1 Counterexample-Guided Inductive Learning

Given a specification, we build up a counterexample-guided inductive learning system to exploit information from the specification. The diagram of the system can be seen in Fig.**??**.

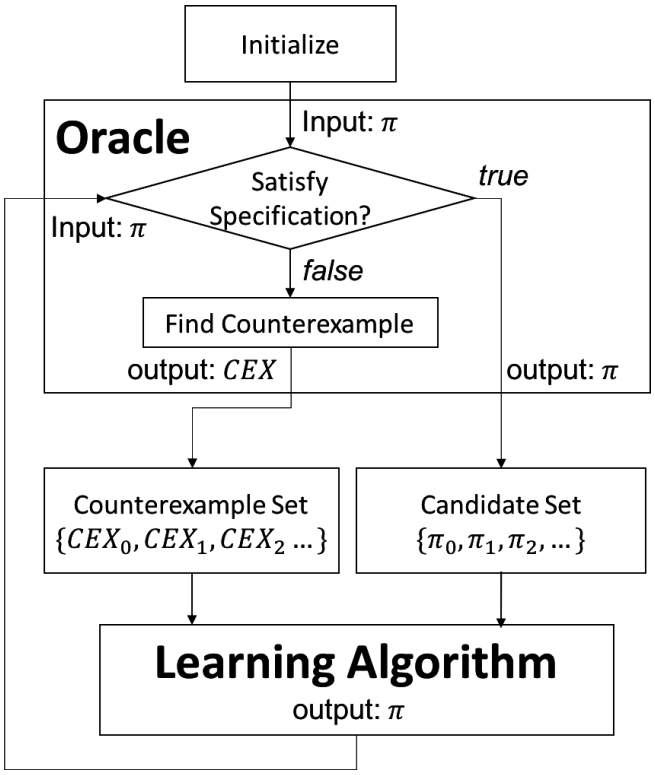

Fig. 1. Diagram of our counterexample-guided Inductive learning framework.

The system contains four major components: learning algorithm, a verification oracle, a set of candidate output and a set of counterexamples. The learning algorithm shall both learn from expert and from counterexamples provided by the verification oracle. Starting from the initial randomly generated policy, every time the learning algorithm learns a new policy, the policy shall be sent to oracle to verify if satisfying the specification. If true, then this policy shall be added to the candidate output set, otherwise, the oracle shall generate a minimal counterexample, which will be added to the counterexample set. Learning algorithm use the newly updated either counterexample set or candidate output set to learn the next policy. The verification and minimal counterexample generation shall be done by probabilistic model checking technique.

Considering the MDP M and safety specification (**??**) for the grid world in section 3, if a policy π doesn't satisfy ϕ, a counterexample CEX shall be a set of $|CEX|$ trajectories $\{\tau_0, \tau_1, ..., \tau_{|CEX|-1}\}$ which all end in some 'unsafe' state. Assuming that the initial state of a trajectory τ_i is $\tau_i(s^{(0)})$, then a counterexample CEX should satisfies that $\sum_{\tau_i \in CEX} P(\tau_i) D(\tau_i(s^{(0)})) \geq p^*$ and the minimal counterexample CEX_π is the one that minimizes the left side of the inequality. The normalized expected feature counts of CEX_π is:

$$\mu_{CEX_\pi} = \frac{\sum_{\tau_i \in CEX_\pi} P(\tau_i) \sum_{s^{(t)} \in \tau_i} \gamma^t f(s^{(t)})}{\sum_{\tau_i \in CEX_\pi} P(\tau_i)} \tag{1}$$

As any $\tau \in CEX_\pi$ should be shorter than the maximal step length t=64, we regard the last state, which must be an 'unsafe' state, to be an absorbing, but only when calculating $\sum_{s^{(t)} \in \tau_i} \gamma^t f(s^{(t)})$. This amplifies the significance of features of the 'unsafe' state in μ_{CEX_π}.

To learn from counterexample CEX_π, we resemble the way of learning from expert. Yet, instead of trying to maximize the distance between expert feature counts μ_E and convex combinations of candidate feature counts $\Pi = \{\mu_0, \mu_1, ...\}$, we want to maximize the difference between the counterexample feature counts μ_{CEX_π} and all candidate feature counts as shown in Fig.(**??**). The optimization function should be:

$$\delta = max_\omega min_{\mu_i \in \Pi} \ \omega^T (\mu_i - \mu_{CEX_\pi}) \tag{2}$$

$$s.t. \quad ||\omega||_2 \leq 1 \tag{3}$$

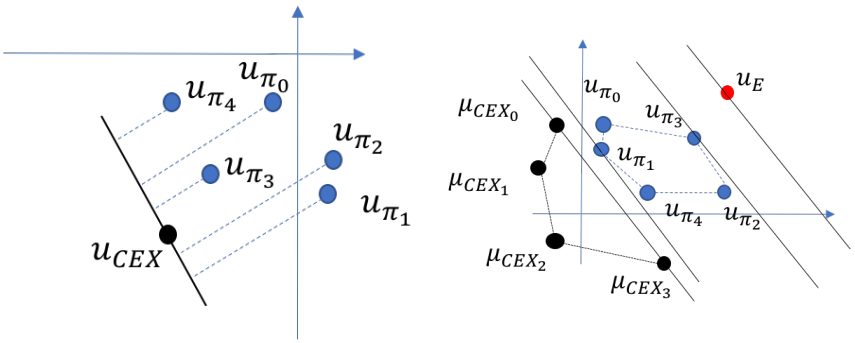

Fig. 2. Inductively learn from both expert and counterexample.

When multiple counterexamples have been found, we define Π_{CEX} as the set of expected feature counts of the counterexample trajectory sets, $\Pi_{CEX} = \{\mu_{CEX_{\pi_0}}, \mu_{CEX_{\pi_1}}, \mu_{CEX_{\pi_2}}, ...\}$. We want to maximize the distance between two

convex hull of Π_{CEX} and Π, which is equivalent to maximize the distance between the parallel support hyperplanes of Π_{CEX} and Π. Then the optimization function becomes:

$$\delta = max_{\omega} min_{\mu_i \in \Pi, \mu_{CEX_{\pi_j}} \in \Pi_{CEX}} \ \omega^T (\mu_i - \mu_{CEX_{\pi_j}}) \tag{4}$$

$$s.t. \quad ||\omega||_2 \leq 1 \tag{5}$$

Meanwhile, we still want to learn form μ_E. Then it becomes a multi-objective optimization problem that combines two optimization function (**??**) and (**??**):

$$max_{\omega} min_{\mu \in \Pi, \mu_{CEX} \in \Pi_{CEX}} (\omega^T (\mu_E - \mu), \omega^T (\mu - \mu_{CEX})) \tag{6}$$

$$s.t. \quad ||\omega||_2 \leq 1 \tag{7}$$