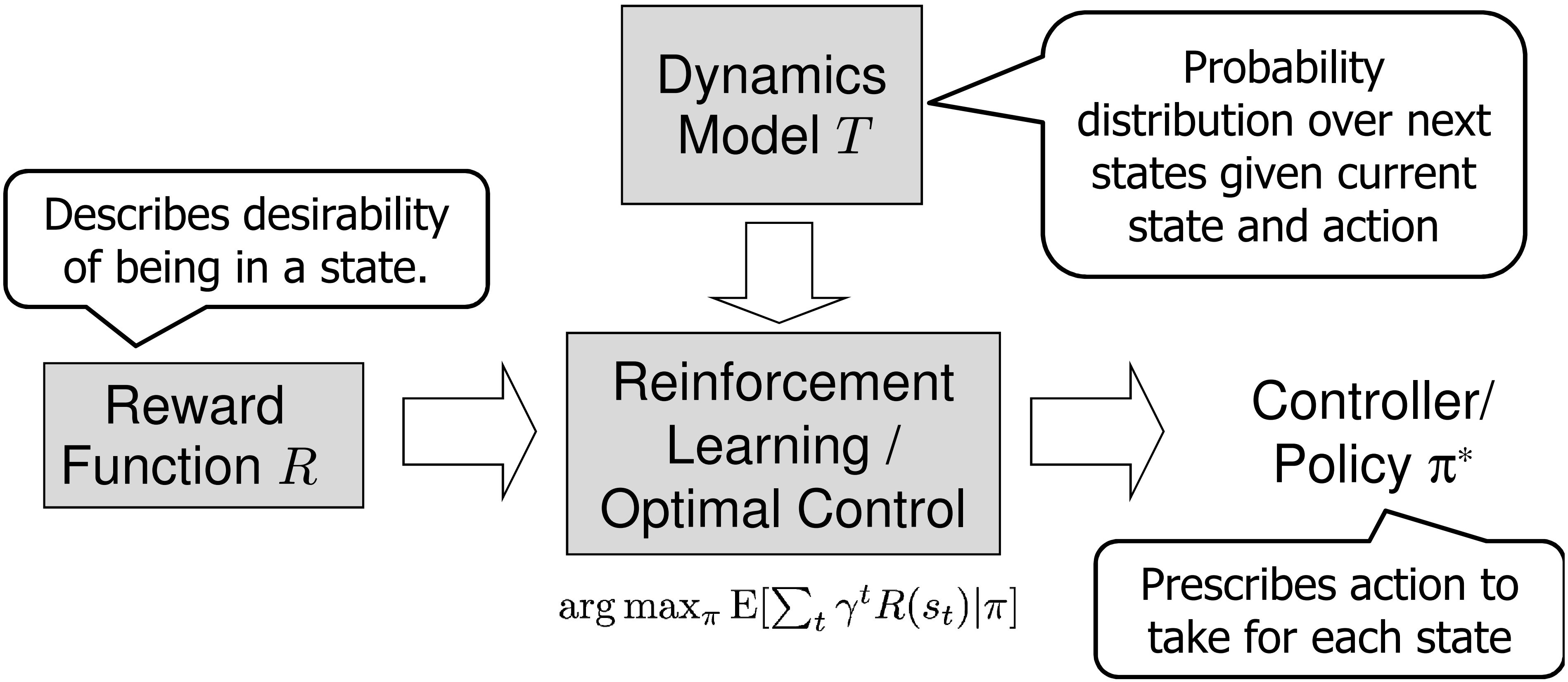
High-level picture



Inverse RL:

Given π^* and T, can we recover R?

More generally, given execution traces, can we recover R?

Motivation for inverse RL

- Scientific inquiry
  - Model animal and human behavior
    - E.g., bee foraging, songbird vocalization. [See intro of Ng and Russell, 2000 for a brief overview.]
- Apprenticeship learning/Imitation learning through inverse RL
  - Presupposition: reward function provides the most succinct and transferable definition of the task
  - Has enabled advancing the state of the art in various robotic domains
- Modeling of other agents, both adversarial and cooperative

Problem setup

- Input:
  - State space, action space
  - Transition model $P_{sa}(s_{t+1} \mid s_t, a_t)$
  - *No* reward function
  - Teacher's demonstration: $s_0, a_0, s_1, a_1, s_2, a_2, \ldots$
    (= trace of the teacher's policy π^*)
- Inverse RL:
  - Can we recover R ?
- Apprenticeship learning via inverse RL
  - Can we then use this R to find a good policy ?
- Behavioral cloning
  - Can we directly learn the teacher's policy using supervised learning?

