CS294-40 Learning for Robotics and Control Lecture 22 - 11/18/2008

Inverse Reinforcement Learning

Lecturer: Pieter Abbeel *Scribe: Ankur Mehta*

1 Inverse Reinforcement Learning

Given a policy π_E or state-action traces $\{s_t, a_t\}$, can we recover the reward function R? Assume the dynamics model T is known.

1.1 Why?

- Build a computational model for human / animal behavior (e.g. are they minimizing energy? Torques?)
- Design control policies (often hard in practice to specify R)
- Model opponents in multi-agent systems
- Imitation learning – "Behavioral cloning"

Special case (Kalman): Given a linear-quadratic system, given the feedback matrix, can we extract the cost matrix?

General case...

1.2 Formal problem statement

Find R s.t.:

$$E\left[\sum_t \gamma^t R(s_t)|\pi_E\right] \geq E\left[\sum_t \gamma^t R(s_t)|\pi\right] \quad \forall\pi. \tag{1}$$

Let's define

$$\lambda^\pi(s) = E\left[\sum_t \gamma^t \mathbf{1}\{s_t = s\}|\pi\right] \tag{2}$$

is the discounted number of times the policy π hits state s. Then, the problem is equivalent to finding R s.t.:

$$\sum_s \lambda^{\pi_E}(s)R(s) \geq \sum_s \lambda^\pi(s)R(s) \quad \forall\pi. \tag{3}$$

This is now a linear system of inequalities in R, which is efficient to solve using a linear program if we know all π. This permits a solution $R = 0$, so to avoid that we need to introduce additional assumptions.

1.3 Take 1

Find R s.t. the expert policy is better than any other policy by a constant margin less some slack:

$$\min_{R,\xi} \sum \xi(\pi) : \sum_s \lambda^{\pi_E}(s)R(s) > \sum_s \lambda^\pi(s)R(s) + 1 - \xi(\pi) \quad \forall\pi \neq \pi_E. \tag{4}$$

Problem: may not find a solution if there are policies very similar to the expert policy

1.4 Take 2

Find R s.t. the expert policy is better than any other policy by a scaled margin less some slack:

$$\min_{R,\xi} \sum \xi(\pi) : \sum_s \lambda^{\pi_E}(s)R(s) > \sum_s \lambda^{\pi}(s)R(s) + m(\pi) - \xi(\pi) \quad \forall \pi \neq \pi_E.$$

where $m(\pi)$ is larger for π more "different" from π_E.

Still has some impracticalities:

- λ^{π_E} is only known from samples, so need to use estimate $\hat{\lambda}^{\pi_E}$ derived from data.
- $\{\pi\}$ is very large, of size $|A|^{|S|}$.
  - Sample a subset of constraints
  - Generate constraints dynamically
- $R \in \mathbb{R}^{|S|}$ is very rich, so we will end up overfitting to our approximate $\hat{\lambda}^{\pi_E}$. Instead, use features:

$$R(s) = \sum_i w_i \phi_i(s) \tag{5}$$

- Runtime O(number of variables in LP), so make ξ a scalar, constant over π.

1.5 Take 3

1.5.1 Fixed constraint sample set

Use a subset of policies $\Pi \subset \{\pi\}$. Then find R s.t.:

$$\min_{R,w,\xi} \xi + c||w|| : \begin{cases} R(s) = \sum_i w_i \phi_i(s) & \forall s \\ \sum_s \lambda^{\pi_E}(s)R(s) > \sum_s \lambda^{\pi}(s)R(s) + m(\pi) - \xi & \forall \pi \in \Pi. \end{cases} \tag{6}$$

This is a convex optimization problem that can be solved with linear programming if minimizing the 1- or ∞- norm of w.

1.5.2 Constraint generation

- Initialize $\Pi_0 = \{\}$.
- for i in 0,1,2,...
  - Solve (6) for $\Pi = \Pi_i$ to get $R^{(i)}, w^{(i)}, \xi^{(i)}$
  - Find which policy $\pi^{(i)} \in \pi$ violates (3) the most:
    - $\pi^{(i)}$ is almost the optimal policy w.r.t. reward function $R^{(i)}$
    - $\pi^{(i)} = \arg\max_\pi E\left[\sum_t \gamma^t R(s_t)|\pi\right] + m(\pi)$
  - If $\pi^{(i)}$ satisfies (3), then $R = R^{(i)}$; exit.
  - Otherwise, $\Pi_{i+1} = \Pi_i \cup \{\pi^{(i)}\}$.