Notes on Why Behavior-cloning Fails

Zhihan Yang
Carleton College
yangz2@carleton.edu

May 2020

Contents

1 Quantifying the goodness of behavior-cloning using the zero-one loss

The zero-one loss is defined as follows:

$$c(\mathbf{s}, \mathbf{a}) = \begin{cases} 0 \text{ if } \pi^\star(\mathbf{s}) = \mathbf{a} \\ 1 \text{ otherwise.} \end{cases}$$

(review)
where $\pi^\star$ is the policy provided by the expert. In plain English, the loss of taking action $\mathbf{a}$ in state $\mathbf{s}$ is 0 if $\mathbf{a}$ is the action that the expert would take in state $\mathbf{s}$ and 1 if otherwise.

We then seek a bound on the expected loss on the following "tight-rope" scenario (Figure 1). At each state (except the 3 ending states), there are 3 possible next states to choose from. The sequences of states that the expert has traversed is marked in green. The goal of the behavior-cloning agent is to traverse the expert' path as close as possible.

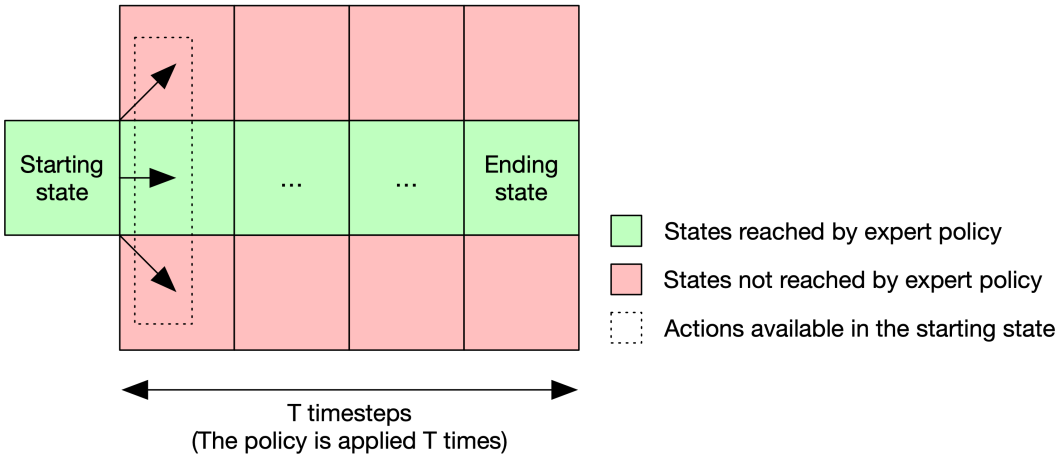

Figure 1: The 'tight-rope" scenario.

Supervised learning on expert's policy. Suppose that an arbitrary path in the tight-rope scenario would involve applying an arbitrary policy T times. Then obviously the expert's path is of length T. At each time-step, we treat the expert's state

as an training example and the experts' action as the corresponding training label. This one-to-one mapping can be seen as the expert's *policy* - how it reacts (takes actions') to various (but not all!) situations (states). Aggregating such pairs over all time-steps, we obtain a training set of $T-1$ labeled examples. Finally, we use supervised learning models such as a deep neural network to fit this data. Note that this model then becomes a policy on its own and we denote it by $\pi_\theta(\mathbf{s})$ where θ parameterizes the model. Importantly, we assume that $\pi_\theta(a \neq \pi^\star(\mathbf{s}) \mid \mathbf{s}) \leq \epsilon$. Note that the entire purpose of ϵ is to quantify the strength of the supervised learner.

Expected cost of $\pi_\theta(\mathbf{s})$ over T time-step. The expected cost can be computed intuitively as follows:

$$\mathbb{E}\left[\sum_t c(\mathbf{s}_t, \mathbf{a}_t)\right] \leq \underbrace{\epsilon T}_{\text{Mistake at } T=1} + \underbrace{(1-\epsilon)(\underbrace{\epsilon(T-1)}_{\text{Mistake at } T=2} + \underbrace{(1-\epsilon)(\epsilon(T-2)+\cdots)}_{\text{Correct at } T=2})}_{\text{Correct at } T=1},$$

assuming that once we make a mistake, we will always be making mistakes since we will be dealing with unseen states. This is, of course, unrealistic in this scenario where there are only 3 actions to choose from (corresponding to the 3 available) states), but this is reasonable as a bound, especially when the state space is large.

Complexity of expected cost.

- T terms because we are making decisions T times (each time we make a decision we create a mistake term)
- Each term has $O(\epsilon T)$
- total this is $O(\epsilon T^2)$

Such a complexity is terrible.

