Policy Shaping from Simulated Critique in Domains with Multiple Optimal Policies

Himanshu Sahni, Brent Harrison, Kaushik Subramanian,
Thomas Cederborg, Charles Isbell, Andrea Thomaz
Georgia Institute of Technology

Abstract

In many domains, there exist multiple ways for an agent to achieve optimal performance. Feedback may be provided along one or more of them to aid learning. In this work, we evaluate the interactive reinforcement learning algorithm Policy Shaping in domains with multiple optimal policies. We codify different feedback strategies as automated oracles and analyze their effect on the agent's learning performance. Our experiments show that the best feedback strategy depends on certain domain characteristics, such as risk during exploration, number of optimal policies and stochasticity. The feedback strategy employed has a statistically significant effect on the agent performance across domains. Thus, it is important to take these considerations into account while learning from simulated oracles or human critique.

1 Introduction

In real world environments, an expert may not always be at hand to train the behavior of a human-guided learning agent. Thus, such agents must be able to incorporate feedback from non-expert teachers. There are several ways of incorporating such feedback into reinforcement learning [Isbell *et al.*, 2001; Knox and Stone, 2008; Thomaz and Breazeal, 2008; Knox and Stone, 2010]. Policy shaping [Griffith *et al.*, 2013] attempts to use feedback directly as policy advise and combine it with standard reinforcement learning approaches such as Bayesian Q-learning [Dearden *et al.*, 1998]. The algorithm was recently shown be robust to noisy and sparse human feedback [Cederborg *et al.*, 2015]. For a specific pac-man domain, human teachers were even shown to outperform a simulated teacher which only approved of actions consistent with one fixed optimal policy.

A complicated task in a real-world environment will likely have multiple optimal ways of solving it. A non-expert, end user critiquing the agent may provide feedback for all of them at the same time. Or they may choose to ignore all but one in order to simplify their task of teaching the agent. A human teacher may also try to adapt their feedback to encourage more of what the agent is already doing correctly. It has been shown that humans may differ in preference over feedback strategy when faced with the same task [Sahni *et al.*, 2016]. Our hypothesis is that which feedback strategy performs the best depends on characteristics of the domain. A teacher's feedback strategy may guide the agent towards a policy that takes longer to learn than others, or accumulates higher cost during learning, even though it may be equally optimal. The result may be systems that spend precious human interactions on learning redundancies not required for completing the task, leading to frustrating human-agent interfaces. Therefore, a systematic understanding of which feedback strategies lead to best learning performance in which domains is important. Such an investigation will lend us greater insight into designing oracles that provide automated feedback. It may also aid in designing learning from critique algorithms by taking into account differing feedback strategies.

In this work, we study this effect closely in four domains with multiple optimal solutions. We focus on the following domain characteristics:

1. the risk of high negative reward during exploration,
2. number of optimal solutions to the task and,
3. stochasticity in starting location and action selection.

Our results show that there is a statistically significant difference in performance of different feedback strategies along these characteristics. We carefully designed the domains to control for these characteristics and show that performance of the feedback strategies is domain dependent. We also employed automated oracles as the teachers to allow us exact control over what feedback strategy is used.

In the following sections, we provide some background on reinforcement learning and policy shaping, describe the different teaching strategies we consider, the domains and algorithms we use, and finally present our conclusions.

2 Background

Before proceeding further, we provide a brief introduction to reinforcement learning and policy shaping.

2.1 Reinforcement Learning and MDPs

Reinforcement learning addresses the problem of choosing behavior that maximizes some notion of long term cumulative reward. It is typically formulated as a Markov decision process (MDP). An MDP is characterized by the tuple

$< S, A, T, R, \gamma >$. S is the set of states an agent can be in. $A(S)$ is the set of actions the agent has available to it in each state. Typically, the agent chooses an action to execute from A, which may lead it into a new state according to the transition function $T : S \times A \rightarrow P(S)$. $R : S \times A \rightarrow \mathbb{R}$ is a scalar value received upon executing an action in a state. Finally, $0 \leq \gamma \leq 1$ is the discount factor.

