |
| MDP-ECR-**Unres** | vs. | Random-**Unres** | 0.66 | 0.99 |

· marginally significant at $p < 0.1$; * significant at $p < 0.05$.

**Conclusion.** We find that the ATI effect exists in the post-hoc comparisons. Specifically, the Unresponsive groups are less sensitive to the effectiveness of policies since they achieve similar transfer post-test scores, whereas the Responsive groups are more sensitive in that their learning performance is significantly dependent on the policy. Specifically, the *WIG-det* policy outperforms the *MDP-ECR* and *Random* policies in terms of transfer post-test score for the responsive students. Results suggest that the WIG-Low and possibly the Ensemble feature selection approaches can facilitate the MDP inducing more effective policies than the Random policy, while the ECR-based feature selection approach cannot be as effective as the former two approaches.

### 8.3. Problem Solving vs. Worked Example under policies

Table 16(a) presents the mean and SD of PS and WE count decided by policies across Experiment 1-3, and Table 16(b) shows results of one-way ANOVA tests between Responsive and Unresponsive under each policy condition. One-way ANOVA tests show that the significant difference on PS Count only exists between *Ensemble-Imme-Unres* and *Ensemble-Imme-Resp* in that the former assigned the significantly more PS than the later, while there is no significant difference on both PS and WE counts between Responsive and Unresponsive group under other four policies.

Additionally, Table 16(c) shows results of the Tukey HSD tests for each pairwise policy comparison under each group type {Responsive, Unresponsive, Total}. Particularly, *Ensemble-Imme* has the significantly different PS and WE counts comparing with the other four policies, among which there are some significant differences on WE Count instead of PS Count. For *Total* groups without splitting students into *Responsive* and *Unresponsive*, *WIG-det* and *Com-Random* assigned the significant more WE than both *Ensemble-Delay* and *MDP-ECR*. For *Unresponsive* groups, *WIG-det-Unres* had the significant more WE than *MDP-ECR-Unres*, and *Com-Random-*

Table 16: PS and WE Counts and Comparisons for each Policy across Experiment 1-3

(a): PS and WE Count for each group

| Policy | PS Count | | | WE Count | | |
|---|---|---|---|---|---|---|
| | Unres | Resp | Total | Unres | Resp | Total |
| WIG-det | 6.22(1.21) | 6.04(1.23) | 5.81(0.99) | 6.22(0.94) | 6.25(0.79) | 6.23(0.85) |
| Ensemble-Imme | 2.46(1.85) | 1.27(0.59) | 1.96(1.44) | 9.08(0.86) | 9.13(0.52) | 9.11(0.68) |
| Ensemble-Delay | 5.33(0.65) | 5.89(1.36) | 5.57(1.03) | 5.33(0.65) | 5.00(1.12) | 5.19(0.87) |
| Com-Random | 5.37(1.79) | 5.97(1.68) | 5.65(1.76) | 6.23(1.19) | 5.89(1.05) | 6.07(1.13) |
| MDP-ECR | 5.82(0.95) | 5.80(1.08) | 6.12(1.21) | 5.00(0.35) | 5.20(0.86) | 5.09(0.64) |

(b): Unresponsive vs. Responsive comparison results for each policy

| Policy | One-way ANOVA Tests | |
|---|---|---|
| | PS Count | WE Count |
| WIG | $F(1,40) = 0.22,\quad p = 0.64$ | $F(1,40) = 0.01,\quad p = 0.92$ |
| Ensemble-Imme | $F(1,26) = 5.6,\quad p = \mathbf{0.025}$ | $F(1,26) = 0.05,\quad p = 0.83$ |
| Ensemble-Delay | $F(1,19) = 1.54,\quad p = 0.23$ | $F(1,19) = 0.74,\quad p = 0.4$ |
| Com-Random | $F(1,80) = 2.44,\quad p = 0.12$ | $F(1,80) = 1.81,\quad p = 0.18$ |
| MDP-ECR | $F(1,30) = 0.004,\quad p = 0.95$ | $F(1,30) = 0.77,\quad p = 0.39$ |

