Project Report: Reducing Human's Burden in Deep Inverse Reinforcement Learning From Human Feedback

Negar Mehr and Amin Ghafari

December 5, 2017

1 Introduction

Deep reinforcement learning is proven to be able to successfully perform complex and complicated tasks. A key element in the anatomy of almost every reinforcement learning algorithm is the existence of a well-defined reward function which quantitatively evaluates the consequence of each action. However, such reward functions may not necessarily be a priorly known or formally defined even for simple tasks such as manipulation tasks. To address this hindrance, researchers have proposed Inverse Reinforcement Learning (IRL) [1] for learning such reward functions from demonstrations provided by an expert like a human. This implies that deployment of IRL requires an expert that is capable of demonstrating how to accomplish a task.

However, existence of an expert's demonstration might not necessarily be the case. For instance, when a robot wishes to perform a task that a human cannot achieve, the traditional IRL approaches render infeasible. Nevertheless, even in such cases, an expert such as a human might still be able to distinguish an expert–like demonstration from a certain set of demonstrations. In particular, an expert might be able to describe his/her preference in a query. Therefore, such an information can be captured to obtain the reward function required for a task of interest. Aligned with this, there has been a large body of literature on how to take advantage of experts', specifically humans' capability of providing preferences in the learning process of artificially intelligent agents. Examples of such works include [2], [3], [4], [5], [6], [7], [8] and [9]. These research works tackled the problem of obtaining a reward function from human's preferences. In some of these works, the obtained reward function was further used by an

RL algorithm for decision making.

In [10], IRL for continuous tasks with a large number of degrees of freedom was considered. Due to complex nature of the environment and tasks, the reward was represented by a neural network in this work and preference elicitation was performed over short clips of agent trajectories rather than demonstrating the whole trajectories. A human was asked to choose between a pair of short clips of agent's performance. The main contribution of this work was scaling the IRL framework up to large and complex domains. Later on, deep TAMER was proposed in [11] where promising results in an atari environment were achieved by a very small number of queries from human. In each query, the human was asked to assign a number from 1 to 5 to the performance of the agent. However, providing such rankings and feedback on continuous complex tasks might be very difficult for humans. As a result, in such settings, asking for preferences over a pair of trajectories might be more feasible.

In this work, we aim to reduce the human's burden and number of required queries from human in a framework similar to that of [10]. We are looking for ways to extract the amount of information from the minimum number of queries. In particular, we examined ways to generate informative queries so as to boost the learning process. In the sequel, we describe the ideas we have explored and their corresponding results.

2 Incorporating A Critique

When running the framework of [10], we realized that for a majority of queries, the human cannot decide between the two shown trajectories. In other words, for most of the queries, the human is indecisive. Therefore, we decided to ask for further information when the human is indecisive. In such cases, we ask for whether the two trajectories are both satisfactory or not. Inspired by [12], When such information is available, we train a critique for learning the informative pairs of trajectories, which in fact learns to demonstrate a satisfactory trajectory versus an unsatisfactory trajectory. We tried different network architectures for our critique network. The following subsections describe the effect of each choice and the results we got in each case.

2.1 Fully Connected Critique

When training a fully connected network as the critique, we consider the critique as a classifier. We assume two labels: satisfactory versus unsatisfactory and train a neural network with cross–entropy loss function. Then, we utilize the

	Hooper	Ant
Baseline	131	54
Critique	202	117

Table 1: The number of informative queries out of the first 1000 generated queries.

trained classifier to decide whether a trajectory segment would be satisfactory or not. Then, amongst a set of generated trajectories, we create a trajectory pair from two trajectories: a trajectory that the critique predicts to be satisfactory with high probability versus a trajectory that the critique predicts to be unsatisfactory with high probability.

When such a framework is used, the number of informative queries increases significantly. As Table 1 shows, the number of informative queries is doubled via the critique utilization in the two tested environments.