A policy, $\pi : S \rightarrow P(A)$, informs the agent on which action to execute in each state. The goal of a reinforcement learning agent is to find π^*, the optimal policy, which maximizes the long term expected reward, or utility, in each state. Note that π^* does not have to be unique, i.e. there can be multiple actions in a state with the same utility.

2.2 Policy Shaping

Policy Shaping [Griffith *et al.*, 2013] extracts policy level information directly from feedback and combines it with traditional, environment based signals. It considers teacher feedback as an evaluation of the most recent action decision made by the agent. Thus, the feedback on an action only extends to the state in which it was taken.

A major component of Policy Shaping is called Advise. Advise contains a model of the quality of feedback provided by the teacher. In this work, the agent models this as a matrix of probabilities, C. Throughout our experiments, we keep these probabilities constant, as specified in Table 1. Note that this is the agent's belief of the reliability of the teacher's feedback, and does not impact how the feedback is actually generated. A property of the teacher, on the other hand, is their frequency of providing feedback. This is controlled by a parameter L, which is the probability that the teacher will provide feedback at a given state. This parameter is usually not known to the learner, which just receives feedback as it learns. We keep L fixed at 0.3 in all our experiments to control for feedback frequencies across different teachers.

	f_1 = good	f_2 = bad
optimal	$P(f_1\|\text{opt}) = 0.7$	$P(f_2\|\text{opt}) = 0.3$
non-optimal	$P(f_1\|\neg\text{opt}) = 0.2$	$P(f_2\|\neg\text{opt}) = 0.6$

Table 1: The C matrix, with separate probabilities for optimal and non-optimal actions. f_1 and f_2 represent positive and negative critique respectively. $P(f_1|\text{opt})$ is the probability of observing feedback f_1 given that the action was optimal.

Policy shaping makes use of Advise in combination with another algorithm for learning from environment signals. We use tabular Q-Learning with Boltzmann exploration strategy as our environment learning algorithm due to its speed and simplicity [Watkins, 1989] .

2.3 Policy Shaping with Humans

In a study with 26 human participants and a simulated teacher based on a policy that always won, Cederborg et al. discovered that human provided critique led to better performance than the simulated teacher on two different pac-man domains [Cederborg *et al.*, 2015]. This was a surprising result due to the many apparent shortcomings of human critique, such as trying to evaluate imagined future actions, pressing the wrong buttons accidentally, looking away from the screen during the experiment, and so on. The simulated oracle, on the other hand, was highly likely to provide correct feedback in a state. This was because it was constructed by running a Q-learning algorithm with Boltzmann exploration until convergence, and assigning positive critique to the highest value action in each state visited by the agent. This critique was then made available to the agent during testing from the beginning.

Cederborg et al. state that the difference in performance could be because humans seemed to give positive feedback to any strategy that appeared optimal. The simulated teacher, on the other hand, had a single fixed optimal policy that was computed beforehand. Sahni et al.'s study on humans providing feedback in a gridworld with two optimal policies shows that some participants indeed encourage the agent to learn both optimal policies [Sahni *et al.*, 2016]. It may also be the case that certain optimal policies can be learnt faster and the human teachers were guiding the agent towards them. Cederborg et al.'s experimental setup was not designed to verify these conjectures, but instead focused on testing policy shaping with human teachers. The question of whether it is better to provide feedback along all optimal policies simultaneously, or only certain ones remains open. This work addresses this question by explicitly constructing different feedback strategies for handling multiple optimal policies using automated oracles and studying their performance on different domains.

