Information Transfer of Actions in Unexplored States using Policy Shaping

Ishita Sarraf
sarrafis@grinnell.edu
Grinnell College
Grinnell, Iowa, USA

Elaine Schaertl Short
elaine.short@tufts.edu
Tufts University
Medford, Massachusetts

Abstract

Previous work in supervisory attention driven exploration restricts the robot to only exploit those actions that a human has given positive feedback to when the human is not paying attention. We extend that work by proposing two algorithms - Best Action and Similarity so that the robot can identify actions that are similar to those that a human gave positive feedback to in explored state spaces, so that the robot can choose safe actions even in unexplored state spaces without human attention. Therefore, significantly improving the robot's learning efficiency.

Keywords

Human Robot Interaction, Policy Shaping, Human Attention, Robot Learning

ACM Reference Format:
Ishita Sarraf and Elaine Schaertl Short. 2024. Information Transfer of Actions in Unexplored States using Policy Shaping. In . ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Reinforcement Learning (RL) is a technique that allows a robot to learn any task through trial and error methods [2]. There are many existing methods that make the robot learn tasks faster and more optimally. It has been proven that using humans in this learning technique makes the robot learn faster - a technique that is called policy shaping [3]. The robot is allowed to explore and exploit its environment using policy shaping. However, a drawback to policy shaping is that it needs the human to be present continuously and give feedback which is not possible in the real world. Faulkner et al. [5] identified this problem and proposed Attention-Modified Policy Shaping (AMPS).

AMPS, however, restricts the robot to exploit actions that a human has given positive feedback to when the human is not paying attention. We extend that work by testing two different algorithms - Best Action and Similarity on AMPS to remove the limitation. Best Action selects the best probable action in an unexplored state space that a robot can take based on previous feedback on explored state spaces. Similarity uses a similarity metric to identify explored state spaces using the features defined for each state space and then picking the best action. Identifying the best action from explored state spaces will allow the robot to choose safe actions even in unexplored state spaces without human attention. Therefore, the robot learns faster and earns greater rewards.

2 Background

RL is a machine learning technique used to teach an agent to perform a task by providing positive and negative feedback, meaning that they do not need to have any prior knowledge of the task. [7]. Interactive RL builds on RL by adding humans whose feedback can be integrated into the original RL algorithm using policy shaping which directly influences the policy that the robot uses to learn its task [6].

Both proposed algorithms are developed using policy shaping and the AMPS algorithm. As previously stated, policy shaping is a technique that incorporates human feedback directly in its reinforcement learning algorithm [3]. Faulkner et al. [5] use Policy Shaping along with results from human-robot integration and curiosity-driven learning to develop the AMPS algorithm. Prior work in Human-Robot Interaction (HRI) does consider human attention to modify the robot's behavior but it does not directly change the learning style of the robot based on the human's attention [9, 13]. In [11], the authors develop a technique that allows the robot to adapt its behavior based on its perception of the human's attention.

Curiosity-driven learning allows the robot to explore its environment based on maximizing learning and information-potential, not just maximizing rewards [1, 4, 10] which means the robot gives more emphasis on exploring unknown states and gathering information, than merely exploiting known states for higher rewards. Previous work in curiosity-driven learning used combined curiosity-driven learning with human teachers by allowing the robot to choose between exploring and following human advice, but it too assumes that the human is paying attention the entire time [10].

Researchers [8] have also modeled a human's feedback that reduces its need for attention the more it becomes confident of its model of the teacher. This allows the humans to take more breaks from teaching the robot. However, our work not only allows the humans to take breaks, but also allows the robot to learn faster by identifying actions that it can safely perform in new, unexplored states without human attention.

3 Algorithm

In this section, we explain the different baseline algorithms used to create Best Action and Similarity - Reinforcement Learning, Policy Shaping, Attention Modified Policy Shaping. The two proposed

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

algorithms, Best Action and Similarity, can identify best actions in unexplored state spaces. Best Action helps the robot to identify the overall best action, based on prior human feedback, to take when it is in a new state. Similarity helps the robot identify states similar to the new state and then pick the best overall action using the feedback given in the similar states.