After observing the fact that the critique can successfully identify the informative queries, we decided to go beyond generating queries based on critique. What we do is that when the critique is pretty confident about the labels of trajectories in a query, we let the critique synthetically label the query and provide its preference, and further add this labeled pair to the training data set of reward function. This allows us to provide more data for learning the reward function while the human is not actually providing any feedback. Figure 1 shows the learning curve of this approach versus the baseline approach. The baseline approach requires a total of about 6000 queries from the human up to time step 4500 whereas our approach took a total of around 1200 queries. As Figure 1 demonstrates similar learning behavior is observed while providing much less number of queries. It is important to note that our numbers are not comparable to that of the [10] since our computational platforms are different. Since we have used our personal laptops for running our study, the number of generated and required queries are different from that of [10].

2.2 Recurrent Neural Network as Critique

When deploying a critique for evaluating the quality of a trajectory pair, we also investigated the usefulness of training an RNN for learning which trajectory pairs were informative and which were not. In particular, for each trajectory pair, we considered two labels: informative versus uninformative and trained an

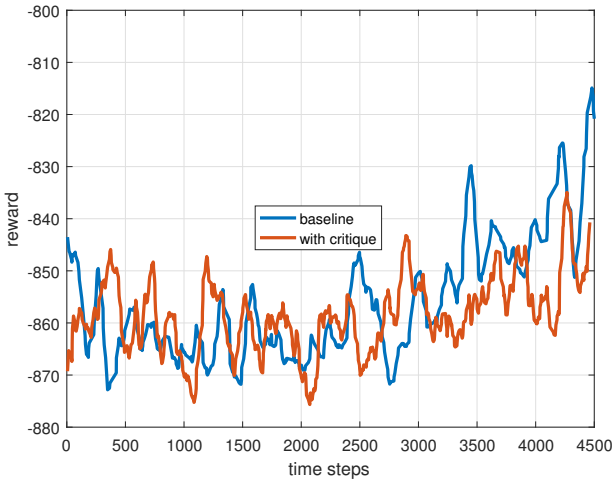

Figure 1: Learning curve of the baseline approach versus the critique–based approach. The baseline approach took about 6000 human queries whereas the critique–based approach required about human 1200 queries in the ant environment.

RNN using the queried pairs and their corresponding labels. We were hoping that our RNN critique would be capable of distinguishing the informative pairs. However, when we implemented the approach, the results were not promising, and our RNN critique was not able to identify the informative queries.

3 Bootstrap Utilization

We saw that critique–based query generation can be very useful specifically at the beginning of the learning process where separating good trajectories from the badly–behaved ones is doable. We believe that even if a critique is not used, we can further improve the agent's learning behavior by more intelligently generating queries. In particular, we are considering bootstrapping as a method of incentivizing for exploration. In this section, we describe how we utilized bootstrapping for explorative query generation and what results it leads to.

We consider bootstrapping in both rewards and policies. Let $\mathcal{D}_R$ represent the data set we use for training the reward function and $\mathcal{D}_\pi$ represent the data set we use for training the agent's policy via TRPO. We assume that we define J rewards through J deep networks $R^j, 1 \leq j \leq J$ and $K+1$ policies through K deep networks $\pi^k, 1 \leq k \leq K$. At every iteration of the learning process, we divide $\mathcal{D}_R$ into to J subsets $\mathcal{D}_R^1, \cdots, \mathcal{D}_R^J$ and $\mathcal{D}_\pi$ into K subsets $\mathcal{D}_\pi^1, \cdots, \mathcal{D}_\pi^K$

while allowing for replacements.

Utilizing the introduced data sets, we use $\mathcal{D}_R^j$ for training the j_{th} reward R^j and use $\mathcal{D}_\pi^k$ for training the k_{th} policy. We also keep training a nominal policy π^0 on the whole data set $\mathcal{D}_\pi$ using TRPO. Then, in order to ask for human's preference, we generate K queries, where the k_{th} query is obtained from two trajectory segments: a trajectory segment resulting from the baseline policy versus the trajectory segment resulting from the k_{th} policy. Then, we ask the J trained reward functions to label the K generated queries. We pick the query which receives the maximum disagreement among the rewards as the one we should show the human. This in fact captures that fact that we ask for human's preference on a trajectory pair which has the maximum uncertainty.

