DemPref vs. T-REX

Subashri Ramesh

December 22, 2022

1 Introduction

In this project I compared the DemPref and T-REX models as described by Palan et al.[PLSS19] and Brown et al.[BGNN19] respectively. Both of these algorithms are based around improving off of demonstrations provided to the algorithm. Video games are difficult by nature, and that is why we find them interesting. If we were able to provide optimal demonstrations immediately, then we would get bored very quickly. Therefore, video games are based around improving off of our past performance, similar to these algorithms. Through the comparison of these algorithms, we show that both objectively and subjectively, DemPref is more successful at the lunar lander task. It outperforms T-REX while requiring lower effort to train.

2 Background

This project is comparison between the T-REX model proposed by Brown et al.[BGNN19] and DemPref proposed by Palan et al.[PLSS19]. Both of these algorithms have a goal of improving off of suboptimal demonstrations, but they have different methods to accomplish this. This can lead to very different experiences for the user and subsequently, different performance.

The T-REX model was successfully implemented by Brown et al.[BGNN19] and tested on high-dimensional Atari video game tasks. Their model is specifically created to extrapolate on demonstrations and outperform them. From ranked demonstrations, they produce a neural network classifier which determines preferences between trajectories and represents the learned reward function. This learned reward function is used to find the best policy through reinforcement learning. The performance of their model within the Atari environment significantly improved upon the demonstrations provided by the users. It was also shown that T-REX is fairly robust to noise in the labelling of preferences between trajectories as well as unlabeled time-based data.

The DemPref model was created by Palan et al.[PLSS19] to solve the weaknesses of inverse reinforcement learning and preference learning through the use of a new combined framework. Demonstrations for inverse reinforcement learning contain high amounts of information but can have low accuracy. It is very difficult to demonstrate every possible situation and demonstrations are not necessarily perfect. Preference learning tends to be inefficient since it uses binary feedback to learn a high dimensional problem, but its information is very accurate. In this paper, they propose DemPref which combines inverse reinforcement learning and preference learning to learn a reward function. The framework uses the demonstrations to generate better queries for preference learning. They vary the demonstration quality between high and low quality for testing and they find that the algorithm generates better trajectories which are different from the original demonstration.

These two algorithms work very differently, T-REX extrapolates off of demonstrations which requires some training to generate optimal trajectories. On the other hand, DemPref is an active reward learning algorithm which requires the participant to actively provide preferences between trajectories so it is a more involved experience for users. They are similar in that they analyze preferred algorithms against each other because T-REX works with ranked preferences between trajectories while DemPref works with pairwise trajectories. T-REX seems to be able to extrapolate even from expert demonstration since it tries to learn an approximation of the reward function. However, DemPref was not able to extrapolate as well from expert demonstration but was instead only capable of imitating expert demonstrations. In this project I will investigate the effects of these differences.

2.1 Related Works

Laidlaw et al.[LR21] studied human preferences in uncertain situations because humans do not always know the optimal decision but usually have a preference. In their paper, they show that it is easier to identify human preferences when there is more uncertainty, and this has implications in improving preference learning methods with suboptimal people. People who make decisions under uncertainty provide more information than decisions with certainty because in certainty there is one right answer, but uncertainty shows more about people's preferences. They analyze inverse decision theory to discover how many decisions are needed to learn the decision makers' cost function and the effects of a suboptimal decision maker. From their mathematical results, they concluded, that realistic suboptimal decisions allow for rich information such as which data to ignore and which evidence to over/under weight. They define unrealistic suboptimal decision making as always choosing incorrectly, or purely randomly. Suboptimal preferences may lead to improved performance of models because they contain more information.

