Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

Summary of Lecture 22: Inverse Reinforcement Learning

Summarized By: Amirhossein Asadi

Inverse Reinforcement Learning

- Modeling human behavior using optimal control assumes access to optimal trajectories and a well-defined cost function. However, in practice, human actions are often suboptimal and the true objective is hard to specify, limiting the applicability of this approach.
- **RL vs. IRL:** In RL, the reward function is known and denoted as $r_\psi(s,a)$, and the objective is to learn the optimal policy π^*. In contrast, IRL assumes access to samples from π^* and aims to recover the underlying reward function $r_\psi(s,a)$ that explains the observed behavior.
- In heuristic IRL algorithms, it is commonly assumed that the reward function is a linear combination of features. This linearity facilitates generalization and simplifies the learning process.

$$r_\psi(s,a) = \psi^\top \phi(s,a)$$

- **Feature Matching IRL:** As a first step in feature-based IRL, we select the parameter vector ψ such that the expected feature counts under the learned policy π^{r_ψ} match those under the expert policy π^*:

$$\mathbb{E}_{\pi^{r_\psi}}[\mathbf{f}(s,a)] = \mathbb{E}_{\pi^*}[\mathbf{f}(s,a)]$$

  This principle serves as the foundation for algorithms like Apprenticeship Learning, where the goal is to find a reward function for which the optimal policy replicates expert behavior in terms of feature usage.
- To address the under-specification problem in IRL, one can adopt the maximum **margin principle**. The idea is to find a reward parameter ψ and margin m such that the expert policy is not only optimal but also yields a significantly higher expected reward than any alternative policy. This can be formulated as:

$$\max_{\psi,m} \; m \quad \text{s. t.} \quad \psi^\top \mathbb{E}_{\pi^*}[\phi(s,a)] \geq \max_{\pi \in \Pi} \; \psi^\top \mathbb{E}_{\pi}[\phi(s,a)] + m$$

  This constraint ensures that the reward function distinguishes the expert policy from suboptimal ones by a margin m, enhancing the robustness of reward inference.
- To avoid hard constraints in maximum margin IRL, we reformulate the optimization using a divergence-based regularization:

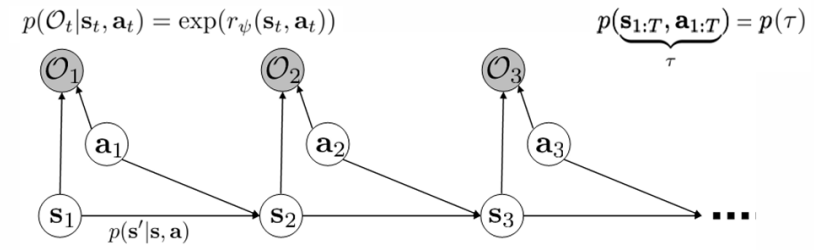
$$\min_{\psi} \; \frac{1}{2}\|\psi\|^2 \quad \text{s. t.} \quad \psi^\top \mathbb{E}_{\pi^*}[\phi(s,a)] \geq \max_{\pi \in \Pi} \left(\psi^\top \mathbb{E}_{\pi}[\phi(s,a)] + D(\pi,\pi^*)\right)$$

  This formulation ensures that the expert policy π^* achieves higher expected reward than any other policy by a margin proportional to their divergence $D(\pi,\pi^*)$. Typical choices for D include KL-divergence.
- The constraint can be absorbed into the objective using Lagrangian relaxation, yielding an unconstrained problem:

$$\mathcal{L}(\psi,\lambda) = \frac{1}{2}\|\psi\|^2 + \lambda \left(\max_{\pi \in \Pi} \left[\psi^\top \mathbb{E}_{\pi}[\phi(s,a)] + D(\pi,\pi^*)\right] - \psi^\top \mathbb{E}_{\pi^*}[\phi(s,a)] \right)$$

This approach enables gradient-based optimization while still favoring reward functions that distinguish expert behavior.

- **Regularized IRL Still DOESN'T WORK:** Even with divergence penalties and Lagrangian relaxation, the method requires solving hard constrained optimizations and estimating policy maxima, which remains intractable. This approach named **Apprenticeship Learning**.
- Generative models in RL can produce plausible trajectories but lack a sense of optimality. To address this, we introduce an auxiliary variable representing the **optimal value**, guiding the model to generate trajectories aligned with optimal actions. This approach enhances the learning of reward functions from high-quality expert data.



this optimality is local or per-step, not global over trajectories.

- $$p(\tau \mid \mathcal{O}_{1:T}) = \frac{p(\tau, \mathcal{O}_{1:T})}{p(\mathcal{O}_{1:T})} \propto p(\tau) \prod_t \exp(r(s_t, a_t)) = p(\tau) \exp\left(\sum_t r(s_t, a_t)\right)$$

  This means that the higher the sum of rewards along a trajectory, the exponentially more likely that trajectory becomes.

- In IRL, we maximize the likelihood of expert trajectories by computing the following expressions:

$$p(\tau \mid \mathcal{O}_{1:T}, \psi) \propto \exp\left(\sum_t r_\psi(s_t, a_t)\right)$$

$$Z = \int p(\tau) \exp(r_\psi(\tau))\, d\tau$$

$$\mathcal{L}(\psi) = \frac{1}{N} \sum_{i=1}^{N} r_\psi(\tau_i) - \log Z$$

$$\nabla_\psi \mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \nabla_\psi r_\psi(\tau_i) - \frac{1}{Z} \int p(\tau) \exp(r_\psi(\tau)) \nabla_\psi r_\psi(\tau)\, d\tau$$

$$\nabla_\psi \mathcal{L} = \mathbb{E}_{\tau \sim \pi^*(\tau)} \left[\nabla_\psi r_\psi(\tau)\right] - \mathbb{E}_{\tau \sim p(\tau|\mathcal{O}_{1:T}, \psi)} \left[\nabla_\psi r_\psi(\tau)\right]$$

Higher-reward trajectories are exponentially more likely, and learning optimizes the reward parameters to match the expert's expected reward features.