Lemma 3\(^1\) in (Syed and Schapire, 2010) shows that if for each time step \(\epsilon_t \leq \epsilon\), then \(V(\hat{\pi}) \geq V(\pi^*) - \epsilon T^2\). Hence, if we are interested in learning a \(\hat{\pi}\) whose overall regret is no more than \(\epsilon\), then we need to provide at least \(N_p(\hat{\pi}, \hat{\pi}, \frac{\epsilon}{T})\) examples to \(L_p\) to ensure that \(\epsilon_t \leq \frac{\epsilon}{T}\) holds at all time steps with probability at least \(1 - \delta\). Therefore, the passive label complexity of this algorithm is \(T \cdot N_p(\hat{\pi}, \hat{\pi}, \frac{\epsilon}{T})\).

Our goal now is to provide a reduction from active imitation learning to i.i.d. active learning that can achieve an improved label complexity. This is not as simple as replacing the calls to \(L_p\) in the above approach with calls to an active learner \(L_a\). This is because the active learner at time step \(t\) would require the ability to sample from the unlabeled distribution \(d^t\), which requires executing the expert policy for \(t\) steps, which in turn requires \(t\) label queries to the expert that would count against the label complexity.

It turns out that for a slightly more sophisticated reduction to i.i.d. passive learning introduced by Ross and Bagnell (2010), it is possible to simply replace \(L_p\) with \(L_a\) and maintain the potential benefit of active learning. Ross and Bagnell (2010) introduced the forward training algorithm for non-stationary policies, which trains a non-stationary policy in a series of \(T\) iterations. In particular, iteration \(t\) trains policy \(\hat{\pi}_t\) by calling a passive learner \(L_p\) on a labeled data set drawn from the state distribution induced at time \(t\) by the non-stationary policy \(\hat{\pi}^{t-1} = (\hat{\pi}_1, \ldots, \hat{\pi}_{t-1})\), where \(\hat{\pi}_1\) is learned on states drawn from the initial distribution \(I\). The motivation for this approach is to train the policy at time step \(t\) based on the same state-distribution that it will encounter when being run after learning. By doing this, Ross and Bagnell (2010) show that the algorithm has a worst case regret of \(T^2\epsilon\) and under certain assumptions can achieve a regret as low as \(O(T\epsilon)\).

Importantly, the state-distribution used to train \(\hat{\pi}_t\) given by \(d^t_{\hat{\pi}_{t-1}}\) is easy for the learner to sample from without making queries to the expert. In particular, to generate a sample the learner can simply simulate \(\hat{\pi}^{t-1}\), which is available from previous iterations, from a random initial state and return the state at time \(t\). Thus, we can simply replace the call to \(L_p\) at iteration \(t\) with a call to \(L_a\) with unlabeled state distribution \(d^t_{\hat{\pi}_{t-1}}\) as input. More formally, the active forward training algorithm is given by the following iteration:

\[
\hat{\pi}_{t} = L_a(\epsilon, \frac{\epsilon}{T}, D^1),
\]

where \(D^1 = I\) and \(D^t = d^t_{\hat{\pi}_{t-1}}\).

Theorem 3.1 in (Ross and Bagnell, 2010) gives the worst case bound on the regret of the forward training algorithm which assumes the generalization error at each iteration is bounded by \(\epsilon\). Since we also maintain that assumption when replacing \(L_p\) with \(L_a\) (the active variant) we immediately inherit that bound.

**Proposition 1.** Given a PAC i.i.d. active learning algorithm \(L_a\), if active forward training is run by giving \(L_a\) parameters \(\epsilon\) and \(\frac{\epsilon}{T}\) at each step, then with probability at least \(1 - \delta\) it will return a non-stationary policy \(\hat{\pi}^T\) such that \(V(\hat{\pi}^T) \geq V(\pi^*) - \epsilon T^2\).

Note that \(L_a\) is run with \(\frac{\epsilon}{T}\) as the reliability parameter to ensure that all \(T\) iterations succeed with the desired probability. Proposition 1 shows that the overall label complexity of active forward training in order to achieve a regret less than \(\epsilon\) with probability at least \(1 - \delta\) is \(T \cdot N_a(\frac{\epsilon}{T}, \frac{\epsilon}{T})\). Comparing to the corresponding label complexity of passive imitation learning \(T \cdot N_p(\frac{\epsilon}{T}, \frac{\epsilon}{T})\) we see that an improved label complexity of active learning in the i.i.d. case translates to improved label complexity of active imitation learning. In particular, in the realizable learning case, we know that \(N_a\) can be exponentially smaller than \(N_p\) in terms of its dependence on \(\frac{1}{\epsilon}\).

**4.2 Stationary Policies**

A drawback of active forward training is that it is impractical for large \(T\) and the resulting policy cannot be run indefinitely. We now consider the case of learning stationary policies, first reviewing the existing results for passive imitation learning.

In the traditional approach, a stationary policy \(\hat{\pi}\) is trained on the expert state distribution \(d_{\pi^*}\) using a passive learning algorithm \(L_p\) returning a stationary policy \(\hat{\pi}\). Theorem 2.1 in (Ross and Bagnell, 2010) states that if the generalization error of \(\hat{\pi}\) with respect to the i.i.d. distribution \(d_{\pi^*}\) is bounded by \(\epsilon\) then \(V(\hat{\pi}) \geq V(\pi^*) - \epsilon T^2\). Since generating i.i.d. samples from \(d_{\pi^*}\) can require up to \(T\) queries (see Section 3) the passive label complexity of this approach for guaranteeing a regret less than \(\epsilon\) with probability at least \(1 - \delta\) is \(T \cdot N_p(\frac{\epsilon}{T^2}, \delta)\).

The above approach cannot be converted into an active imitation learner by simply replacing the call to \(L_p\) with \(L_a\), since again we cannot sample from the unlabeled distribution \(d_{\pi^*}\) without querying the expert. To address this issue, we introduce a new algorithm called RAIL (Reduction-based Active Imitation Learning) which makes a sequence of \(T\) calls to an i.i.d. active learner, noting that it is likely to find a useful stationary policy well before all \(T\) calls are issued. RAIL is an idealized algorithm intended for analysis, which achieves the theoretical goals but has a number of inefficiencies from a practical perspective. Later in Section 5 we describe the practical instantiation that

\(^1\) The main result of (Syed and Schapire, 2010) holds for stochastic expert policies and requires a more complicated analysis that results in a looser bound. Lemma 3 is strong enough for deterministic expert policies, which is the assumption made in our work.