We can never really call a human demonstration an optimal demonstration unless that is exactly what we want the robot to do. Basu et al.[BYH+17] showed that people preferred that autonomous vehicles drove more carefully and at lower speeds instead of aggressively like them. In their paper, they conducted a user study to test people's preferences in driving style for autonomous vehicles. Driving is a realm where humans are suboptimal examples but we expect autonomous cars to learn optimal policies. From their study, they also found that the majority of participants preferred that the car drove as they believed that they themselves drove, but they inaccurately categorized their own driving. This paper is relevant to the suboptimality problem because it shows the necessity for machines to extrapolate on our suboptimal behavior. For machines to play a larger role in our lives, we need models to build off of our suboptimal behavior and behave optimally.

3 Approach

T-REX uses ranked demonstrations to learn a reward function from which we can create an optimal policy. First, demonstrations will be collected for the task. Then the demonstrations will be ranked by their performance. The trajectories from the demonstrations will be split into smaller segments while preserving the preferenec labels to produce more training data. Then, the reward learning network will be trained on this data to produce a reward function. Using this reward function, I will produce a policy which optimizes this function. The original T-REX model is trained on images of the game so that relevant features need to be extracted by the reward learning neural network. In order to make the algorithms more comparable, I modified the reward learning network to instead use only one layer with no bias which will take the set of features defined by DemPref and produce an expected reward. These features include relevant features such as orientation and distance to target.

DemPref uses demonstrations to initialize a reward distribution and then generate queries which provide maximum information in order to optimally update the reward distribution. So first, demonstrations will again be collected. Then, the reward distribution will be initialized from the demonstrations. Then the model will query the user with pairs of trajectories for which the user will provide a preference. Then the distribution is updated with this preference.

I would like the relative workload of each of these algorithms to be equivalent. Additionally, the amount of information provided by the user should be similar between both algorithms to ensure that the difference comes from the type of information instead of the amount. I will have to tune the number of demonstrations and preference queries accordingly.

I chose 2 demonstrations and 25 preference queries which is equivalent to what was used by the Palan et al. in their original work. For T-REX I chose 8 demonstrations because it seemed to take a similar amount of time to produce and rank 8 demonstrations.

3.1 User Study

The users will play the Lunar Lander game provided within the OpenAI Gym environment. They will record their demonstrations through this UI with keyboard inputs. For T-REX, they will be able to play back each of the demonstrations and provide a ranking between them when they are ready. For DemPref they will watch trajectories and be able to play them back and then be queried by the UI to choose the optimal one. Additionally, the users will be randomly assigned one of the algorithms to try first so that there is no ordering bias.

First, the participant was given an overview of the rules and controls for the lunar lander game. They were given 6 trial runs to become familiar with the controls and the response of the game. Then each participant encountered 2 different study conditions, one for each algorithm. In the case that they were assigned the T-REX model first, they would provide 8 demonstrations. Then, they were allowed to rewatch their trials and produce a ranking of the demonstrations. After this, they were asked to complete the NASA TLX survey for the task. Then, they were asked to provide 2 more demonstrations and respond to the queries produced by the model. They would then complete another NASA TLX survey for the second task. In general, each trial took around 20 minutes.

4 Evaluation

I hypothesize that this comparison will show that T-REX is a more optimal algorithm because it seems that it is more robust to noise. Additionally, preference learning algorithms assume that preferences are provided by an expert. It will be interesting to see what happens when the preferences themselves are not optimal, but from Laidlaw et al.[LR21] they describe those uncertain decisions can be more informative of the user's preferences in which case they will improve the learning process if the suboptimal preferences are "realistic." If the objective metrics show that the overall performance of the T-REX algorithm is better than the DemPref algorithm in relation to the quality of the original demonstrations provided, then I can conclude that T-REX is better at learning.

I also hypothesize that DemPref will be rated to take less effort since the queries are a lot simpler, even though this leads to lower amounts of information. I will use the subjective results from DemPref to see if there is a significant difference in the different subjective metrics such as effort. This information will provide insight into how we should query subjects for information since they seem to be more informative in improving on demonstrations. Alternatively, if there is no significant difference then this information is also valuable so that we can conclude that even though the subject's interaction with the video game is different between the algorithms, the results are very similar.

