**Truncated Horizon Policy Search: Combining Reinforcement Learning and Imitation Learning**

Wen Sun, Drew Bagnell, and Byron Boots

\{wensun, dbagnell\}@cs.cmu.edu, bboots@cs.gatech.edu

---

**Problem Setup: Reinforcement Learning**

Sequential Decision Making

\[ \pi(s) \rightarrow a \]

Minimize Discounted Expected Total Cost

\[ J(\pi) = \mathbb{E}[c_1 + \gamma c_2 + \gamma^2 c_3 + \ldots | a \sim \pi(\cdot|s)] \]

**Extra Source of Help: Imitation**

Some Expert’s Cost-to-go Oracle: \( \hat{V}^e(s) \)

- Learned from Expert’s demonstration (e.g., TD)
- Prior knowledge of the task [Reward Shaping, Ng, 99]
- From imperfect model (e.g., learned model)

**Challenge**: How we can leverage such an imperfect oracle to speed up learning, if possible?

**Previous Pure Imitation Learning Works**

AggreVaTe & AggreVaTeD

[Ross & Bagnell, 14; Sun et al, ICML, 17]

\[ \hat{\pi}(s) = \arg\min_a c(s, a) + \gamma \mathbb{E}[s' \sim P(c(s, a))\hat{V}^e(s')] \]

They can learn near-optimal policy when the oracle provides unbiased estimate of the optimal policy’s cost-to-go, i.e.,

\[ \hat{V}^e(s) = V^*(s) \]

**Cost (Reward) Shaping**

\[ c'(s, a) = c(s, a) + \gamma \mathbb{E}_{s' \sim P(c(s, a))} \Phi(s') - \Phi(s) \]

New MDP with \( c' \) shares the same optimal policy as the original MDP [Ng, 99]

\[ \Phi(s) = V^*(s) \Rightarrow \pi^*(s) = \arg\min c'(s, a) \]

Optimal Cost-to-go => a new one-step greedy MDP

AggreVaTe & AggreVaTeD is solving one-step Greedy MDP, which explains why it’s faster than RL

**Oracle Accuracy VS Planning Horizon**

**Core idea**: Imitation Learning via Cost Shaping with \( \hat{V}^e(s) \)

\[ \hat{V}^e(s) - V^*(s) \approx \epsilon \text{ in } \mathbb{R}^+ \]

- AggreVaTe can fail with imperfect oracle:
  \[ |\hat{V}^e(s) - V^*(s)| \approx \epsilon \text{ in } \mathbb{R}^+ \]

**Core idea**: Be less greedy and do Multi-step look ahead

\[ \hat{\pi}(s) = \arg\min_a [\mathbb{E} \sum_{i=1}^N \gamma^{i-1} c'(s_i, a_i) | s_1 = s, a \sim \hat{\pi}] \forall s. \]

\[ c(s_1, a_1) + \gamma c(s_2, a_2) - \hat{V}(s_4) = c'(s_1, a_1) + \gamma c'(s_2, a_2) \]

Optimizing the tree policy = Optimizing a reshaped MDP with 2 steps

Find a policy that optimizes K steps of the reshaped MDP (cost function \( c' \))

\[ J(\hat{\pi}) - J(\pi^*) \leq O\left( \frac{\gamma^k \epsilon}{1 - \gamma} \right) \]

**Experiments**

1. Learned \( \hat{V}^e \) from a set of expert demonstrations using TD.
2. Use Actor-Critic (TRPO-GAE [schulman et al., 16]), where critic only estimates k-step Q.

**Sparse Reward Setting**

For large state space problems, the oracle learned from a set of demonstrations is inaccurate. We need expert during the training loop to improve the critic.