3.1 Reinforcement Learning

The RL algorithm is constructed as an MDP and uses Q-Learning that learns Q-values for each state-action pair to solve the MDP. An MDP is defined by (S, A, T, R, γ) where S is a set of states, A is a set of actions, T is a transition probability function $S \times A \longrightarrow Pr[S]$, R is a reward function $S \times A \longrightarrow \mathbf{R}$, and γ is a discount factor where $0 \leq \gamma \leq 1$. RL methods select a policy $\pi : S \times A \longrightarrow \mathbf{R}$ that can get the maximum possible reward in the environment. Q values $Q(s, a)$ are used to calculate future expected reward for action $a \epsilon A$ and state $s \epsilon S$. The Q-Learning algorithm uses Boltzmann exploration [12], for which the probability of selecting each action is

$$Pr_q(a) = \frac{e^{Q(s,a)/\tau}}{\Sigma a' e^{Q(s,a')/\tau}} \tag{1}$$

where τ is the exploration constant set to 0.5 which decreases by 1% each learning episode. The Q-Learning parameters α (learning rate) is set to 0.1 and γ (discount factor) is set to 0.9 to maximize the performance of policy shaping.

3.2 Policy Shaping

We add Policy Shaping with Q Learning to incorporate human feedback. Policy Shaping takes positive or negative binary feedback and changes the current policy based on the feedback. To account for inconsistent feedback from the human teachers, there is a parameter C that gives the probability that the human teacher is correct. Here, we set C to 0.9 to indicate that the teacher will be correct 90% of the time. We chose this value for C to match with the AMPS algorithm [5]. We estimate that the probability that any action in a given state is good by the different between the positive and negative human feedback for that action. Below is the equation used for this probability:

$$Pr_c(a) = \frac{C^{\delta_{s,a}}}{C^{\delta_{s,a}} + (1 - C)^{\delta_{s,a}}} \tag{2}$$

where $\delta_{s,a}$ is the difference between positive and negative feedback received for any given state s and action a [6].

The final probability of taking any action in a given state as used in [3] is

$$Pr(a) = \frac{Pr_q(a) Pr_c(a)}{\Sigma_{\alpha \epsilon A} Pr_q(\alpha) Pr_c(\alpha)} \tag{3}$$

3.3 Attention-Modified Policy Shaping

We implement the AMPS algorithm that keeps track of the state-action pairs the teacher has seen A_{seen} and state-action pairs the teacher gave positive feedback for A_{good}. When the human is paying attention, there is 0.5 probability that the robot will choose to explore new states or exploit good states. If the lists, A_{seen} or A_{good}, are empty, then the robot reverts to the policy shaping algorithm. The periods of attention and inattention are predetermined into the algorithm. However we did not implement the part— when there is no human attention, the robot only chooses from the A_{good} list and instead came up with two different algorithms for the robot to identify good actions in unexplored state spaces.

3.4 Best Action

In the Best Action algorithm, the robot chooses the action which has received the most positive feedback across all states $f(s, a)$, when it is in an unexplored state as shown in Algorithm 1 when the human is not paying attention. It goes through feedback for each action for all the states and checks which action got positive feedback across all states. If that action is possible to be taken in that unexplored state space, then it increases the original probability P_a, as calculated by the final policy shaping equation (3), of taking that action by 50%.

Algorithm 1 Best Action Algorithm

```
for $f(s, a)$ in possible actions do
    if $f(s, a) > 0$ then
        return $P_a * 0.5$
    end if
end for
```

3.5 Similarity

The Similarity algorithm allows the robot to find similar states using the predefined features of the states, a string of binary digits, as shown in Algorithm 2 when the human is not paying attention. It finds states similar to the new state using the state features and stores it as *similarity* list. Then it goes through the feedback for all those states in *similarity* and checks which action got positive feedback. If that action is possible to be taken in that unexplored state space, then it increases the original probability P_a, as calculated by the final policy shaping equation (3), of taking that action by 50%.

Algorithm 2 Similarity Algorithm

```
for $feature$ in state features do
    if $feature$ similar to $new_{statefeature}$ then
        if $f(s, a) > 0$ then
            return $P_a * 0.5$
        end if
    end if
end for
```

4 Simulation Experiment

We compare both algorithms with the prior approach in AMPS for a simulated cup placement test for 7x7 grid. The robot's goal is to push the cup to the desired location in the most optimal, i.e. shortest, way possible.