5 Results

The Lunar Lander game has a pretty small action space so instead of Q-learning, I used a random sampler to produce policies. I randomly generated a large set of controls and used the learned reward functions to evaluate the expected reward of the trajectory which results from those controls. Then, I chose the policies with highest expected reward as the optimal policies.

The subjective metrics from the NASA TLX survey were quite interesting. The survey asks the subject to rate mental demand, physical demand, temporal demand, performance, effort and frustration on a 21 point scale. I also added questions on handedness and familiarity with video games. For each of these metrics, I evaluated the p-value to test if the differences between the models were significant ($\alpha = 0.01$).

(Figures 1 - 4 are part of the results section)

6 Discussion

The lunar lander task was harder than anticipated, especially because the lander needs to stand still for some time at the end which was difficult for the model to learn. There was only one successful landing by a participant in the study. Neither T-REX or DemPref performed significantly well on this task with the data provided. I believe that T-REX had an especially difficult time because the real rewards were generally highly clustered. If the lander crashes, it receives a huge instantaneous negative reward, and if it lands successfully, it receives an equally huge positive reward. It is difficult to evaluate if T-REX correctly learned the reward function on a graph such as in Figure 1 due to this clustering. The blue points are the demonstrations which the model was trained on, we see that in Figure 1 there is one successful landing of the lunar lander. The reward is only positive when the landing is successful.

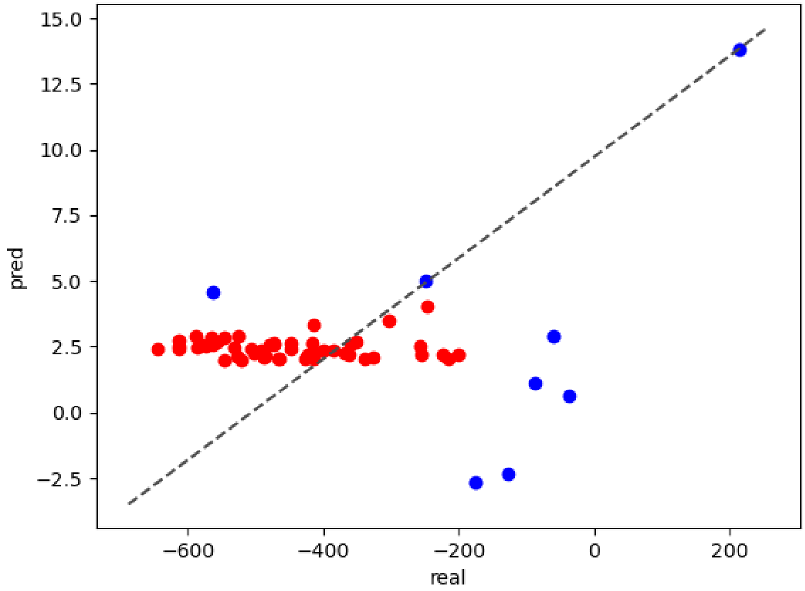

Figure 1: The blue points are the demonstrations and the red points are trajectories produced by the reward function.