Behavioral cloning

- Formulate as standard machine learning problem
  - Fix a policy class
    - E.g., support vector machine, neural network, decision tree, deep belief net, …
  - Estimate a policy (=mapping from states to actions) from the training examples $(s_0, a_0), (s_1, a_1), (s_2, a_2), \ldots$

- Two of the most notable success stories:
  - Pomerleau, NIPS 1989: ALVINN
  - Sammut et al., ICML 1992: Learning to fly (flight sim)

Inverse RL vs. behavioral cloning

- **Which has the most succinct description: π* vs. R*?**

- Especially in planning oriented tasks, the reward function is often much more succinct than the optimal policy.

Inverse RL history

- 1964, Kalman posed the inverse optimal control problem and solved it in the 1D input case
- 1994, Boyd+al.: a linear matrix inequality (LMI) characterization for the general linear quadratic setting
- 2000, Ng and Russell: first MDP formulation, reward function ambiguity pointed out and a few solutions suggested
- 2004, Abbeel and Ng: inverse RL for apprenticeship learning---reward feature matching
- 2006, Ratliff+al: max margin formulation

Three broad categories of formalizations

- Max margin
- Feature expectation matching
- Interpret reward function as parameterization of a policy class

Basic principle

- Find a reward function R^* which explains the expert behaviour.
- Find R^* such that

$$\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi] \quad \forall \pi$$

- In fact a convex feasibility problem, but many challenges:
  - R=0 is a solution, more generally: reward function ambiguity
  - We typically only observe expert traces rather than the entire expert policy π^* --- how to compute left-hand side?
  - Assumes the expert is indeed optimal --- otherwise infeasible
  - Computationally: assumes we can enumerate all policies

Feature based reward function

- Let $R(s) = w^\top \phi(s)$, where $w \in \Re^n$, and $\phi : S \rightarrow \Re^n$.

$$\begin{aligned}
\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi] &= \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t w^\top \phi(s_t)|\pi] \\
&= w^\top \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi] \\
&= w^\top \underbrace{\mu(\pi)}
\end{aligned}$$

AKA Successor features!

Expected cumulative discounted sum of feature values or "feature expectations"

- Subbing into $\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi] \quad \forall \pi$

  gives us:

$$\text{Find } w^* \text{ such that } {w^*}^\top \mu(\pi^*) \geq {w^*}^\top \mu(\pi) \quad \forall \pi$$

Feature based reward function

$$\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|\pi] \quad \forall \pi$$

Let $R(s) = w^\top \phi(s)$, where $w \in \Re^n$, and $\phi : S \to \Re^n$.

Find w^* such that $w^{*\top} \mu^{(}\pi^*) \geq w^{*\top} \mu(\pi) \quad \forall \pi$

- Feature expectations can be readily estimated from sample trajectories.
- The number of expert demonstrations required scales with the number of features in the reward function.
- The number of expert demonstration required does *not* depend on
  - Complexity of the expert's optimal policy π*
  - Size of the state space

Ambiguity

- Standard max margin:

$$\min_w \|w\|_2^2$$
$$\text{s.t.} \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + 1 \qquad \forall \pi$$

- "Structured prediction" max margin:

$$\min_w \|w\|_2^2$$
$$\text{s.t.} \quad w^\top \mu^{(}\pi^*) \geq w^\top \mu(\pi) + m(\pi^*, \pi) \qquad \forall \pi$$

- Justification: margin should be larger for policies that are very different from π*.
- Example: m(π, π*) = number of states in which π* was observed and in which π and π* disagree

Expert suboptimality

- Structured prediction max margin with slack variables:

$$\min_{w,\xi} \|w\|_2^2 + C\xi$$
$$\text{s.t.} \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + m(\pi^*, \pi) - \xi \qquad \forall \pi$$

- Can be generalized to multiple MDPs (could also be same MDP with different initial state)

$$\min_{w,\xi^{(i)}} \|w\|_2^2 + C \sum_i \xi^{(i)}$$
$$\text{s.t.} \quad w^\top \mu(\pi^{(i)*}) \geq w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)}) - \xi^{(i)} \qquad \forall i, \pi^{(i)}$$

Three broad categories of formalizations

- Max margin (Ratliff+al, 2006)
  - Feature boosting [Ratliff+al, 2007]
  - Hierarchical formulation [Kolter+al, 2008]

- *Feature expectation matching (Abbeel+Ng, 2004)*
  - *Two player game formulation of feature matching (Syed+Schapire, 2008)*
  - *Max entropy formulation of feature matching (Ziebart+al,2008)*

- Interpret reward function as parameterization of a policy class. (Neu+Szepesvari, 2007; Ramachandran+Amir, 2007; Baker, Saxe, Tenenbaum, 2009; Mombaur, Truong, Laumond, 2009)

Feature matching

- Inverse RL starting point: find a reward function such that the expert outperforms other policies

Let $R(s) = w^\top \phi(s)$, where $w \in \Re^n$, and $\phi : S \rightarrow \Re^n$.