(c): Tukey multiple comparison results among policies

| Pairwise Policy Comparison | | | PS Count (p-value) | | | WE Count (p-value) | | |
|---|---|---|---|---|---|---|---|---|
| | | | Unres | Resp | Total | Unres | Resp | Total |
| Ensemble-Imme | vs. | WIG-det | **<1e-5** | **<1e-5** | **<1e-5** | **<1e-5** | **<1e-5** | **<1e-5** |
| Ensemble-Imme | vs. | Ensemble-Delay | **9.6e-4** | **<1e-5** | **<1e-5** | **<1e-5** | **1e-05** | **<1e-5** |
| Ensemble-Imme | vs. | Com-Random | **7.6e-4** | **<1e-5** | **<1e-5** | **<1e-5** | **<1e-5** | **<1e-5** |
| Ensemble-Imme | vs. | MDP-ECR | **1.6e-4** | **<1e-5** | **<1e-5** | **<1e-5** | **<1e-5** | **<1e-5** |
| WIG-det | vs. | Ensemble-Delay | 0.15 | 1.0 | 0.67 | **0.049** | 0.10 | **5.4e-4** |
| WIG-det | vs. | Com-Random | 0.37 | 1.0 | 0.91 | 1.0 | 1.0 | 1.0 |
| WIG-det | vs. | MDP-ECR | 1.0 | 1.0 | 1.0 | **3.9e-4** | **0.007** | **<1e-5** |
| Ensemble-Delay | vs. | Com-Random | 1.0 | 1.0 | 1.0 | **0.016** | 0.49 | **0.004** |
| Ensemble-Delay | vs. | MDP-ECR | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| Com-Random | vs. | MDP-ECR | 1.0 | 1.0 | 1.0 | **<1e-5** | 0.18 | **<1e-5** |

Bold value indicates the significant difference at $p < 0.05$.

*Unres* had the significant more WE than both *Ensemble-Delay-Unres* and *MDP-ECR-Unres*. For *Responsive* groups, *WIG-det-Resp* assigned the significant more WE than both *Ensemble-Delay-Resp* and *MDP-ECR-Resp*.

As the summary, although the PS and WE counts reflect the difference of policies, it is not the key reason why *Ensemble-Imme* and *WIG-det* policies outperform *Random*, which requires further data analysis.

## 9. Conclusions, Limitations, & Discussion

We conducted four experiments to investigate the effectiveness of reinforcement learning induced policies using the MDP framework. Overall, an aptitude-treatment interaction effect consistently exists among Experiments 1–3 and the post-hoc comparisons. Furthermore, our students were split based on their response time, and we found the Unresponsive groups have similar learning performance under different policies employed by the ITS, whereas Responsive groups are more sensitive to the induced policies in that those under an effective policy would perform significantly better than those under an ineffective policy.

When applying RL to induce policies, we explored the impact of reward function, state representation, and policy execution. For policy execution, no significant improvement was found for the stochastic policy execution due to a ceiling effect. For future studies, the ceiling effect may be eliminated if we assign harder questions to students during the transfer post-test and adjust the grading rubric for the post-test to provide more finely grained evaluation and continuous scores.

In many domains, RL is applied with an immediate reward function. For example, in an automatic call center system, the agent can receive an immediate reward for every question it asks because the impact of each question can be assessed instantaneously (Williams, 2008). Immediate rewards are often chosen for RL-based policy induction because it is easier to assign appropriate credit or blame when the feedback is tied to a single decision. The more that rewards or punishments are delayed, the harder it becomes to properly assign credit or blame. However, for an ITS, the most appropriate rewards to use are student learning gains, which are typically unavailable until the entire tutoring session is complete. This is due to the complex nature of the learning process, making it difficult to assess student learning moment by moment. More importantly, many instructional interventions that boost short-term performance may not be effective over the long-term; for example, an instructional intervention may reduce the time a student spends solving a problem, but may also lead to shallow learning of the material (Baker et al., 2004). We explored both immediate and delayed rewards in our policy induction and empirically evaluated the impact of the induced policies on student learning. Our results show that using immediate rewards can be more effective than using delayed rewards, probably because of the vanishing reward problem: the discount factor in the MDP framework makes the rewards in the early decisions become extremely small with respect to the delayed reward.