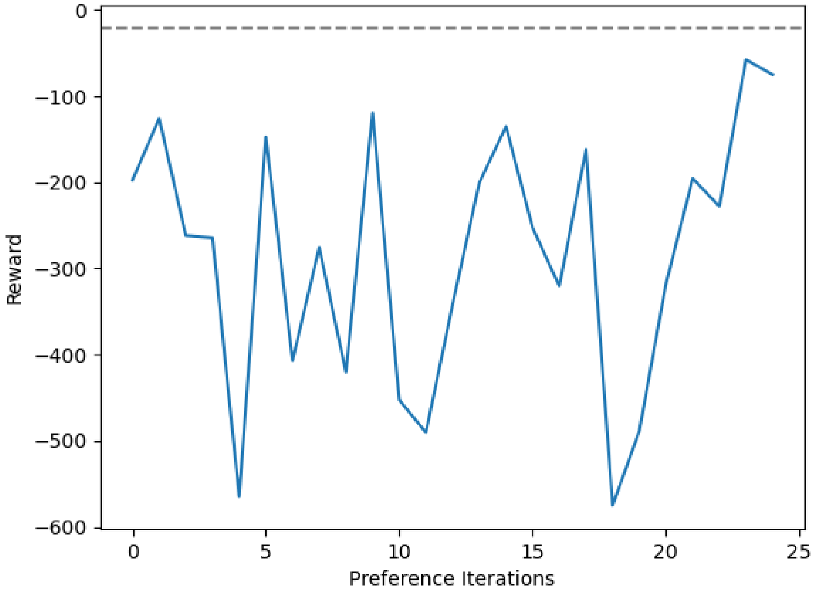

Figure 2: The grey line represents the reward from the best demonstration and the blue line is the reward at each epoch.

The DemPref algorithm was able to find slightly better trajectories because the ending reward value, and overall reward, of the trained model was better than T-REX. The DemPref provided trajectories to which participants indicated their preference. This leads to a highly fluctuating reward function as seen in Figure 2 because even though the model is learning, it may have to fail before it succeeds. These results come to the conclusion that objectively DemPref is a superior algorithm

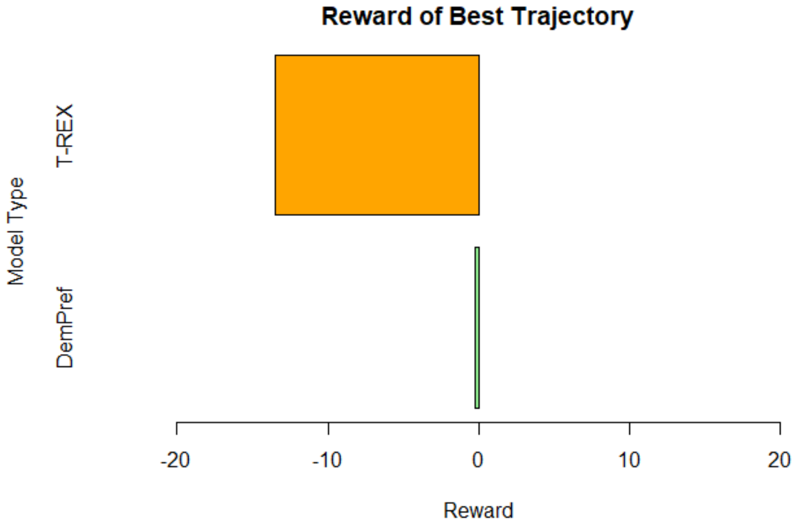

Figure 3: Both models were unable to succesfully complete the task, but DemPref was able to get closer.

