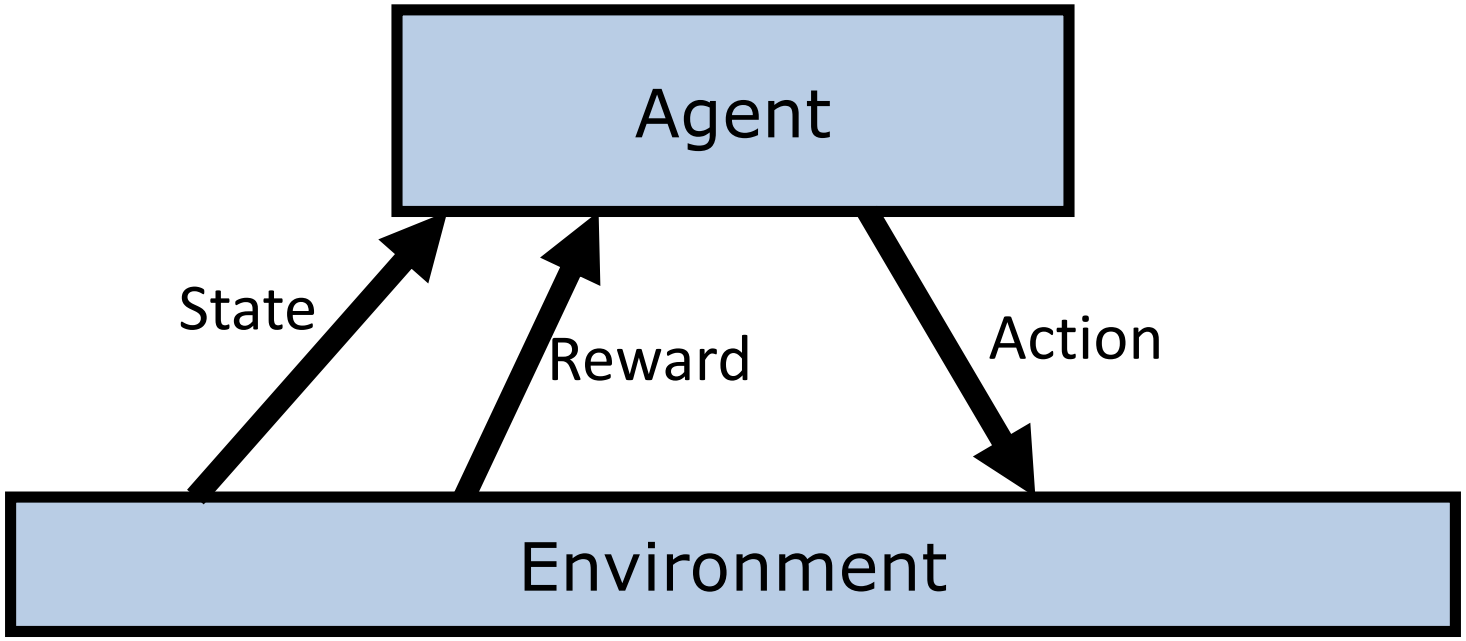
CS885 Reinforcement Learning Lecture 17c: June 27, 2018

Inverse Reinforcement Learning

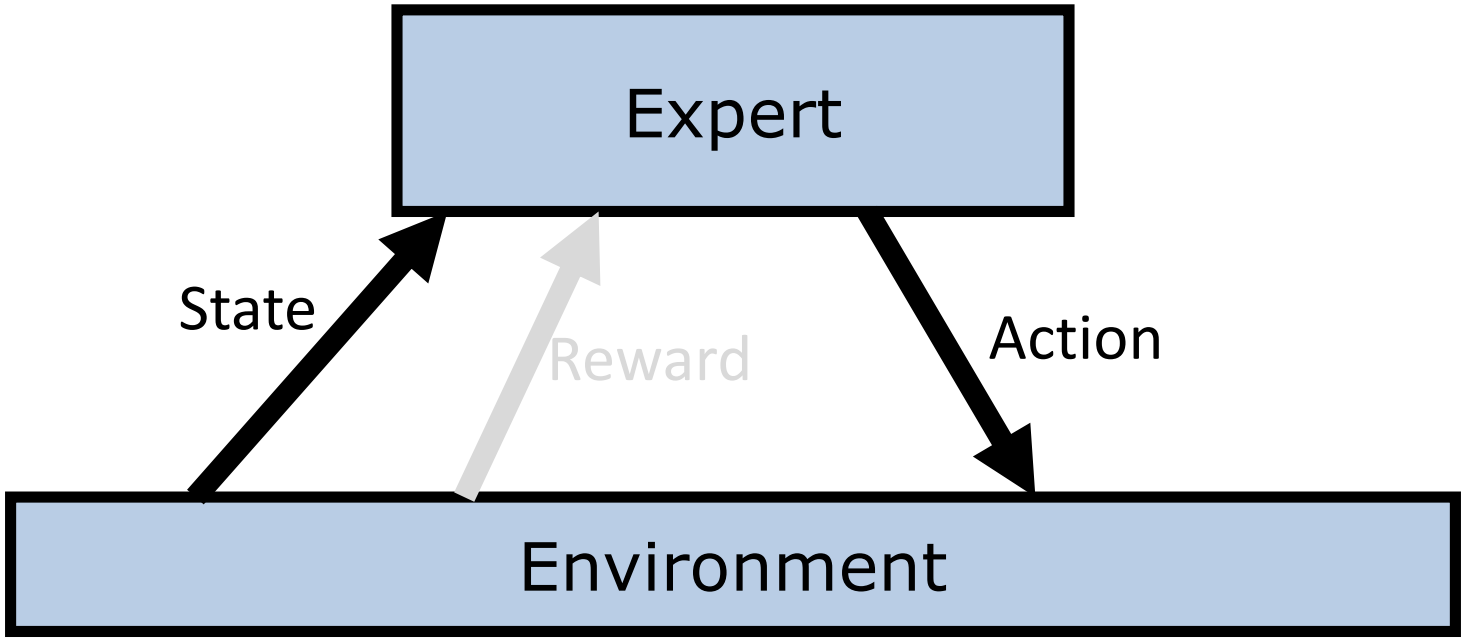
[Abbeel, Ng, ICML-2004]