For state representation, we explored feature selection based on the MDP framework. Although many feature selection methods such as embedded incremental feature selection (Wright et al., 2012), *LSPI* (Li et al., 2009), and *Neighborhood Component Analysis* (Goldberger et al., 2005) can be applied to RL, most of these methods are designed for *model-free* RL, and we focus on *model-based* RL due to the high cost of collecting training data on ITSs. While correlation-based feature selection methods have been widely used for supervised learning for selecting the most relevant state features to the output label (Hall, 1999; Yu and Liu, 2003), in this work we explored five correlation-based metrics with two options: one option is to select the next feature that is the **most correlated (High)** to the currently selected feature set whereas the other option is to select the **least correlated (Low)**. Choosing the most correlated feature may be effective since the feature is more likely to be related to decision making; however, it may not make much more of a contribution than the currently selected feature set. Alternatively, choosing the least correlated feature may raise the diversity of the feature set, enriching the state representation; however, this has the risk of selecting irrelevant or noisy features. Our results show that low correlation methods perform significantly better than high correlation methods, the RL-based approach from our previous work (Chi et al., 2011), and the baseline random method in terms of the expected cumulative reward (ECR). In particular, low correlation methods improve over high correlation methods as much as 142.48%, with an average of 45.2% improvement in ECR. In general, we have: Low correlation-based $>$ Ensemble $>$ High correlation-based $>$ ECR-based $\gg$ Random (Sec. 5.6).

Empirical results from Experiments 2 and 3 show that by applying effective feature selection to MDP, the Responsive groups using an RL-induced policy can significantly outperform their peers using a random policy. Additionally, post-hoc comparison results (Sec. 8.2) show that the empirical effectiveness of policies can be ordered as: *WIG-det* $>$ *MDP-ECR*, *Random* (Sec. 8.2). Therefore, our results suggest that a low correlation-based feature selection approach is more effective than other feature selection methods for RL.

There are several caveats in our experiments that provide enlightenment regarding future work. First of all, we retrospectively split students into Responsive vs. Unresponsive groups using response time because we do not fully understand why the differences between Responsive vs. Unresponsive groups exist. To answer such a question, we need to perform deep log analysis for our future work. Second, although we detect different performance among the different RL-induced policies, it is still not clear what makes them effective or why they are effective. Future work is needed to shed some light on understanding the induced policies and to compare the machine induced policies with existing learning theory. Third, we mainly compare the RL-induced policies with a Random policy in our experiments and it is not clear if the same results would hold if we compare them against a stronger baseline such as those used in previous research (McLaren and Isotani, 2011; McLaren et al., 2014; Najar et al., 2014; Salden et al., 2010). Finally, in this work, we selected a small set of features from 133 observable state features which severely limits the effectiveness of tabular MDP methods. Many of the relevant factors such as motivation, affect, and prior knowledge, cannot be observed directly nor are they described explicitly. On the other hand, Partially-observable MDPs (POMDPs) model unobserved factors by using a belief state space. Thus POMDPs for ITSs can explicitly represent two sources of uncertainty: non-determinism in the control process and partial observability of the students' knowledge levels. In the former case the outcome of the tutorial actions and the students' knowledge levels are represented by a probability distribution, and in the latter case, the underlying knowledge levels are observed indirectly via incomplete or imperfect observations. In short, using the belief state space gives POMDP two potential advantages over MDPs: better handling of uncertainty in the state representation, and the ability to incorporate a large range of state features. As a result, we believe that POMDPs will be more effective than tabular MDPs for ITSs.