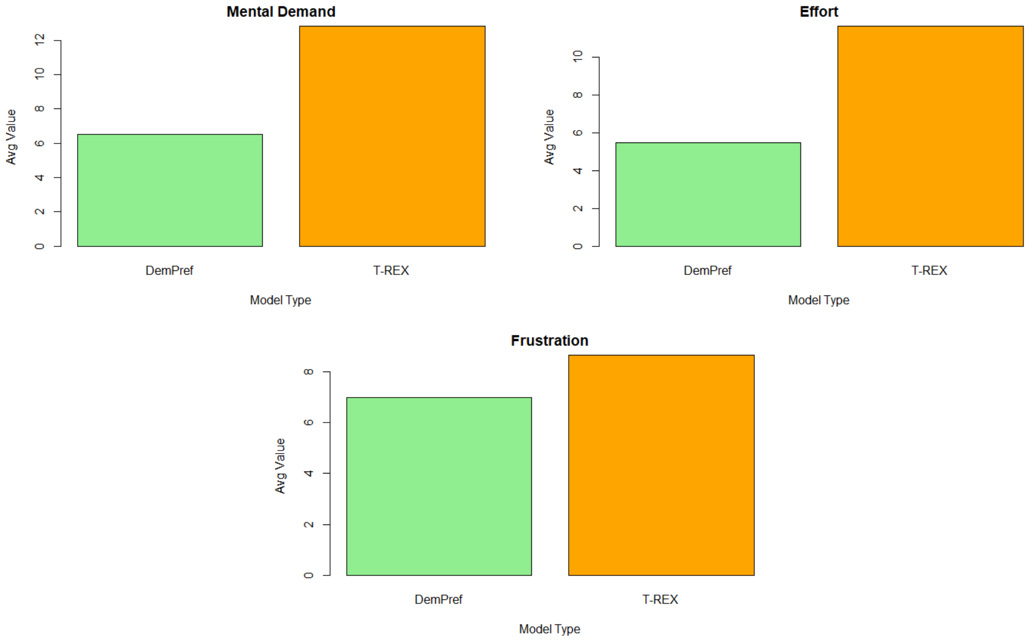

Figure 4: Mental Demand and Effort are the only metrics which had significant differences between T-REX and DemPref.

because it outperformed T-REX. This may however be due to lack of data or lack of exploration in the middle of the reward range. This result is opposite of my hypothesis that T-REX would be the optimal algorithm.

The most interesting part of the results came from the survey feedback. I attempted to make sure that the training of both models would take a similar amount of effort by choosing the amount of demonstrations and queries such that they take similar amounts of time. The result, however, did not reflect this. For all of the metrics within the NASA TLX survey, the only ones with significant results between groups were Mental Demand and Effort. The participants noted that DemPref was a significantly easier algorithm to train in terms of mental demand and effort. This is probably because they were not required to provide more demonstrations but rather provide simpler binary answers.

The other metric I noted is frustration because even though there was no significant difference between groups, I think this result itself is significant. There were multiple comments by participants about feeling frustrated that the agent did not seem to learn the right things when training DemPref. Similar comments were not made for T-REX, but I guess that frustration is equal to the frustration of failing the lunar lander task for more iterations. There were also comments from the subjects that they were not sure what constituted a better trajectory because sometimes both options were really bad. These comments were both for DemPref and for their rankings in T-REX. These results are consistent with my hypothesis that DemPref would be easier to train subjectively.

7 Future Work

I think that there may be some differences in performance of these two algorithms when placed within different game environments. From Ibarz et al.[ILP+18] we see that human preferences created models which fell into "reward pits" since humans prefer exploratory trajectories. From this, I expect that DemPref may perform better in a game which requires more exploration since it does not expect demonstrations to be optimal like Ibarz et al.[ILP+18]'s model. We also see from the T-REX paper that it has a hard time approximating more complex reward functions, so in those cases DemPref may perform better. I think in the future it may be worth evaluating these models in other domains.

Additionally, the original T-REX model used screenshots of the game screen to train the model, while I used more specific features defined in DemPref. How does this change in dimensionality of the features affect the algorithm? While the subset of features given in DemPref seem to be more relevant for the task, there may be some features we have not taken into account. How do we maximize the quality of the data provided to the model?

8 Conclusion

DemPref and T-REX did not perform too well on the Lunar Lander task, but this did not mean that there were no significant results. It seems that DemPref may be a superior algorithm because it outperforms T-REX in a similar amount of time and also seems to be easier to train. I believe These results may be specific to the lunar lander task, it would be interesting to see how they compare in different domains.

References

[BGNN19] Daniel S. Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations, 2019.

[BYH+17] Chandrayee Basu, Qian Yang, David Hungerman, Mukesh Singhal, and Anca D. Dragan. Do you want your autonomous car to drive like you? In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, March 2017.

[ILP+18] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari, 2018.

[LR21] Cassidy Laidlaw and Stuart Russell. Uncertain decisions facilitate better preference learning, 2021.

[PLSS19] Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions by integrating human demonstrations and preferences, 2019.