4.1 Experimental Design

The goal location of the cup is (6,6) with the grid indexed from 0. The problem is formulated as an MDP with S = (x,y) which are the coordinates of the grid, A = north, south, east, west as the actions that robot can push the cup in. For the transition function, T, each action moves the cup one grid square in the desired direction. The reward is +10 for reaching the desired location and -10 for not. All other states have a negative reward of -1 to make the robot reach the desired location faster. The maximum reward possible is -1 which is achieved by pushing the cup to the goal in 12 steps.

To represent the human teacher, we used an oracle that gives positive feedback of +1 when the robot moves towards the cup, and -1 when it does the opposite.

4.2 Experiment

The robot starts from (0,0) and learns to go to the desired location using all 3 algorithms - AMPS, Best Action, and Similarity. The learning period is over 50 episodes with 40 steps for each episode and is run 15 times. The robot is given attention in two batches - for the first five episodes and then for episodes 20-25 for each run.

4.3 Results

Figure 1 shows the learning curves of all 3 algorithms. The shaded sections indicates that the robot was given attention for those episodes. AMPS performs comparably during and after the first round of attention but after the second round of attention, Best Action and Similarity both greatly outperform AMPS. The average area under the reward curve for Best Action (Mean (M) = 10.62133, Standard Deviation (SD) = 8.986989) has a difference of about 9 with the average area under the reward curve for AMPS (Mean (M) = 1.584, Standard Deviation (SD) = 1.498756), $p - value = 0 < 0.05$ (using One-way ANOVA, followed by Tukey HSD test). The average area under the reward curve for Similarity (Mean (M) = 8.04, Standard Deviation (SD) = 7.505112) has a difference of about 6 with the average area under the reward curve for AMPS, $p - value = 0 < 0.05$ (using Tukey HSD test). The average area under the reward curve for Best Action has a difference of about 3 with the average area under the reward curve than Similarity. We do not have a significant $p - value$ for that test so we cannot say that Best Action and Similarity have any significant mean difference.

These results suggest that both Best Action and Similarity are outperforming AMPS during periods of inattention and allowing the robot to identify similar states safely and learn faster.

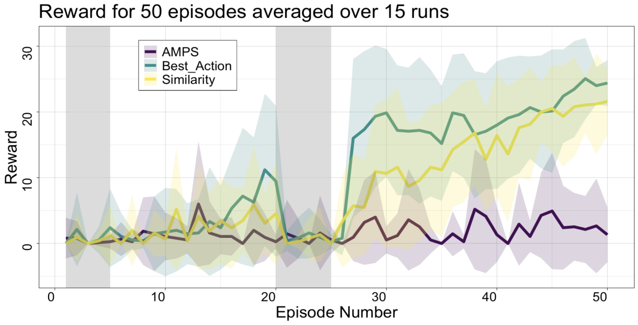


Figure 1: Total rewards during learning for 50 episodes. All rewards are averaged over 15 runs. The shaded background indicates attention.

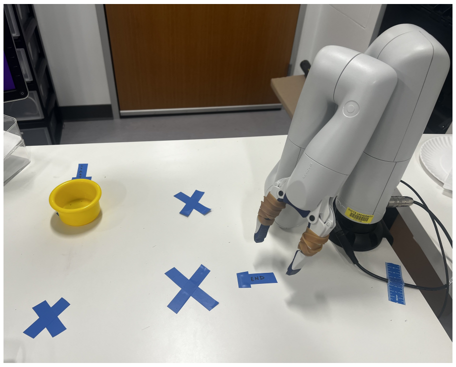


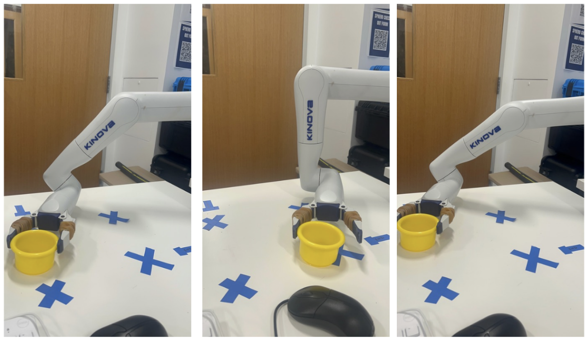
Figure 2: Set Up for the Experiment with the Kinova Gen3 arm

5 Real-World Experiment

We test the experiment on a robot to see it learn and push an actual cup to the desired location in a 7x7 grid to verify real-world performance.

