KWIK Inverse Reinforcement Learning

Vaishnavh Nagarajan
Indian Institute of Technology Madras
vaish@cse.iitm.ac.in

Balaraman Ravindran
Indian Institute of Technology, Madras
ravi@cse.iitm.ac.in

Abstract

Imitation learning or learning from demonstrations is a means of transferring knowledge from a teacher to a learner, that has led to state-of-the-art performances in many practical domains such as autonomous navigation and robotic control. The aim of the learning agent is to learn the expert's policy through trajectories demonstrated by the expert. One solution to this problem is inverse reinforcement learning (IRL), where the learner infers a reward function over the states of the Markov Decision Process on which the mentor's demonstrations seem optimal. However, since expert trajectories are practically expensive, it becomes crucial to minimize the number of trajectory samples required to imitate accurately. Moreover, when the state space is large, the agent must be able to generalize knowledge acquired from demonstrations covering a small subset of the state space, confidently to the rest of the states. To address these requirements, we first propose a novel reduction of IRL to classification where determining the separating hyperplane becomes equivalent to learning the reward function itself. Further, we also use the power of this equivalence to propose a Knows-What-It-Knows (KWIK) algorithm for imitation learning via IRL. To this end, we also present a novel definition of admissible KWIK classification algorithms which suit our goal. The study of IRL in the KWIK framework is of significant practical relevance primarily due to the reduction of burden on the teacher: a) A self-aware learner enables us to avoid making redundant queries and cleverly reduce the sample complexity. b) The onus is now on the learner (and no longer on the teacher) to proactively seek expert assistance and make sure that no undesirable/sub-optimal action is taken.

Keywords: Reinforcement Learning; Markov Decision Process; Learning from Demonstrations; Imitation Learning; KWIK; Knows what it knows; Online learning; Inverse Reinforcement Learning;

1 Introduction

Imitation learning is broadly solved in two different ways. One approach is to pose it as a supervised learning problem where a classifier learns the action labels for all states in the state space based on training data (the expert's demonstrations). The other approach is a model-based solution that uses inverse reinforcement learning (IRL) to find a mapping from states to real-valued rewards that makes the expert trajectories seem optimal. The learner hence follows the optimal policy on these rewards. IRL methods have the advantage of representing the acquired knowledge succinctly as rewards over the states.

In practice, both these approaches could suffer from a considerable burden on the teacher who is expected to produce sufficient trajectories for accurate imitation. What makes this more undesirable is that many of the trajectories happen to be redundant and yet expensive. To this end, Judah et al. [4] have proposed active learning algorithms for supervised learning based imitation learning and analyse their PAC label complexity. On the other hand, there has been very little work on formally understanding active imitation learning through IRL. Silver et al. [8] have studied active learning heuristics where the learner requests trajectories in such a way that the knowledge about the reward function acquired is either novel or reduces uncertainty in the current beliefs. Lopes et al. [6] only provide an empirical technique to actively query by choosing states with the greatest uncertainty with respect to the policy that was learnt using Bayesian IRL.

A drawback with all of the above active learning algorithms is that they assume that the learner has complete access to the state space and can also query the expert for a demonstration on any of these states. Often this might not be desirable because some states may not even be realizable and furthermore, this knowledge might not be accessible to the learner. Chernova and Veloso [2] address this by allowing the learner to 'interactively' request the teacher's demonstrations whenever it encounters a state where its confidence on the learnt policy is below a threshold. They provide an algorithm that pertains only to the supervised learning based imitation learning and support it with empirical results.

To overcome these multiple issues, we propose the novel idea of considering imitation learning in the Knows-What-It-Knows (KWIK) framework [5]. A KWIK algorithm is an online learning algorithm that is considered to be self-aware i.e., if and only if the learner believes that it has insufficient experience to predict on a new sample, does the learner ask the expert for the answer. Considering imitation learning in this framework significantly benefits us in four ways. First, we overcome the problems that come with allowing the learner to query on any arbitrary state. The learner makes queries only on the states that it encounters. Secondly, we are able to allow the learner to enact its policy and learn on-the-fly. Thirdly, the burden on the expert is substantially reduced as the learner only selectively requests demonstrations. Finally, we are guaranteed that the learner does not mistakenly assume that it knows what to do when it actually does not. This is of practical importance because we would not want the learner to take a non-optimal and possibly dangerous action which could have been avoided if the expert had intervened.