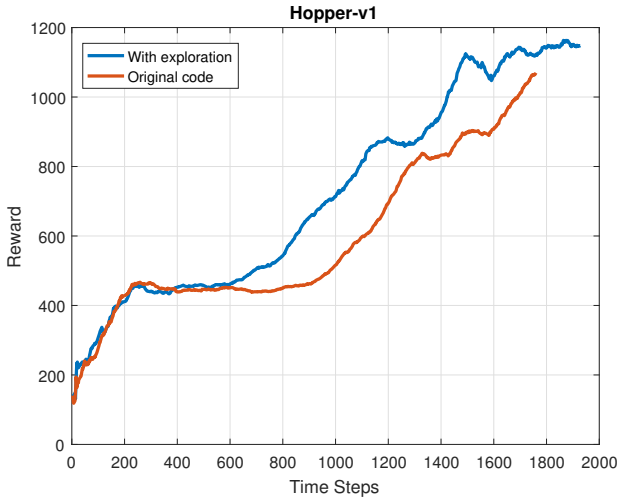

Figure 2: Learning curve of the baseline approach versus the explorative approach in the hopper environment with synthetic feedback.

We ran our query generation method in the hopper environment and compared it to the baseline query generation method used by [10]. We used three extra policies for exploration in the policy space and three reward functions for capturing the uncertainty in reward. It is important to note that since we our computational capabilities were limited, we used synthetic feedback in this experiment for labeling the queries. In other words, the queries were labeled based off of the true reward value that each segment would have received. As Figure 2 demonstrates, the explorative policy has better performance and learning behavior compared to that of the nominal one. It is important to mention that we expect that larger improvements in the performance of our query generation method is expected with a larger number of extra policies and rewards.

Unfortunately, we could not try such settings on our personal computers.

4 Future Work

As of future directions, we are interested in examining the practicality of our proposed ides in more a more divers set of experiments and tasks. Furthermore, it would be interesting to combine the critique with explorative query generation. In particular, in would be interesting to utilize critique in the early stages of learning where bootstrapping in query generation does not make a difference (as seen in Figure 2) and later on use exploration to generate queries. The value of this idea will be highlighted in the environments such as ant where the early stage of training is time-consuming.

As a next stage, we believe that asking for how much a trajectory is preferred over the other one in a query pair might be informative as well. Asking for human's preference on a Likert scale would allow for extracting more information from the queries while not incurring a huge burden on human in tasks with which the human might be unfamiliar.

References

[1] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[2] Riad Akrour, Marc Schoenauer, and Michele Sebag. Preference-based policy learning. *Machine learning and knowledge discovery in databases*, pages 12–27, 2011.

[3] Patrick M Pilarski, Michael R Dawson, Thomas Degris, Farbod Fahimi, Jason P Carey, and Richard S Sutton. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE, 2011.

[4] Riad Akrour, Marc Schoenauer, and Michèle Sebag. April: Active preference learning-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 116–131. Springer, 2012.

[5] Hiroaki Sugiyama, Toyomi Meguro, and Yasuhiro Minami. Preference-learning based inverse reinforcement learning for dialog control. In *Thir-*

teenth Annual Conference of the International Speech Communication Association, 2012.

[6] Christian Daniel, Oliver Kroemer, Malte Viering, Jan Metz, and Jan Peters. Active reward learning with a novel acquisition function. *Autonomous Robots*, 39(3):389–405, 2015.

[7] Layla El Asri, Bilal Piot, Matthieu Geist, Romain Laroche, and Olivier Pietquin. Score-based inverse reinforcement learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 457–465. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[8] Christian Wirth, J Furnkranz, Gerhard Neumann, et al. Model-free preference-based reinforcement learning. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 2222–2228, 2016.

[9] Sida I Wang, Percy Liang, and Christopher D Manning. Learning language games through interaction. *arXiv preprint arXiv:1606.02447*, 2016.

[10] Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*, 2017.

[11] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. *arXiv preprint arXiv:1709.10163*, 2017.

[12] Yuchen Cui and Scott Niekum. Active learning from critiques via bayesian inverse reinforcement learning.