5.1 Robot Setup

We used a robot, with a KINOVA Gen3 arm with 6 degrees of freedom shown in Figure 2. The robot was initialized to a home position and made to move towards the cup that was always kept in a predetermined position and then it would grip the cup. To push the cup, the gripper held the cup by its sides and moved it to a predetermined distance forward, backward, left, or right. The setup was the same for each experiment. When the robot reached the end state in a episode, it would release the cup. If it did not reach the end state in that episode, it would go to the next episode with the cup still in its hand. At the end of all episodes, it would go back to its home position. If there was any malfunction such as the robot did not grip the cup properly or turned in a position that made

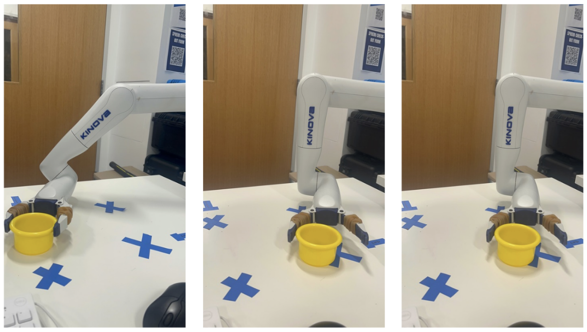


Episode 7 (40 steps, does not reach the desired final state)

Episode 22 (28 steps, reaches the desired final state)

Episode 47 (40 steps, does not reach the desired final state)

Figure 3: Results for AMPS experiment showing the final position of the robot at the end of 3 different stages



Episode 7 (40 steps, does not reach the desired final state)

Episode 22 (28 steps, reaches the desired final state)

Episode 47 (12 steps, reaches the desired final state)

Figure 4: Results for Best Action experiment showing the final position of the robot at the end of 3 different stages

it stuck, the robot had to be shut down and the experiment was restarted.

5.2 Experiment Results

The robot was run for each experiment using the oracle as human feedback.

AMPS algorithm:

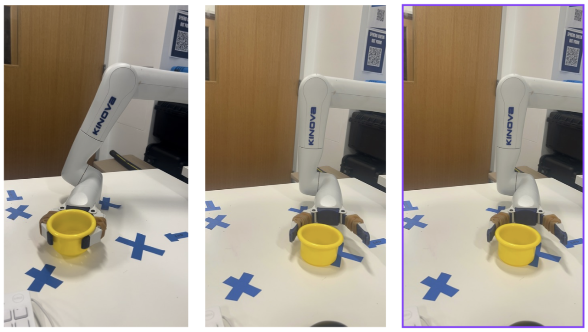
Figure 3 shows the performance of the robot at the end of episodes 7, 22, and 47. These episodes show the robot's performance after the first round of attention, during the second round of attention, and after the second round of attention respectively. We can see that the last position of the robot at the end of the respective episodes. The robot was only able to reach the desired goal at the end of episode 22 when it had human attention but not in episode 7 or episode 47. It shows that the robot did not learn the task by the end of the experiment.

Best Action algorithm:

Figure 4 shows the performance of the robot at the end of episodes 7, 22, and 47. These episodes show the robot's performance after the first round of attention, during the second round of attention, and after the second round of attention respectively. We can see that the last position of the robot at the end of the respective episodes. The robot was able to reach the desired goal at the end of both episode 22 and episode 47 but not episode 7. It shows that the robot learned the task by the end of the experiment.

Similarity algorithm:

Figure 5 shows the performance of the robot at the end of episodes 7, 22, and 47. These episodes show the robot's performance after the first round of attention, during the second round of attention, and after the second round of attention respectively. We can see that the last position of the robot at the end of the respective episodes. The robot was able to reach the desired goal at the end of both episode 22 and episode 47 but not episode 7. It again shows that the robot learned the task by the end of the experiment.



Episode 7 (40 steps, does not reach the desired final state)

Episode 22 (28 steps, reaches the desired final state)

Episode 47 (12 steps, reaches the desired final state)

Figure 5: Results for Similarity experiment showing the final position of the robot at the end of 3 different stages

We observe that after 47 episodes, the robot was not able to learn the task using AMPS, but was able to learn the task using both Best Action and Similarity.

6 Discussion

The results show that the average areas under both the Best Action and Similarity algorithm curves are consistently higher than the AMPS algorithm, especially after the second round of attention. Therefore, these results suggest that the robot performs better without attention with the proposed algorithms than the AMPS algorithm.