Though Walsh et al. [10] study what is called as a generalized apprenticeship learning protocol in relation to KWIK learnable classes, their problem domain, as they claim, is fundamentally different from the imitation learning problem that we consider. They study a learner that has access to the rewards during exploration, while the teacher augments this knowledge.

Next, we propose a reduction of the KWIK apprenticeship learning problem via IRL to KWIK classification. Note that this reduction to classification is not the same as direct imitation learning methods that use a classifier to learn a mapping from states to actions. We are primarily interested in finding the unknown reward function defined over the state-action pairs and not just what action label a state corresponds to. We show that learning the reward function is equivalent to learning the separating hyperplane in the classification problem.

Our next contribution in this work is a novel definition of KWIK classification that applies to this equivalence in imitation learning. The KWIK framework requires that the learner achieves point-wise accuracy, unlike in a PAC-learner. That is, if the learner makes a prediction on a new sample without seeking expert advice, the learner must be ϵ-accurate. However, it is not possible to define ϵ-accuracy on discrete labels (unless we consider a continuous action space in which case we would opt for KWIK online regression algorithms [9]). One could overcome this by defining accuracy with respect to the prediction about the distance of the sample from the separating hyperplane, as considered in some *selective sampling* algorithms [1, 3]. However, these algorithms have significantly different assumptions than that expected by the KWIK imitation learning agent and are hence not applicable. We further motivate the validity of our definition of KWIK classification by providing polynomial KWIK bounds for 1-D classification. Finally, we also provide a KWIK protocol for imitation learning that uses an underlying KWIK classifier that suits our requirement that the learner takes ϵ-optimal actions.

2 Preliminaries

Definition A Markov Decision Process (MDP) is represented as a 5-tuple (S, A, T, γ, R) where S is a set of states; A is a set of actions; T is a set of state transition probabilities; $\gamma \in (0, 1]$ is a discount factor; and $R : S \rightarrow A$ is the reward function.

We assume that the state-action pairs can be mapped to a $k-$dimensional vector of features, $\phi : S \times A \rightarrow [0,1]^k$. For example, in a maze where a puddle has to be avoided we could have three boolean features each describing whether the state is a puddle or a goal or neither. Thus, the actual reward $R(s,a)$ corresponding to an action at a state is equal to $\vec{w} \cdot \phi(s,a)$ where $w \in \mathbb{R}^k$. We need to ensure that $||w||_1 \leq 1$ so that the rewards themselves are upper-bounded by 1.

A policy π is a mapping from states to (probability distributions over) actions. The value of a state-action pair under a policy π is

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_\pi[R(s,a) + \sum_{t=1}^{\infty} \gamma^t R(s_t,a_t)|\pi] \\ &= \vec{w} \cdot \mathbb{E}_\pi[\phi(s,a) + \sum_{t=0}^{\infty} \gamma^t \phi(s_t,a_t)|\pi] \end{aligned}$$

where $s_1, s_2 \ldots$ are the subsequent states visited by following the policy and $a_1, a_2 \ldots$ are the corresponding actions taken at those states. Thus we see that the $Q-$values can also be linearly parametrized. We assume that the learning agent is presented these $Q-$value parameters in the form of $\phi_Q : S \times A \rightarrow \mathbb{R}^k$. The IRL task now reduces to interacting with the expert and making an accurate estimate of the vector $\vec{w}$.

Definition We define the **Knows-What-It-Knows protocol** with parameters (ϵ, δ) ($\epsilon \in [0,1], \delta \in [0,1)$) as follows . Consider a hypothesis space $H \subseteq (X \rightarrow Y)$ from which the adversary picks a target function $h^* \in H$. During a run, for each time-step the adversary picks an input $x \in X$ for which the learner either emits a $\perp$ (dont-know) or makes a prediction $\hat{h}(x)$. If the learner emits $\perp$, the adversary informs the learner of $h^*(x)$. For an algorithm to be an admissible KWIK algorithm, we need that with probability $1-\delta$ the following conditions hold good:

- Whenever the algorithm makes a prediction $\hat{h}(x)$, $|\hat{h}(x) - h^*(x)| \leq \epsilon$
- The number of time-steps for which the algorithm emits $\perp$ is bounded by $B(\epsilon, \delta)$ a function that is polynomial in $1/\epsilon$, $1/\delta$ and some parameters that define H.

