Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Active Advice Seeking for Inverse Reinforcement Learning

Phillip Odom and **Sriraam Natarajan**
School of Informatics and Computing
Indiana University Bloomington
phodom,natarasr@indiana.edu

Abstract

Intelligent systems that interact with humans typically require demonstrations and/or advice from the expert for optimal decision making. While the active learning formalism allows for these systems to incrementally acquire demonstrations from the human expert, most learning systems require all the advice about the domain in advance. We consider the problem of actively soliciting human advice in an inverse reinforcement learning setting where the utilities are learned from demonstrations. Our hypothesis is that such solicitation of advice reduces the burden on the human to provide advice about every scenario in advance.

Learning from demonstrations in a sequential decision-making setting has recently received much attention under the formalism of Inverse Reinforcement Learning (IRL) (Ng and Russell 2000; Abbeel and Ng 2004). These methods assumed that the given demonstrations were optimal and potentially required a significant number of examples before they were able to learn a suitable reward function. Kunpauli et al. (2013) built on this work by allowing an expert to provide advice about certain scenarios in addition to the demonstrations. This resulted in learning from significantly fewer and (possibly) even sub-optimal demonstrations. While their framework succeeded in reducing the effort of the expert when providing demonstrations, the expert was required to identify the quantity of advice as well as the important features of states where advice might help. Thus it removed the burden of trajectories while introducing a lesser burden of providing targeted advice. An ideal IRL algorithm should determine where to solicit advice and when it has accumulated sufficient advice. We propose to employ the concept of *active learning* (Settles 2012) for this task.

Active Learning is a paradigm that has been extensively applied to supervised learning where the learning algorithm works by (actively) selecting a small but representative set of examples to be labeled by the expert. They are able to learn a more accurate classifier while at the same time minimizing the number of expert labeled examples.

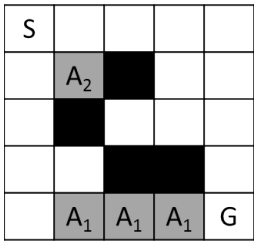

Figure 1: Advice for the WumpusWorld domain. A_1 and A_2 mark the pieces of advice about which the learning algorithm might ask.

While active learning has been traditionally applied to soliciting example labels, it has not been considered in the realm of seeking advice from an expert which is crucial in reducing the burden of the expert when providing the combination of trajectories and advice. The goal now is to select areas in the feature space to solicit advice from the expert. Actively seeking advice has two distinct advantages over active learning. While active learning is constrained to selecting a single example to label, an active advice seeker can get advice over a larger section of the feature space. This allows the learner to query about similar areas of the feature space together potentially reducing the number of advice that the learning algorithm requires. Traditional active learning is also limited in the communication with the expert as the only information the expert is allowed to give is a label for a given example. Advice can be much more expressive, for example by allowing a set of potential classes instead of a single class. Consider the example in figure 1 which shows a WumpusWorld domain common in reinforcement learning. In WumpusWorld, the goal is to navigate from the starting state (S) to the goal state (G) while avoiding the obstacles. Two potential advice areas are labeled. Notice how with advice you can query over multiple states that are deemed to be similar. An example of advice for A_1 would be that the agent should prefer to move right and should not prefer to move up, down or left.

Active Advice Seeking

The main difficulty for active advice seeking is deciding the parts of the state/action space where advice would be most beneficial. A common method in traditional active learning is the uncertainty (using entropy or some other uncertainty measure) w.r.t each state in the feature space (Settles 2012). The idea is to ask for more information about states about which the learner is more uncertain. While the first step of

Algorithm 1 Active Advice Seeking IRL Algorithm

Require: Demonstrations (Δ), Maximum Advice (M)
Require: N number advice to ask for at once
Require: $Expert(S)$ which returns advice for $s_i \in S$
  $Advice = \emptyset$
  while $|Advice| < M$ **do**
    $Reward = AIRL(\Delta, Advice)$
    $Uncertainty(x) = u(x)$
    $S = HighestUncertainty(Uncertainty, N)$
    $A = Expert(S)$
    $Advice = Advice \cup A$
  end while
  return $AIRL(\Delta, Advice)$

IRL is to learn a reward function, the final output of interest is still a policy, i.e., a mapping from each state to a distribution over the actions. The uncertainty with respect to the reward function may not be equivalent to the uncertainty with respect to the policy as there are many reward functions that result in the same policy (Lopes, Melo, and Montesano 2009). For example in the WumpusWorld example, a reward function with +10 in the goal state and -1 for all other states will correspond to a policy that finds the shortest path to the goal. However, a reward function that gives +50 in the goal state and -2 in all other states will correspond to the same policy. To counter this issue, our approach seeks to combine two potential sources of uncertainty in IRL. The uncertainty might be either due to the paucity of demonstrations or due to the final policy learned from the reward function.