Furthermore, previous work (Renkl, 2002; Gerjets et al., 2006; Taylor et al., 2006; Atkinson et al., 2003) has shown that adding self-explain steps in WE and PS (prompting for self-explanation) can significantly improve students learning. In the future, we will expand our research scope on not only WE vs. PS but also on whether or not to ask students to self-explain.

## ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1432156 "Educational Data Mining for Individualized Instruction in STEM Learning Environments", 1651909 "CAREER: Improving Adaptive Decision Making in Interactive Learning Environments", and 1726550 "Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments".

## EDITORIAL STATEMENT

This article was originally submitted to the EDM 2018 Journal Track Special Issue. Min Chi had no involvement with the journal's handling of this article in order to avoid a conflict with her Special Track Editor role. The review process was originally managed by Special Track Editor Irena Koprinska with oversight from JEDM Editor Andrew Olney and later exclusively by Andrew Olney when the deadline passed for the EDM 2018 Journal Track Special Issue.

## REFERENCES

ATKINSON, R. K., RENKL, A., AND MERRILL, M. M. 2003. Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps. *Journal of Educational Psychology 95,* 4, 774–783.

BACH, F. R. 2009. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems*. 105–112.

BAKER, R. S., CORBETT, A. T., KOEDINGER, K. R., AND WAGNER, A. Z. 2004. Off-task behavior in the cognitive tutor classroom: When students game the system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 383–390.

BECK, J., WOOLF, B. P., AND BEAL, C. R. 2000. Advisor: A machine learning architecture for intelligent tutor construction. *AAAI/IAAI 2000,* 552-557, 1–2.

BEHROOZ, M. AND TIFFANY, B. 2017. Evolution of an intelligent deductive logic tutor using data-driven elements. *International Journal of Artificial Intelligence in Education 27,* 1, 5–36.

BROWN, J. S., COLLINS, A., AND DUGUID, P. 1989. Situated cognition and the culture of learning. *Educational Researcher 18,* 1, 32–42.

CHANDRASHEKAR, G. AND SAHIN, F. 2014. A survey on feature selection methods. *Computers & Electrical Engineering 40,* 1, 16–28.

CHI, M. AND VANLEHN, K. 2010. Meta-cognitive strategy instruction in intelligent tutoring systems: How, when, and why. *Journal of Educational Technology & Society 13,* 1, 25–39.

CHI, M., VANLEHN, K., LITMAN, D., AND JORDAN, P. 2011. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction 21,* 1-2, 137–180.

CLEMENT, B., OUDEYER, P.-Y., AND LOPES, M. 2016. A comparison of automatic teaching strategies for heterogeneous student populations. In *Proceedings of the 9th International Conference on Educational Data Mining*, T. Barnes, M. Chi, and M. Feng, Eds. 330–335.

CRONBACH, L. J. AND SNOW, R. E. 1977. *Aptitudes and instructional methods: A handbook for research on interactions.* Oxford, England: Irvington.

D'MELLO, S., LEHMAN, B., SULLINS, J., DAIGLE, R., COMBS, R., VOGT, K., PERKINS, L., AND GRAESSER, A. 2010. A time for emoting: When affect-sensitivity is and isn't effective at promoting deep learning. In *International Conference on Intelligent Tutoring Systems*. Springer, 245–254.

GERJETS, P., SCHEITER, K., AND CATRAMBONE, R. 2006. Can learning from molar and modular worked examples be enhanced by providing instructional explanations and prompting self-explanations? *Learning and Instruction 16,* 2, 104–121.

GOLDBERGER, J., HINTON, G. E., ROWEIS, S. T., AND SALAKHUTDINOV, R. R. 2005. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*. 513–520.

GONZÁLEZ-ESPADA, W. J. AND BULLOCK, D. W. 2007. Innovative applications of classroom response systems: Investigating students' item response times in relation to final course grade, gender, general point average, and high school act scores. *Electronic Journal for the Integration of Technology in Education 6*, 97–108.