However, there remains additional testing needed to verify the performance of these algorithms - starting with using human participants to test the algorithms. We expect that the algorithms will outperform the AMPS algorithm with human participants because of the results from the simulation experiments but it needs to be tested. Additionally, the algorithms were only tested on one task. There should be various tasks tested and the scope of the grid should be increased as well. Moreover, the experiments were run only 15 times, there should also be more runs to have a better average reward curve. The proposed algorithms use a greedy approach when it comes to selecting the action but should be modified to choose the action with the highest positive feedback. Furthermore, the state features are defined by a string of binary digits, changing that to the different positions of the robot in different states will make it easier to do more complicated tasks.

7 Conclusion

The study shows that using the proposed algorithms the robot can safely identify actions similar to the ones it received positive feedback for our task. This study introduces how humans can multitask while teaching robots - thus, giving them breaks and allowing the robot to learn faster because it can quickly identify similar states. The results from the two proposed algorithms show that the robot improves its learning efficiency by identifying similar actions for new states and also outperforms AMPS. These results suggest that robots could possibly learn to lightly explore new states without human attention in order to learn faster, for certain tasks.

References

[1] Joshua Achiam and Shankar Sastry. 2017. Surprise-based intrinsic motivation for deep reinforcement learning. (2017). https://arxiv.org/abs/1703.01732 arXiv: 1703.01732 [cs.LG].

[2] Andrew G. Barto. 1997. Chapter 2 - reinforcement learning. In *Neural Systems for Control*. Omid Omidvar and David L. Elliott, (Eds.) Academic Press, San Diego, 7–30. ISBN: 978-0-12-526430-3. DOI: https://doi.org/10.1016/B978-012526430-3/50003-9.

[3] Thomas Cederborg, Ishaan Grover, Charles L. Isbell, and Andrea L. Thomaz. 2015. Policy shaping with human teachers. In *Proceedings of the 24th International Conference on Artificial Intelligence* (IJCAI'15). AAAI Press, Buenos Aires, Argentina, 3366–3372. ISBN: 9781577357384.

[4] Nuttapong Chentanez, Andrew Barto, and Satinder Singh. 2004. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*. L. Saul, Y. Weiss, and L. Bottou, (Eds.) Vol. 17. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2004/file/4be5a36cbaca8ab9d2066debfe4e65c1-Paper.pdf.

[5] Taylor Kessler Faulkner, Elaine Schaertl Short, and Andrea Lockerd Thomaz. 2018. Policy shaping with supervisory attention driven exploration. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 842–847. DOI: 10.1109/IROS.2018.8594312.

[6] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. 2013. Policy shaping: integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*. C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, (Eds.) Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2013/file/e034fb6b66aacc1d48f445ddfb08da98-Paper.pdf.

[7] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: a survey. *Journal of artificial intelligence research*, 4, 237–285.

[8] Taylor Kessler Faulkner, Reymundo A. Gutierrez, Elaine Schaertl Short, Guy Hoffman, and Andrea L. Thomaz. 2019. Active attention-modified policy shaping: socially interactive agents track. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (AAMAS '19). International Foundation for Autonomous Agents and Multiagent Systems, Montreal QC, Canada, 728–736. ISBN: 9781450363099.

[9] Marek Piotr Michalowski, S. Sabanovic, and Reid Simmons. 2006. A spatial model of engagement for a social robot. (Mar. 2006).

[10] Sao Mai Nguyen and Pierre-Yves Oudeyer. 2013. Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner. *Paladyn Journal of Behavioural Robotics*, 3, (May 2013), 136–146. DOI: 10.2478/s13230-013-0110-z.

[11] Pramila Rani and N. Sarkar. 2005. Operator engagement detection and robot behavior adaptation in human-robot interaction. In vol. 2005. (May 2005), 2051–2056. DOI: 10.1109/ROBOT.2005.1570415.

[12] Chris Watkins. 1989. *Models of delayed reinforcement learning*. Ph.D. Dissertation. Cambridge University.

[13] Qianli Xu, Liyuan Li, and Gang Wang. 2013. Designing engagement-aware agents for multiparty conversations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). Association for Computing Machinery, Paris, France, 2233–2242. ISBN: 9781450318990. DOI: 10.1145/2470654.2481308.