3 Related Work

Interactive machine learning is emerging as a promising field with many useful applications, such as machine teaching [Zhu, 2015], human robot interaction (HRI) [Chernova and Thomaz, 2014], and interactive storytelling [Chi and Lieberman, 2011]. Chen et al. [Chen *et al.*, 2013] and Lazewatsky et al. [Lazewatsky and Smart, 2014] explore interfaces for human interaction with robots for people with disabilities. Lu and Smart akin HRI to theater and propose it as an evaluation platform [Lu and Smart, 2011]. In the context of reinforcement learning, combining human input with environment based signals has been a challenging problem. Prior work has attempted to convert feedback signals into rewards, akin to those coming from the environment [Ng *et al.*, 1999]. However, human feedback may be noisy, inconsistent, and not easily translated into rewards (how much reward should be associated with "good job" vs. "that was amazing!"?). An alternative solution has been to use feedback to influence the policy, instead of the reward signal. For example, this has been done by adding the option to reverse an agent's actions [Thomaz and Breazeal, 2008], or in the TAMER framework by learning the human reinforcement function in a supervised manner [Knox and Stone, 2010]. Policy shaping also attempts to sidestep this issue by using the feedback to directly modify the policy [Griffith *et al.*, 2013].

This work explores the effects of using policy shaping in situations where there are multiple optimal ways of behaving. Teachers may have different preferences over which policy they would like the agent to learn. The effect of teaching preference on learning performance in a different con-

text has been examined by Loftin et al. [Loftin *et al.*, 2014]. They designed the I-SABL algorithm to automatically learn the teacher's preference over remaining silent vs. providing critique simply by observing their feedback behaviour. They perform two user studies on a contextual bandit task. By taking the teacher's preference to remain silent into account, they were able to show improved performance over two state of the art algorithms. In our work, we focus on studying performance of simulated feedback strategies as certain characteristics of the domain are varied. We build upon the study by Sahni et al., in which they showed humans have differing feedback preferences over the same domain characteristics [Sahni *et al.*, 2016].

The idea of different optimal policies having different learning performances has been introduced before. A modification to Q-learning, $\bar{Q}$-learning, performs optimally when combined with an exploration strategy [John, 1994]. John notices that paths along the wall of a gridworld can be faster to learn as a sub-optimal move has less of a chance of sending the agent further away from the goal. Convergence of the $\bar{Q}$-learning algorithm was proved along with a more general class of MDPs using non-expansions [Littman and Szepesvári, 1996]. In essence, our work explores the same effect but in the setting of learning from critique. When there are multiple solutions to a task, all equally optimal, how does providing feedback along one or more of them affect the learning performance of the agent?

4 Teaching Strategies

We study a carefully designed set of teachers that span a range of evaluation behaviors. We use automated oracles to provide feedback consistent with each behavior, allowing us fine control over the frequency and type of feedback provided to the agent. This is crucial so we can vary other parameters of the MDP and study the effects on learning performance. Our assumption in this work is that the teacher has at least partial knowledge of an optimal policy and will provide feedback consistent with their teaching strategy.

Broadly speaking, we consider four different kinds of feedback strategies, each with different preference of optimal behavior. In certain domains, we also design oracles to follow a specific optimal policy which may be easier or more difficult to learn than others.

1. The *all policy* oracle knows all optimal policies in the domain. In a given state with multiple optimal actions, it will provide positive feedback if the agent chooses any of them. Negative feedback is given upon choosing any non-optimal action. Thus, it encourages the agent to explore states along all optimal policies.
2. The *single policy* oracle has knowledge of only one optimal policy. Thus, in every state, it provides positive feedback on only one optimal action and punishes the rest, whether optimal or not.
3. The *single path* oracle knows one optimal trajectory through the state space. In all the states along this trajectory, it knows one optimal action. In states not on the trajectory, it does not provide any feedback at all. Thus, unlike the *single policy* teacher, it does not know the complete policy, but a path to the goal.
4. The *adaptive* oracle adapts its feedback strategy to the behavior of the agent. In states with multiple optimal actions, it is more likely to positively critique ones that the agent has already tried before. The exact probability of positive critique is described as

$$P(f_{+ve}|s,a_i) = \begin{cases} \exp(n_{s,a_i} - n_{s,a_{\max}}) & a_i \in A_{s,opt} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In state s, $P(f_{+ve}|s,a_i)$ is the probability of positive feedback for executing action a_i, n_{s,a_i} is the number of times action a_i has been executed, $n_{s,a_{\max}}$ is the maximum such count over all optimal actions, and $A_{s,opt}$ is the set of of all optimal actions. This probability starts out as 1 for all the optimal actions in the state and exponentially decreases for all except the one that has been taken by the agent the maximum number of times. Thus, this oracle mimics the *all policy* oracle in the beginning but quickly starts encouraging the agent along a single policy that it has explored the most.