HALL, M. A. 1999. Correlation-based feature selection for machine learning. Ph.D. thesis, The University of Waikato.

IGLESIAS, A., MARTÍNEZ, P., ALER, R., AND FERNÁNDEZ, F. 2009a. Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. *Applied Intelligence 31,* 1, 89–106.

IGLESIAS, A., MARTÍNEZ, P., ALER, R., AND FERNÁNDEZ, F. 2009b. Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems 22,* 4, 266–270.

IGLESIAS, A., MARTÍNEZ, P., AND FERNÁNDEZ, F. 2003. An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. *Informatics in Education 2*, 223–240.

JAAKKOLA, T., SINGH, S. P., AND JORDAN, M. I. 1995. Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in Neural Information Processing Systems*. 345–352.

KALYUGA, S., AYRES, P., CHANDLER, P., AND SWELLER, J. 2003. The expertise reversal effect. *Educational psychologist 38,* 1, 23–31.

KELLER, P. W., MANNOR, S., AND PRECUP, D. 2006. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 449–456.

KENT, J. T. 1983. Information gain and a general measure of correlation. *Biometrika 70,* 1, 163–173.

KOEDINGER, K. R. AND ALEVEN, V. 2007. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review 19,* 3, 239–264.

KOENIG, S. AND SIMMONS, R. 1998. Xavier: A robot navigation architecture based on partially observable Markov decision process models. *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, 91–122.

KOLTER, J. Z. AND NG, A. Y. 2009. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 521–528.

KOPRINSKA, I., RANA, M., AND AGELIDIS, V. G. 2015. Correlation and instance based feature selection for electricity load forecasting. *Knowledge-Based Systems 82*, 29–40.

LEE, C. AND LEE, G. G. 2006. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing & Management 42,* 1, 155–165.

LI, L., WILLIAMS, J. D., AND BALAKRISHNAN, S. 2009. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *10th Annual Conference of the International Speech Communication Association*. 2475–2478.

LITTMAN, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings*. Elsevier, 157–163.

LUCE, R. D. ET AL. 1986. *Response times: Their role in inferring elementary mental organization*. Number 8. Oxford University Press on Demand.

MANDEL, T., LIU, Y.-E., LEVINE, S., BRUNSKILL, E., AND POPOVIC, Z. 2014. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*. 1077–1084.

MARTIN, K. N. AND ARROYO, I. 2004. Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*. 564–572.

MCHUGH, M. L. 2013. Chi-squared test of independence. *Biochem Med (Zagreb) 23,* 2, 105–133.

MCLAREN, B. M. AND ISOTANI, S. 2011. When is it best to learn with all worked examples? In *International Conference on Artificial Intelligence in Education*. Springer, 222–229.

MCLAREN, B. M., LIM, S.-J., AND KOEDINGER, K. R. 2008. When and how often should worked examples be given to students? New results and a summary of the current state of research. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*. 2176–2181.

MCLAREN, B. M., VAN GOG, T., GANOE, C., YARON, D., AND KARABINOS, M. 2014. Exploring the assistance dilemma: Comparing instructional support in examples and problems. In *Intelligent Tutoring Systems*. Springer, 354–361.

MNIH, V., KAVUKCUOGLU, K., SILVER, D., GRAVES, A., ANTONOGLOU, I., WIERSTRA, D., AND RIEDMILLER, M. 2013. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

MNIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A. A., VENESS, J., BELLEMARE, M. G., GRAVES, A., RIEDMILLER, M., FIDJELAND, A. K., OSTROVSKI, G., ET AL. 2015. Human-level control through deep reinforcement learning. *Nature 518,* 7540, 529–533.

MOSTAFAVI, B., ZHOU, G., LYNCH, C., CHI, M., AND BARNES, T. 2015. Data-driven worked examples improve retention and completion in a logic tutor. In *Artificial Intelligence in Education*. Springer, 726–729.