The uncertainty $u(x)$ of a state x is a weighted combination of the uncertainty due to the demonstrations and the resulting policy and is given by $u(x) = w_p f(x) + (1 - w_p) g(x)$ where $f(x)$ is the uncertainty w.r.t the number of demonstrations provided for state x, w_p is the corresponding weight and $g(x)$ is a the uncertainty w.r.t the policy learned for x.

Uncertainty w.r.t the demonstrations ($f(x)$) can be calculated as the entropy of the distribution of actions taken by the expert in every state x. Laplace correction can be used as the expert may never visit some states or take every action in every state. We expect that as the number of actions taken in a particular state by the demonstrator increases, there will be less uncertainty about the reward learned in that particular state. Thus, $f(x) = H(A)$ where H is entropy and A is the set of admissible actions in x.

Uncertainty w.r.t the policy is more complicated as the final policy is only optimal w.r.t the learned reward function (Sutton and Barto 1998) which is calculated from the set of demonstrations and advice. As the advice changes, new reward functions must be induced leading to different uncertainty over the policy space unlike the uncertainty over the demonstrations which remain constant. If we use Boltzmann distribution to compute the policy given the Q-values,

$$p(x, a) = \frac{e^{Q(x,a)}}{\sum_{b \in A} e^{Q(x,b)}} \quad (1)$$

entropy can be used to calculate uncertainty over the policy.

It is important to understand that the states chosen to solicit advice are the best states to get w.r.t the most current reward function. After getting advice, the learner should be able to learn an improved reward function that will in turn provide a potentially different set of uncertain states from which to collect additional advice. This iterative procedure is presented in Algorithm 1.

Table 1: Results - We show the percentage of games won (reached *G*) for each method while varying the number of advice solicited.

	1 adv	3 adv	5 adv	10 adv
Uncertainty	0	.1	6.1	33.3
Random	3.0	1.0	9.7	9.9
Only Trajectories	.8			

We present initial results from WumpusWorld (figure 1) where the goal is to navigate the grid starting in state *S*, with the goal of getting to state *G*. We assume that in two of the key states the demonstrator provides a potential suboptimal training example (he chooses a random action). While varying the number of advice, we compare our approach for choosing states to receive advice (*Uncertainty*) to both a random sampling approach (*Random*) as well as naively using only the trajectories (*Only Trajectories*). We compare the winning percentage (reached *G*) of the policies learned from each method. Our method outperforms both baselines when it can ask for sufficient advice to learn its policy. While our initial results are in small domains, we are currently working to expand this work to large, complex MDPs through the use of factored state spaces as well as considering the application of advice to relational IRL.

Conclusion

Active advice seeking leverages both demonstrations and general advice from an expert in guiding the agent to learn an appropriate reward function and thus policy. This work is the first step towards the ability to solicit advice over potentially large sets of similar states. Finally, an extension of our work could be to model for exactly how much advice is sufficient for the learner to accomplish its goal. Each direction significantly reduces expert effort in learning policies.

Acknowledgements: The authors thank Army Research Office grant number W911NF-13-1-0432 under the Young Investigator Program.

References

Abbeel, P., and Ng, A. 2004. Apprenticeship learning via inverse reinforcement learning. In *ICML*.

Kunapuli, G.; Odom, P.; Shavlik, J.; and Natarajan, S. 2013. Guiding autonomous agent to better behaviors through human advice. In *ICDM*.

Lopes, M.; Melo, F.; and Montesano, L. 2009. Active learning for inverse reinforcement learning. In *ECML*.

Ng, A., and Russell, S. 2000. Algoithms for inverse reinforcement learning. In *ICML*.

Settles, B. 2012. *Active Learning*. Morgan and Claypool.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press.