2 More general analysis

Assume $p(\pi^\star(\mathbf{s}) \neq \pi_\theta(\mathbf{s})|\mathbf{s}) \leq \epsilon$. In essence, this assumption assumes that the training of a deep neural network to work decently but makes a small mistake.

2.1 Case 1: $p_{\textbf{train}}(\mathbf{s}) \rightarrow p_\theta(\mathbf{s})$

Then at every step, whether we make a mistake, we always land in a state that is available in the training set, so we know how to react. Mention the dagger algorithm.

2.2 Case 2: $p_{\textbf{train}}(\mathbf{s}) \neq p_\theta(\mathbf{s})$

$$p_\theta(\mathbf{s}_t) = \underbrace{(1-\epsilon)^t}_{0 \text{ mistake}} p_{\text{train}}(\mathbf{s}_t) + \underbrace{(1-(1-\epsilon)^t)}_{\geq 1 \text{ mistakes}} p_{\text{mistake}}(\mathbf{s}_t)$$

- no mistake wrt to the expert, in this case, still in the training distribution
- made at least one mistake, not sure where it is now; note that since p train and p mistake are probability distributions so they might overlap, we can't really be sure about this (how well the neural net generalize)

We can use the equation above to prove that (this will help us later in proving the bound):

$$\sum_{\mathbf{s}_t} |p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| = \left(1-(1-\epsilon)^t\right) \sum_{\mathbf{s}_t} |p_{\text{mistake}}(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)|$$

where the absolute value operator denotes the *total variation divergence* between two distributions (review).
because

$$\begin{aligned}
&\sum_{\mathbf{s}_t} |p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| \\
=&\sum_{\mathbf{s}_t} \left|(1-\epsilon)^t p_{\text{train}}(\mathbf{s}_t) + (1-(1-\epsilon)^t) p_{\text{mistake}}(\mathbf{s}_t) - (1-\epsilon)^t p_{\text{train}}(\mathbf{s}_t) - (1-(1-\epsilon)^t) p_{\text{train}}(\mathbf{s}_t)\right|
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\mathbf{s}_t} \left| \underbrace{(1-(1-\epsilon)^t)}_{\text{Positive}} p_{\text{mistake}}(\mathbf{s}_t) - (1-(1-\epsilon)^t) p_{\text{train}}(\mathbf{s}_t) \right| \\
&= (1-(1-\epsilon)^t) \underbrace{\sum_{\mathbf{s}_t} |p_{\text{mistake}}(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)|}_{\leq 2} \\
&\leq 2(1-(1-\epsilon)^t) \\
&\leq 2\epsilon t && \text{Using the inequality that } (1-\epsilon)^t \geq 1-\epsilon t \text{ for } \epsilon \in [0,1]
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_{\tau \sim p(\tau)} \left[\sum_t c_t(\mathbf{s}_t, \mathbf{a}_t) \right] &= \sum_t \mathbb{E}_{\mathbf{s}_t \sim p_\theta(\mathbf{s}_t)} \left[c_t(\mathbf{s}_t, \mathbf{a}_t) \right] && \text{By the linearity of expectations ...} \\
&= \sum_t \sum_{\mathbf{s}_t} p_\theta(\mathbf{s}_t) c_t(\mathbf{s}_t, \mathbf{a}_t) \\
&= \sum_t \sum_{\mathbf{s}_t} \{p_{\text{train}}(\mathbf{s}_t) + p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)\} c_t(\mathbf{s}_t, \mathbf{a}_t) \\
&= \sum_t \sum_{\mathbf{s}_t} p_{\text{train}}(\mathbf{s}_t) c_t(\mathbf{s}_t, \mathbf{a}_t) + \{p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)\} c_t(\mathbf{s}_t, \mathbf{a}_t) \\
&\leq \sum_t \sum_{\mathbf{s}_t} p_{\text{train}}(\mathbf{s}_t) c_t(\mathbf{s}_t, \mathbf{a}_t) + |p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| \, c_t(\mathbf{s}_t, \mathbf{a}_t) \\
&\leq \sum_t \sum_{\mathbf{s}_t} p_{\text{train}}(\mathbf{s}_t) c_t(\mathbf{s}_t, \mathbf{a}_t) + |p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| \, c_{\max} \\
&\leq \sum_t \sum_{\mathbf{s}_t} p_{\text{train}}(\mathbf{s}_t) c_t(\mathbf{s}_t, \mathbf{a}_t) + |p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| && c_{\max} = 1 \text{ because we are using the zero-one loss} \\
&= \sum_t \epsilon + 2\epsilon t
\end{aligned}
$$

where $\tau = \{\mathbf{s}_1, \mathbf{a}_1, \cdots, \mathbf{s}_t, \mathbf{a}_t\}$ is a sampled trajectory

3 Questions

- What does it mean to subtract two distributions