It should be noted that all of the oracles described above provide feedback strictly in line with one or more optimal policies, though they have different preferences over them. Our conjecture is that even with teachers based solely on optimal policies, learning performance can vary greatly over different characteristics of the domain.

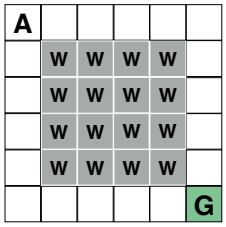
5 Domain Descriptions

As with the teachers, our domains are designed to be simple and easily manipulable. We test on gridworld domains with a single goal state (+10 reward) and stationary 'pit' states (-10 reward), both of which terminate the episode. This simplification allows us to enumerate all optimal policies and build our oracles to measure their effect on performance.

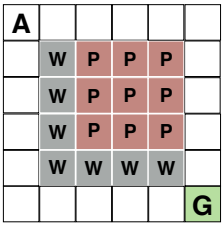
5.1 Significant risk during learning

The *blocks* domain (Figure 1a) is a gridworld with all but two paths to the goal blocked by walls. If the agent tries to move into a grid location occupied by a wall its location remains unchanged. The performance of the agent is measured by the number of steps it takes to reach the goal. Both paths are identical, but during learning the agent may happen to randomly explore one first. Exploration may also be biased by feedback. Here we can test whether adapting feedback to agent's exploratory behavior performs better than non-adaptive feedback along one or both paths.

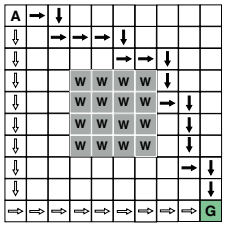
An interesting variation of the *blocks* domain is shown in Figure 1b. Here, one of the paths leading to the goal is lined with pits. The other path, therefore, is clearly safer during exploratory random behavior. Yet in a deterministic setting, they are both optimal. Here, the *safe path* and *risky path* oracles provide feedback only along the safe and risky paths respectively. The *safe policy* and *risky policy* teachers encourage the agent towards their preferred paths, but will still provide correct feedback if the agent happens to take the other one (see Figure 2).



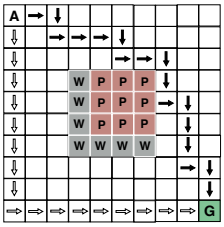
(a) The *blocks* domain has only two optimal policies. Most states contain only one optimal action (except the top left corner).



(b) The *pits* domain also has two optimal policies, but one of them is clearly better for learning with a random exploration strategy.



(c) The *extended blocks* domain has a large number of optimal policies. Most states contain two optimal actions. Shown are two possible optimal paths.



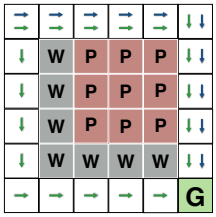
(d) The *extended pits* domain also has a large number of optimal policies. Paths along the walls have a lower chance of moving away from the goal or falling into pits.

Figure 1: The gridworlds used to explore effect of domain characteristics on performance of different feedback strategies. The *pits* domain is used to contrast two optimal paths that differ in ease of learning. In *blocks*, both optimal paths are identical, yet one may be explored more than the other. *Extended pits* and *extended blocks* domains are used to study the effect of a large number of optimal policies.

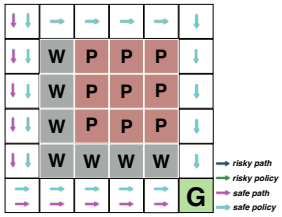
5.2 Large number of optimal policies

In the *extended pits* domain (Figure 1d), multiple optimal pathways exist and different combinations of these paths produce a large number of optimal policies. Despite this large number of possibilities, the symmetry and simplicity of the environment allows us to define all of them. A similar analog exists for the *blocks* domain as *extended blocks* (Figure 1c).