3 KWIK-Learner for Binary Classification

Definition We define **an admissible KWIK-learner for binary classification** as follows. We assume that the hypothesis class is the set of separating hyperplanes $\{\vec{w} | \vec{w} \in \mathbb{R}^k, ||\vec{w}||_1 \leq 1\}$. If the adversary picks a $\vec{w}^*$, the correct label of $\vec{x}$ is given by $\text{SGN}(\vec{w}^* \cdot \vec{x}) \in \{+1, -1\}$. For the learner to be admissible, the following must hold good for every run, with probability $1-\delta$:

- Whenever the algorithm makes a prediction on x, if $|\vec{w}^* \cdot \vec{x}| > \epsilon$, then $\hat{h}(\vec{x}) = \text{SGN}(\vec{w}^* \cdot \vec{x})$
- The number of time-steps for which the algorithm emits $\perp$ is bounded by $B(\epsilon, \delta)$ a function that is polynomial in $1/\epsilon$, $1/\delta$ and k.

Intuitively, we require that the algorithm predicts correctly on all the points that are sufficiently far away from the separating hyperplane; if the sample point is within the $\epsilon-$margin of the hyperplane, we allow the classifier to make mistakes.

This may be compared to the KWIK-MB model proposed by Sayedi et al. [7] where the KWIK algorithm is also allowed to make a fixed number of mistakes. However, our model is significantly different in that we allow infinitely many mistakes but restrict them to a very small space around the separating hyperplane. If we did not allow the learner to make infinitely many mistakes, we would expect the learner to perennially refine its knowledge in the small space around the hyperplane. This might require exponentially many queries to accurately place the hyperplane. Furthermore, we will see that our condition also eventually suits our imitation learning problem where the learner is required to be ϵ-optimal.

Next, we discuss assumptions about noise in the expert's labels. In the KWIK framework, we assume that the noisy observation produced by the expert has an expectation equal to the correct output. Thus, for classification we assume a teacher to be (ϵ_T, ϵ_Y)-optimal, if the teacher outputs y for an input x such that:

$$\begin{aligned} \mathbb{E}[y] &> \epsilon_Y \quad \text{if } \vec{w}^* \cdot \vec{x} \geq \epsilon_T \\ \mathbb{E}[y] &< -\epsilon_Y \quad \text{if } \vec{w}^* \cdot \vec{x} \leq -\epsilon_T \end{aligned}$$

In other words, we expect that for all input points that are at least ϵ_T away from the separating hyperplane, the expert labels them correctly with probability at least $1/2 + \epsilon_Y/2$. A good teacher will have a high ϵ_Y and a small ϵ_T. We note that this is a significantly relaxed assumption when compared to the selective sampling approach of Dekel et al. [3] where they assume that the accuracy of the expert increases with the distance from the separating hyperplane.

3.1 A simple KWIK 1-D classification algorithm

We analyse a naive algorithm for 1-D classification to demonstrate how the KWIK conditions we proposed allow us to design algorithms with a KWIK-bound polynomial in $1/\epsilon$ and $1/\delta$ even under the relaxed noise assumption of the teacher's outputs. Assume the input space spans unit length and that $4\epsilon_T < 2\epsilon < \epsilon_Y$, which is natural because it is not possible for the learner to outdo the teacher.