Find w^* such that ${w^*}^\top \mu^(\pi^*) \geq {w^*}^\top \mu(\pi) \quad \forall \pi$

- Observation in Abbeel and Ng, 2004: for a policy π to be guaranteed to perform as well as the expert policy π*, it suffices that the feature expectations match:

$$\|\mu(\pi) - \mu(\pi^*)\|_1 \leq \epsilon$$

implies that for all w with $\|w\|_\infty \leq 1$:

$$|{w^*}^\top \mu(\pi) - {w^*}^\top \mu(\pi^*)| \leq \epsilon$$

Apprenticeship learning [Abbeel & Ng, 2004]

- Assume $R_w(s) = w^\top \phi(s)$ for a feature map $\phi : S \rightarrow \Re^n$.
- Initialize: pick some controller π_0.
- Iterate for i = 1, 2, … :
  - **"Guess" the reward function**:

    Find a reward function such that the teacher maximally outperforms all previously found controllers.

    $$\max_{\gamma, w : \|w\|_2 \leq 1} \gamma$$
    $$s.t. \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + \gamma \quad \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$

  - **Find optimal control policy** π_i for the current guess of the reward function R_w.
  - If $\gamma \leq \varepsilon/2$ exit the algorithm.

Reward function parameterizing the policy class

- Recall:

$$\begin{aligned} V^*(s;R) &= R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^*(s;R) \\ Q^*(s,a;R) &= R(s) + \gamma \sum_{s'} P(s'|s,a) V^*(s;R) \end{aligned}$$

- Let's assume our expert acts according to:

$$\pi(a|s;R,\alpha) = \frac{1}{Z(s;R,\alpha)} \exp(\alpha Q^*(s,a;R))$$

- Then for any R and α, we can evaluate the likelihood of seeing a set of state-action pairs as follows:

$$P((s_1,a_1)) \ldots P((s_m,a_m)) = \frac{1}{Z(s_1;R,\alpha)} \exp(\alpha Q^*(s_1,a_1;R)) \ldots \frac{1}{Z(s_m;R,\alpha)} \exp(\alpha Q^*(s_m,a_m;R))$$

Reward function parameterizing the policy class --- deterministic systems

- Assume deterministic system $x_{t+1} = f(x_t, u_t)$ and an observed trajectory $(x_0^*, x_1^*, \ldots, x_T^*)$

- Find reward function by solving:

$$
\begin{aligned}
\min_w \quad & \sum_{t=0}^{T} \|x_t^* - x_t^w\|_2 \\
\text{s.t.} \quad & x^w \text{is the solution of:} \\
& \max_x \sum_{t=0}^{T} \sum_i w_i \phi_i(x_t) \\
& \text{s.t.} x_{t+1} = f(x_t, u_t) \\
& \qquad x_0 = x_0^*, \quad x_T = x_T^*
\end{aligned}
$$

[Mombaur, Truong, Laumond, 2009]

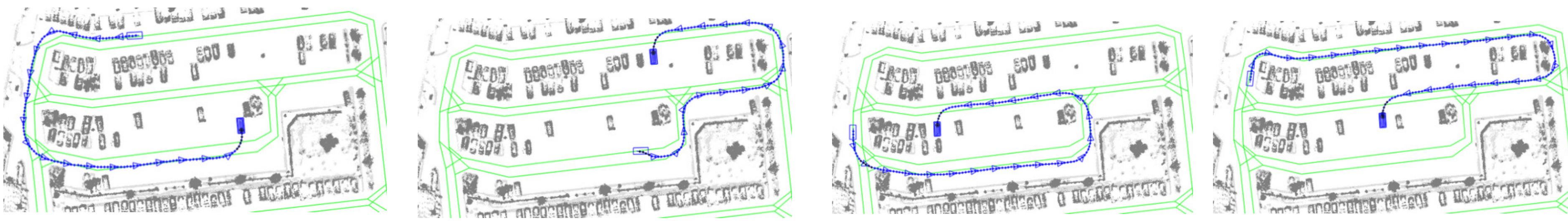
Parking lot navigation

- Reward function trades off:
  - Staying "on-road,"
  - Forward vs. reverse driving,
  - Amount of switching between forward and reverse,
  - Lane keeping,
  - On-road vs. off-road,
  - Curvature of paths.

[Abbeel et al., IROS 08]

Experimental setup

- Demonstrate parking lot navigation on "train parking lots."



- Run our apprenticeship learning algorithm to find the reward function.
- Receive "test parking lot" map + starting point and destination.
- Find the trajectory that maximizes the *learned reward function* for navigating the test parking lot.

Nice driving style

Sloppy driving-style

“Don’t mind reverse” driving-style