We would like to point out that not all optimal paths are equivalent during exploration in these domains. Paths away from the pits are less dangerous when learning with an exploration strategy. Similarly, paths along the walls are speedier as there is less of a chance of moving away from the goal if an action is taken at random [John, 1994]. In each of our experiments, we have uniformly randomly picked over all possible single optimal paths and policies to create the *single path* and *single policy* oracles. The goal here is to determine how the performance of the oracle feedback strategies scales to cases with very large number of optimal policies.



(a) *risky path* and *risky policy* oracles both encourage exploration on the path along the pits. *risky path* oracle does not provide feedback on the other path.



(b) *safe path* and *safe policy* oracles encourage exploration away from the pits. Again, *safe path* oracle provides no feedback on the path next to the pits

Figure 2: Feedback behaviour of oracles in *pits* domain. Arrows indicate the action for which positive feedback is provided in each state. All other actions in that state are negatively critiqued. Absence of an arrow means that no feedback is provided by that oracle in that state.

5.3 Spawn and action stochasticity

Without any stochasticity in the domain, the agent has no need to learn the optimal behaviour on the entire state space if it already knows an optimal way to solve the task. The *single path* teacher effectively limits the agent's search space by culling out other equally optimal policies. As a result, the agent learns the optimal policy along the path preferred by the *single path* teacher, but does not know how to perform optimally in other parts of the state space.

This conjecture can be tested in two ways, both of which involve adding stochasticity to the domains. The first is to randomly initialize the agent in different locations each episode, forcing it to experience more of the state space. The other option is to add stochasticity in the action execution mechanism. With a fixed probability, the action chosen by the agent is ignored and in its place another action, chosen at random uniformly from all available actions, will be executed. This probability was fixed at 0.1. This should display some detrimental effects on learning from *single path* oracles as they provide no feedback on large parts of the grid.

6 Results and Discussion

We discuss performance of the automated oracles and the variations due to changes in domain.

6.1 Effect of safe and risky paths

In a deterministic setting, both the paths through the *pits* domain are optimal. But any random exploration policy will likely lead the agent into the pits sometimes. We study the effect of providing encouraging feedback along the safe and risky policies and along both.

The results presented here are averaged over a 1000 independent experiments. T-tests are performed to ensure statistical significance of observations below. Figure 3 shows the performance of the oracles compared to the case where no feedback is provided (*silent*). As expected, the *safe path* and *safe policy* oracles converge to an optimal policy in the least

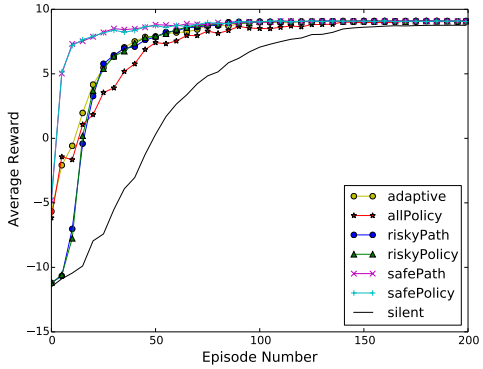


Figure 3: Average rewards obtained by each oracle in the pits domain (shown in Figure 1b).

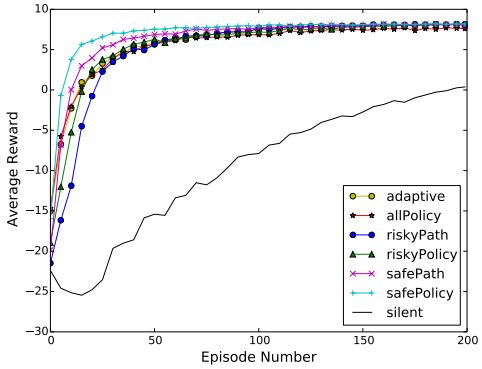


Figure 4: Average rewards obtained by each oracle in the extended pits domain (shown in Figure 1d).

