Approximate Robust Inverse Reinforcement Learning

Brendan Crowe

Reinforcement Learning

- **Reinforcement learning (RL)** is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. [2]

Reinforcement Learning

- **Reinforcement learning (RL)** is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. [2]

- The optimizer knows that reward function r

Reinforcement Learning

- **Reinforcement learning (RL)** is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. [2]

- The optimizer knows that reward function r

- The optimizer wants to find the policy $\pi^\star \in \Pi$

Reinforcement Learning

- **Reinforcement learning (RL)** is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward. [2]

- The optimizer knows that reward function *r*

- The optimizer wants to find the policy $\pi^\star \in \Pi$

- Example: AlphaGO

Inverse Reinforcement Learning

- **Inverse Reinforcement Learning (IRL):** Does not assume that the reward function *r* is known, instead the goal is to find a reward function that best explains some expert data.

Inverse Reinforcement Learning

- **Inverse Reinforcement Learning (IRL):** Does not assume that the reward function r is known, instead the goal is to find a reward function that best explains some expert data.
- The optimizer has data $\mathcal{D}$ that shows the tasks being done (hopefully) optimally

Inverse Reinforcement Learning

- **Inverse Reinforcement Learning (IRL):** Does not assume that the reward function r is known, instead the goal is to find a reward function that best explains some expert data.
- The optimizer has data $\mathcal{D}$ that shows the tasks being done (hopefully) optimally
- The optimizer wants to find a reward function $r^\star \in R$ that best explains $\mathcal{D}$ and produces a good policy $\pi^\star$

Inverse Reinforcement Learning

- **Inverse Reinforcement Learning (IRL):** Does not assume that the reward function r is known, instead the goal is to find a reward function that best explains some expert data.
- The optimizer has data $\mathcal{D}$ that shows the tasks being done (hopefully) optimally
- The optimizer wants to find a reward function $r^\star \in R$ that best explains $\mathcal{D}$ and produces a good policy $\pi^\star$
- Example: Teaching a self driving car using data collected from people driving in the real world

The Problem

- Infinite number of reward functions that fit the data
- The goal is to be able to find a policy that does well for the worst-case reward function
- We can deal with this uncertainty using robust optimization

The optimization

$$
\begin{aligned}
\max_u \min_r \quad & r^T u \\
s.t. \quad & A^T u = c \qquad\qquad (1)\\
& u \geq 0 \qquad [1]
\end{aligned}
$$

The optimization

$$\begin{aligned} \max_{u,z} \quad & z \\ s.t. \quad & A^T u = c \\ & z \le r^T u \quad r \in R \\ & u \ge 0 \qquad [1] \end{aligned} \tag{2}$$

- R: a set of possible reward functions
- u: the occupancy for each state-action pair
- $A = \begin{bmatrix} I - \gamma P_a \\ \vdots \end{bmatrix}$
- P_a: probability of transitioning from one state to the next given an action
- $0 \le \gamma \le 1$: discount rate (a constant)
- c: the distribution over starting states-actions

Limitations

- This requires all states to be known
- Not feasible for large problems
- Not flexible

Approximation

- The solution: approximation
- Use linear features Φ that approximate the states
- Generalizes to states we haven't seen

The primal

$$
\begin{aligned}
\max_{u,z} \quad & z \\
s.t. \quad & \Phi A^T u = \Phi c \\
& z \leq r^T u \quad r \in R \\
& u \geq 0
\end{aligned} \tag{3}
$$

What's going on??

The dual

$$\begin{aligned} \min_{w,\xi} \quad & c^T \Phi w \\ s.t. \quad & \Phi A^T w \geq \hat{\Phi} \tilde{R} \xi \\ & \mathbb{1}^T \xi = 1 \\ & \xi \geq 0 \end{aligned} \tag{4}$$

- $\Phi w \approx v$: represents the value of being at each state
- $\Phi \in \mathbb{R}^{n \times m}$: a features matrix that is number of observed states $\times$ number of features
- $\hat{\Phi} = \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix}$
- $\tilde{R}$: weights that describe inform a reward function $\hat{\Phi}\tilde{R} \approx R$
- ξ: a weight for each possible reward function in $\hat{\Phi}\tilde{R}$

Citations

[1] Daniel S. Brown, Scott Niekum **and** Marek Petrik. *Bayesian Robust Optimization for Imitation Learning*. 2020. arXiv: 2007.12315 [cs.LG].

[2] Wikipedia contributors. *Reinforcement learning — Wikipedia, The Free Encyclopedia*. [Online; accessed 3-May-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Reinforcement_learning&oldid=1019990151.