The learning algorithm discretizes the input space into $\frac{2}{\epsilon}$ segments. When the adversary presents a sample belonging to a segment, the algorithm emits $\perp$ when the number of samples already queried in this segment is fewer than $\mathcal{O}(\frac{1}{\epsilon^2}\ln(\frac{2}{\epsilon\delta}))$. When the number of samples is however greater than this, we can show that if the segment is completely outside the ϵ_T-margin around the separating point, the proportion of queried points in this segment that would have been labelled correctly by the expert will be at least $1/2 + \epsilon_Y/2 - \epsilon > 1/2$ with a high probability of $1 - \epsilon\delta/2$. Thus, after acquiring sufficient samples in each of the segments, we will correctly learn the labels of all the segments outside an $\epsilon-$ margin of the separating point with probability $1 - \delta$. Thus, the number of queries made will be $\mathcal{O}(\frac{1}{\epsilon^3}\ln(\frac{2}{\epsilon\delta}))$.

4 KWIK Inverse Reinforcement Learning Protocol

We now present the reduction of IRL to KWIK classification. At any state $s \in S$, the learner is presented with a set of at most $|A|$ actions of which the learner is required to pick an $\epsilon-$optimal action. Let $\vec{w}$ be the unknown weight vector for the rewards. We assume that the learner has access to a KWIK classification algorithm whose input space is $\mathbb{R}^k$ and whose accuracy parameter is set to be $\epsilon/(|A| - 1)$.

For some $a^*, a' \in A$, we expect the classifier to predict $\text{SGN}(\vec{w} \cdot (\phi_Q(s, a^*) - \phi_Q(s, a')))$ given $(\phi_Q(s, a^*) - \phi_Q(s, a'))$ as input. If it predicts, say, $+1$, from the conditions we stipulated, we know that $\vec{w} \cdot (\phi_Q(s, a^*) - \phi_Q(s, a')) > \epsilon/(|A| - 1)$. We will use this property to use the classifier to identify the action that corresponds to nearly the highest Q-value.

If the classifier is unable to predict, and instead outputs a $\perp$ we request expert advice in the form of a preference over these pair of actions. We note that we could study various other modifications of this algorithm, where the expert only provides knowledge about the best action amongst all actions instead of pairwise preferences.

Algorithm 1 KWIK Inverse Reinforcement Learning Protocol

Require: Teacher $\mathcal{T}(\epsilon_T, \epsilon_Y)$ with true weight vector for rewards $\vec{w}$, Admissible KWIK Classifier $\mathcal{C}(\frac{\epsilon}{|A|-1}, \delta)$ with weight estimate for rewards $\hat{\vec{w}}$

```
for t = 1, 2, ... do
    s = Current State of the Environment
    â_best = a_1
    for i = 2, ... |A| do
        Present φ_Q(s, a_i) − φ_Q(s, â_best) to C
        if Output of C = ⊥ then
            Present φ_Q(s, a_i) − φ_Q(s, â_best) to T
            T outputs noisy observation of SGN(w⃗ · (φ_Q(s, a_i) − φ_Q(s, â_best)))
            C learns from output of T and updates ŵ⃗
            if Output of T = +1 then
                â_best = a_i
        else
            C outputs SGN(ŵ⃗ · (φ_Q(s, a_i) − φ_Q(s, â_best)))
            if Output of C = +1 then
                â_best = a_i
    Perform â_best
```

In Algorithm 1, at any state the learner scans all the possible actions (which is at most $|A|$) and maintains a candidate action that it considers to be the best amongst the actions that have been iterated through. The correctness of this algorithm would follow if we show that after iterating over all the actions, if the algorithm has not queried the teacher, it always chooses an ϵ-optimal action. We prove the following lemma from which the above statement follows directly by setting $i = |A|$.

Lemma 4.1 *After iterating over the first i actions, if the algorithm has not made any queries, the candidate action picked by the algorithm is $(i-1)\frac{\epsilon}{|A|-1}-$ optimal with respect to the best action amongst the first i actions.*

Proof The claim clearly holds good when $i = 1$. For any arbitrary round $i < |A|$ assume that the claim is true. That is, if $\hat{a}_i$ is the candidate action picked by the algorithm, and a_i^* is the best action amongst the first i actions, then:

$$\vec{w} \cdot \phi_Q(s, \hat{a}_i) \geq \vec{w} \cdot \phi_Q(s, a_i^*) - (i-1)\frac{\epsilon}{|A|-1} \tag{1}$$