number of episodes. Interestingly, the *risky path* and *risky policy* oracles also converge faster than the *all policy* oracle. A one-sample t-test rejects the null hypothesis that the average reward achieved by *risky policy* oracle is optimal with $p < 0.05$ until episode 109. For *all policy* oracle this occurs at episode 177. As a reminder, the *all policy* oracle provides positive feedback for both optimal policies, while the *risky policy* oracle prefers to guide the agent towards the pits in the initial state (Figure 2a). The *risky policy* oracle is curtailing the agent's exploration of the state space by a large amount. Even though the path is risky during exploration, the agent learns it faster than a policy over the entire state space.

Another interesting observation is that the *safe path* and *safe policy* teachers show almost exactly the same performance in the learning curve (with $p < 0.05$, two-tailed t-test for independence fails to reject null-hypothesis at most episodes). The distinction between the two is that the *safe path* oracle provides feedback only along the path away from the pits and remains silent otherwise, while the *safe policy* oracle provides feedback over the entire state space but encourages the agent towards the safer path in the initial state (Figure 2b). Since there is only one state with more than one optimal action, in practice, the agent learns very quickly to follow the path both the oracles prefer and hence there is not much difference between the two. This will change in the case where more than one state has multiple optimal actions and is discussed in the next section. Finally we note that the *adaptive* oracle provides a good trade-off between *all policy* and the risky oracles, but is obviously not as good as an oracle that knows the safe path.

6.2 Effect of number of optimal policies

Figures 4 and 5 show the results of the oracles on extended versions of the *pits* and *blocks* domain respectively. The extensions are 10x10 gridworlds with 57 states that have two optimal actions (Figure 1c, 1d), creating 2^{57} optimal policies. As compared to the smaller *pits* domain, the *safe policy* oracle performs better than the *safe path* one. This is verified by running a single-tailed t-test on the area under the curve (AUC) formed by each training sample. With $p < 0.05$ the mean AUC formed by *safe policy* oracle is greater than the one by *safe path* oracle. The reason is simply that there are a lot more optimal paths to the goal now and staying on only one is extremely unlikely during the exploration phase in the beginning. Hence, the *safe path* oracle is less likely to be able to provide feedback to the agent at all in the beginning. Therefore, it takes longer to converge. Surprisingly, the *safe path* oracle still surpasses the *all policy* and *adaptive oracles*. This is again due to the *safe path* oracle limiting exploration to the states away from the pits while the *all policy* oracle encourages exploration along both directions.

In the *blocks* domain, both paths to the goal are identical in all respects. The oracles perform almost identically in the small version of this domain. This is perhaps due to its simplicity. On the larger version of this domain, *extended blocks*, the *single path* teacher is the slowest to converge ($p < 0.05$, similar convergence test as above). In the magnified subplot, it can be seen that the *all policy* oracle performs slightly better than the others (Figure 5).

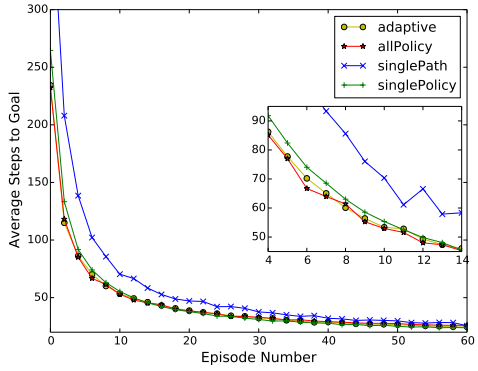


Figure 5: Performance of oracle teachers on the *extended blocks* domain (shown in Figure 1c).

We have seen that oracles able to guide exploration away from risky areas of the domain perform better than those that don't. Surprisingly, oracles that prefer risky areas of the domain still perform better than the oracle encouraging explo-

ration of the entire state space. The key here seems to be in limiting exploration. In domains with no chance of costly death, an oracle providing feedback using the most amount of information on the optimal policies performs the best.