Reinforcement Learning



Goal: Find policy π that maximize rewards

Inverse Reinforcement Learning



Goal: Find reward function that expert is implicitly optimizing

Some Applications

- Robotics: train controller to follow demo trajectories
  - Quadruped locomotion (Ratliff et al. NIPS-07; Kolter et al. NIPS-08)
  - Helicopter Aerobatics (Abbeel et al. IJRR-10)

- Autonomous driving: train by driver demonstration
  - Highway driving (Abbeel, Ng; ICML-04; Syed, Schapire NIPS-07)
  - Parking lot navigation (Abbeel et al. IROS-08)
  - Urban navigation (Ziebart et al. AAAI-08)
  - Driving styles (Kuderer et al. ICRA-15)

MDP without R

- MDP\R
  - Set of states: S
  - Set of actions: A
  - Transition model: $\Pr(s_t|s_{t-1}, a_{t-1})$
  - Reward model: $R(s_t, a_t)$
  - Discount factor: $0 \leq \gamma \leq 1$
  - Horizon (i.e., # of time steps): h
  - Optimal policy: $\pi^*: S \rightarrow A$

- Goal: find reward model $R(s, a)$ optimized by π^*

Transfer Learning

- Since we have π^*, why find R?

- Consider a route planning problem (several possible objectives: time, distance, energy consumption)
  - Suppose we know the route (policy) followed by taxi drivers between any pair of locations in city A
  - An autonomous car can follow the same policy in city A
  - What about city B?
- If we find the objective optimized by taxi drivers in city A, then we can use that objective in city B to find routes with similar properties

Finding underlying R

- Let $V_R^{\pi}(s) = \sum_{t=0}^{h} \gamma^t R\big(s_t, \pi(s_t)\big)$

- Idea: find R that satisfies

$$V_R^{\pi^*}(s) \geq V_R^{\pi}(s)\ \forall s, \pi$$

- Problems:
  - Many R's satisfy the constraints (e.g., $R(s,a) = 0\ \forall s, a$)
  - Enumerating all policies in impractical
  - Optimal policy and transition model may be unknown

Inverse RL

- Problem
  - Set of states: S
  - Set of actions: A
  - Transition model: $\Pr(s_t|s_{t-1}, a_{t-1})$
  - Reward model: $R(s_t, a_t)$
  - Discount factor: $0 \leq \gamma \leq 1$
  - Horizon (i.e., # of time steps): h
  - Expert trajectories: $\{(s_1^1, a_1^1, s_2^1, a_2^1, \dots, s_h^1, a_h^1), (s_1^2, a_1^2, s_2^2, a_2^2, \dots, s_h^2, a_h^2), \dots\}$

- Goal: find reward model $R(s, a)$ that expert trajectories optimize

Transfer Learning

- Since we have expert trajectories, why not find a policy by supervised learning instead of find R?

- The resulting policy will only work in the current environment
- If we find R, then we can optimize policies by RL in new environments with different transition models.

Finding R

- Suppose R depends on features ϕ_i
- $R(s,a) = \sum_i w_i \phi_i(s,a)$ or $neuralNet(\phi(s,a); w)$

- Idea: alternate between
  - Updating R (by revising w)
  - Optimizing π based on R

Apprenticeship Learning

ApprenticeshipLearning($trajectories$)

i indexes features, j indexes trajectories, k indexes policies

$k \leftarrow 0$

Initialize π_k at random

Repeat

$$margin = \max_w \min_{j,k} \sum_{i,t} w_i \phi_i(s_t^j, a_t^j) - \sum_{i,t} w_i \phi_i\left(s_t^j, \pi_k(s_t^j)\right)$$

where w^* is the maximizing w

$k \leftarrow k + 1$

$\pi_k \leftarrow$ optimal policy for w^* found by RL

Until $margin$ is small enough

Return π_k