If the algorithm chose a_{i+1} over $\hat{a}_i$, we know from the KWIK classifier conditions that

$$\vec{w} \cdot \phi_Q(s, a_{i+1}) \geq \vec{w} \cdot \phi_Q(s, \hat{a}_i) - \frac{\epsilon}{|A|-1} \tag{2}$$

If this decision were to be inconsistent with our claim, a_{i+1} must not be an $i\frac{\epsilon}{|A|-1}$-optimal action (among the $i+1$ actions). Then a_i^* must still be the best action amongst the first $i+1$ actions. However, from inequalities (1) and (2), we can see that:

$$\vec{w} \cdot \phi_Q(s, a_{i+1}) \geq \vec{w} \cdot \phi_Q(s, a_i^*) - i\frac{\epsilon}{|A|-1}$$

which makes a_{i+1} an $i\frac{\epsilon}{|A|-1}$-optimal action amongst the first $i+1$ actions, which is a contradiction.

On the other hand, if the algorithm still chose $\hat{a}_i$ over a_{i+1}, it would be inconsistent with our claim only if a_{i+1} was the best action amongst the $i+1$ actions and if $\hat{a}_i$ was not sufficiently optimal. That is,

$$\vec{w} \cdot \phi_Q(s, \hat{a}_i) < \vec{w} \cdot \phi_Q(s, a_{i+1}) - i\frac{\epsilon}{|A|-1} \tag{3}$$

However, this implies a much weaker inequality:

$$\vec{w} \cdot \phi_Q(s, \hat{a}_i) < \vec{w} \cdot \phi_Q(s, a_{i+1}) - \frac{\epsilon}{|A|-1}$$

which would have ensured that the KWIK classifier indicated that a_{i+1} was a better action. Hence, by induction we show that the lemma holds good for all $i = 1, \ldots |A|$. ∎

5 Conclusion and Future Work

In this work, we have provided an understanding of IRL in the KWIK framework, which has not been studied before. Apart from the practical relevance of a KWIK imitation learning algorithm, this setup provides scope for theoretical guarantees that can be provided for active query based IRL which have not been provided so far. We have also laid the groundwork for a new class of algorithms that can be termed as 'KWIK classifiers' that will suit tasks similar to IRL-based imitation learning. Since the existing selective sampling based classification approaches do not apply here due to their strong assumptions, an appropriate direction for future work will be to design admissible KWIK classification algorithms that suit our conditions.

References

[1] Nicolò Cesa-Bianchi, Claudio Gentile, and Francesco Orabona. Robust bounds for classification via selective sampling. In *ICML 2009*, volume 382, pages 121–128.

[2] Sonia Chernova and Manuela M. Veloso. Interactive policy learning through confidence-based autonomy. *CoRR*, abs/1401.3439, 2014.

[3] Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Robust selective sampling from single and multiple teachers. In *COLT 2010*, pages 346–358.

[4] Kshitij Judah, Alan Fern, and Thomas G. Dietterich. Active imitation learning via reduction to I.I.D. active learning. In *UAI 2012*, pages 428–437.

[5] Lihong Li, Michael L. Littman, Thomas J. Walsh, and Alexander L. Strehl. Knows what it knows: a framework for self-aware learning. *Machine Learning*, 82(3):399–443, 2011.

[6] Manuel Lopes, Francisco S. Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In *ECML PKDD 2009 Part II*, pages 31–46.

[7] Amin Sayedi, Morteza Zadimoghaddam, and Avrim Blum. Trading off mistakes and dont-know predictions. In *NIPS 2010*, pages 2092–2100.

[8] David Silver, J. Andrew Bagnell, and Anthony Stentz. Active learning from demonstration for robust autonomous navigation. In *In IEEE ICRA*, pages 200–207.

[9] Alexander L. Strehl and Michael L. Littman. Online linear regression and its application to model-based reinforcement learning. In *NIPS 2007*, pages 1417–1424.

[10] Thomas J. Walsh, Kaushik Subramanian, Michael L. Littman, and Carlos Diuk. Generalizing apprenticeship learning across hypothesis classes. In *ICML 2010*, pages 1119–1126.