6.3 Effect of spawn and action stochasticities

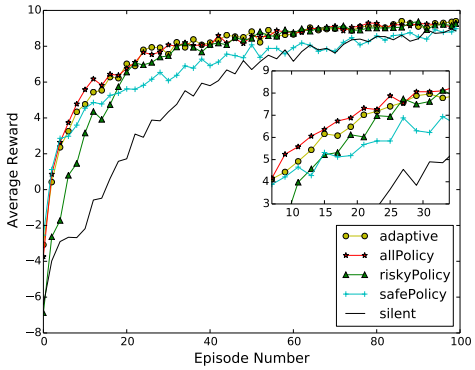


Figure 6: Effect of randomizing agent initial location in *extended pits* domain.

Stochasticity was added to the domains in two different ways. In the first, the agent was spawned in a random location picked uniformly from all empty grid locations at the beginning of each episode. Figure 6 shows the results from the *extended pits* domain with random spawn locations. The *safe path* and *risky path* oracles are not shown for the sake of clarity but their performance closely follows their *policy* oracle counterparts, albeit slightly worse. The first observation we would like to make is that now the *all policy* and *adaptive* oracles performs the best. Their mean AUCs are greater than those of *safe policy* and *risky policy* with p almost equal to 0 for the null hypothesis. The *risky policy* oracle starts out worse than *safe policy* one, but then quickly overtakes it around episode 15. Both of these surprising observations have similar explanations. Since the agent is spawned at a random location each time, it is important for it to know how to behave optimally over the entire state space. The *all policy* oracle encourages exploration along all parts of the domain, allowing the agent to learn such behavior. The *risky policy* oracle encourages the agent to explore the risky parts of the state space. Although this leads to frequent deaths in the beginning, the agent eventually learns to navigate this area better than the *safe policy* and hence avoids the pits when it is spawned there in the future.

Separately, in the *extended blocks* domain, the effect of adding random noise in the action execution was evaluated. The relative performance of the oracles remained similar to that shown in Figure 5, but the *single path* oracle took even longer to converge. Thus, in the stochastic setting, randomizing spawn locations seems to flip the relative learning performance of the oracles. The feedback strategy that encourages exploration of the entire state space now performs best in the long run because it prepares the agent to act optimally wherever it spawns. Adding stochasticity in action selection to the domain does not seem to affect the relative learning performance in the *extended blocks* domain, but the *single path* oracle takes even longer to converge.

7 Conclusions

It is apparent that adoption of different feedback strategies over multiple optimal policies can result in different learning performances for the agent. It is also clear that which feedback strategy works the best is highly dependent on the domain. For example, while we would like to avoid encouraging the agent to explore the risky path in the *pits* domain, add some randomness in spawn location and it becomes important to explore and learn the policy over the risky area. Another example is that feedback provided along only a single policy in the *extended pits* domain surpassed the performance resulting from providing feedback along all optimal policies. This shows how powerful limiting the exploration through critique can be while learning.

In our experiments with oracles in the *pits* domain, we have seen that the safer path accrues less negative reward during learning than the one along the pits. This is because during the initial learning phase, the agent sometimes takes actions at random to learn their effects. The safer path allows the agent to explore without incurring large negative rewards. Hence, if there is no stochasticity in the domain, encouraging the agent to explore only along the safe path will achieve positive rewards faster than trying to teach it both optimal solutions.

It may be the case that the policy which is easiest to learn is not the one the teacher is interested in teaching. A lot of computational and human effort may be wasted in exploring difficult to learn optimal policies before positive results are seen. This can lead to frustrating human-agent interactions. On the other hand, if there is some stochasticity in the domain, it is important that the agent know how to behave along both the paths. Feedback along only the safe path may mean that the risky path remains under-explored and delay convergence to the optimal policy in all states. For this reason, we believe considering the effect of feedback strategy employed by the teacher over the multiple optimal solutions existing in a domain can benefit interactive machine learning algorithms and interfaces.