NAJAR, A. S., MITROVIC, A., AND MCLAREN, B. M. 2014. Adaptive support versus alternating worked examples and tutored problems: Which leads to better learning? In *User Modeling, Adaptation, and Personalization*. Springer, 171–182.

NARASIMHAN, K., KULKARNI, T., AND BARZILAY, R. 2015. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*.

PESHKIN, L. AND SHELTON, C. R. 2002. Learning from scarce experience. *arXiv preprint cs/0204043*.

RAFFERTY, A. N., BRUNSKILL, E., GRIFFITHS, T. L., AND SHAFTO, P. 2016. Faster teaching via pomdp planning. *Cognitive Science 40,* 6, 1290–1332.

RAZZAQ, L. M. AND HEFFERNAN, N. T. 2009. To tutor or not to tutor: That is the question. In *Artificial Intelligence in Education*. 457–464.

RENKL, A. 2002. Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction 12,* 5, 529–556.

RENKL, A., ATKINSON, R. K., MAIER, U. H., AND STALEY, R. 2002. From example study to problem solving: Smooth transitions help learning. *The Journal of Experimental Education 70,* 4, 293–315.

SALDEN, R. J., ALEVEN, V., SCHWONKE, R., AND RENKL, A. 2010. The expertise reversal effect and worked examples in tutored problem solving. *Instructional Science 38,* 3, 289–307.

SCHNIPKE, D. L. AND SCRAMS, D. J. 2002. Exploring issues of examinee behavior: Insights gained from response-time analyses. *Computer-based testing: Building the foundation for future assessments*, 237–266.

SHEN, S. AND CHI, M. 2016. Reinforcement learning: The sooner the better, or the later the better? In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 37–44.

SNOW, R. E. 1991. Aptitude-treatment interaction as a framework for research on individual differences in psychotherapy. *Journal of Consulting and Clinical Psychology 59,* 2, 205–216.

STAMPER, J., EAGLE, M., BARNES, T., AND CROY, M. 2013. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education 22,* 1-2, 3–17.

SUTTON, R. S. AND BARTO, A. G. 1998. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.

TAYLOR, R. S., O'REILLY, T., SINCLAIR, G. P., AND MCNAMARA, D. S. 2006. Enhancing learning of expository science texts in a remedial reading classroom via istart. In *Proceedings of the 7th International Conference on Learning Sciences*. International Society of the Learning Sciences, 765–770.

TETREAULT, J. R., BOHUS, D., AND LITMAN, D. J. 2007. Estimating the reliability of MDP policies: A confidence interval approach. In *Proceedings Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 276–283.

TETREAULT, J. R. AND LITMAN, D. J. 2008. A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Communication 50,* 8, 683–696.

VAN GOG, T., KESTER, L., AND PAAS, F. 2011. Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology 36,* 3, 212–218.

VYGOTSKY, L. 1978. Interaction between learning and development. *Readings on the development of children 23,* 3, 34–41.

WANG, P., ROWE, J., MIN, W., MOTT, B., AND LESTER, J. 2017. Interactive narrative personalization with deep reinforcement learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. 3852–3858.

WHITEHILL, J. AND MOVELLAN, J. 2018. Approximately optimal teaching of approximately optimal learners. *IEEE Transactions on Learning Technologies 11,* 2, 152–164.

WILLIAMS, J. D. 2008. The best of both worlds: Unifying conventional dialog systems and POMDPs. In *Ninth Annual Conference of the International Speech Communication Association*. 1173–1176.

WRIGHT, R., LOSCALZO, S., AND YU, L. 2012. Embedded incremental feature selection for reinforcement learning. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*. Vol. 1. 263–268.

YANG, Y. AND PEDERSEN, J. O. 1997. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning*. Vol. 97. 412–420.

Yu, L. and Liu, H. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *International Conference on Machine Learning*. Vol. 3. 856–863.