8 Future Work

In complicated real world settings, numerous factors influence the decision to provide feedback along one or more optimal policies. We have just scratched the surface of the problem by investigating this phenomena in simple, easily manipulable domains. If we are to build machines that work with and learn from human collaborators, it is important to understand how humans tend to give feedback in such situations and how it impacts the learning performance of the agent.

It is not entirely clear how our conclusions directly extend to continuous and high dimensional domains due to technical factors that would need to be accounted for, including the representation used for the state-action space and generalization ability of the learning algorithm. This can be a fruitful direction for future work. With this work, we hope to show that this phenomena deserves further study.

References

[Cederborg *et al.*, 2015] Thomas Cederborg, Ishaan Grover, Charles L. Isbell, and Andrea L Thomaz. Policy Shaping With Human Teachers. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[Chen *et al.*, 2013] Tiffany L. Chen, Matei Ciocarlie, Steve Cousins, Phillip M. Grice, Kelsey Hawkins, Kaijen Hsiao, Charles C. Kemp, Chih-Hung King, Daniel A. Lazewatsky, Adam E. Leeper, Hai Nguyen, Andreas Paepcke, Caroline Pantofaru, William D. Smart, and Leila Takayama. Robots for humanity: Using assistive robotics to empower people with disabilities. *IEEE Robotics and Automation Magazine*, 20(1):30–39, March 2013.

[Chernova and Thomaz, 2014] Sonia Chernova and Andrea L. Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

[Chi and Lieberman, 2011] Pei-Yu Chi and Henry Lieberman. Intelligent assistance for conversational storytelling using story patterns. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 217–226, New York, NY, USA, 2011. ACM.

[Dearden *et al.*, 1998] Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian q-learning. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 761–768, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[Griffith *et al.*, 2013] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems 26*, pages 2625–2633. Curran Associates, Inc., 2013.

[Isbell *et al.*, 2001] Charles Isbell, Christian R. Shelton, Michael Kearns, Satinder Singh, and Peter Stone. A social reinforcement learning agent. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 377–384, New York, NY, USA, 2001. ACM.

[John, 1994] George H John. When the best move isn't optimal: Q-learning with exploration. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, page 1464, 1994.

[Knox and Stone, 2008] W.B. Knox and P. Stone. Tamer: Training an agent manually via evaluative reinforcement. In *7th IEEE International Conference on Development and Learning (ICDL 2008)*, pages 292–297, Aug 2008.

[Knox and Stone, 2010] W. Bradley Knox and Peter Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '10, pages 5–12, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[Lazewatsky and Smart, 2014] Daniel A. Lazewatsky and William D. Smart. Accessible interfaces for robot assistants. In *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man)*, pages 106–111, Edinburgh, Scotland, 2014.

[Littman and Szepesvári, 1996] Michael L Littman and Csaba Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *13th International Conference on Machine Learning (ICML '96)*, pages 310–318, 1996.

[Loftin *et al.*, 2014] Robert Loftin, James MacGlashan, Bei Peng, Matthew Taylor, Michael Littman, Jeff Huang, and David Roberts. A strategy-aware technique for learning behaviors from discrete human feedback. In *AAAI Conference on Artificial Intelligence*, 2014.

[Lu and Smart, 2011] D.V. Lu and W.D. Smart. Human-robot interactions as theatre. In *Proceedings of the 20th IEEE International Symposium on Robot andHuman Interactive Communication (Ro-Man)*, pages 473–478, July 2011.

[Ng *et al.*, 1999] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[Sahni *et al.*, 2016] Himanshu Sahni, Brent Harrison, Kaushik Subramanian, Thomas Cederborg, Charles Isbell, and Andrea Thomaz. Policy shaping in domains with multiple optimal policies. In *Proceedings of Autonomous Agents and Multiagent Systems (AAMAS)*, 2016.

[Thomaz and Breazeal, 2008] Andrea L. Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716 – 737, 2008.

[Watkins, 1989] Christopher J Watkins. *Models of delayed reinforcement learning*. PhD thesis, Cambridge University, 1989.

[Zhu, 2015] Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI Conference on Artificial Intelligence (Senior Member Track)*, 2015.