

Principles of Robot Autonomy II

Imitation Learning (2)



Today's itinerary

- Recap of Behavioral Cloning and Imitation Learning with Interactive Experts
- Inverse RL – Maximum Margin Planning
- Inverse RL – Maximum Entropy IRL
- Demonstration Quality
- Learning from other sources of data (preferences, physical feedback)

Today's itinerary

- Recap of Behavioral Cloning and Imitation Learning with Interactive Experts
- Inverse RL – Maximum Margin Planning
- Inverse RL – Maximum Entropy IRL
- Demonstration Quality
- Learning from other sources of data (preferences, physical feedback)

Why Imitation Learning?

For the Sake of Robot Learning:

- It is difficult to learn from sparse rewards (unless data is cheap and you don't care about seeing lots of failures).
- Hand-designing rewards is hard.

For the Sake of Learning Human Models:

- Learning human's intents, preferences, and underlying reward functions.



Problem Setup

MDP with no reward functions:

- State space, S (sometimes partially observable)
- Actions space, A
- An expert policy π^* that maps states to distributions over actions: $\pi^*(s) \rightarrow P(s)$
- Transition model $P(s_{t+1}|s_t, a_t)$: simulator or environment

Goal: Learn an imitating policy $\pi_\theta(s)$ that imitates the expert demonstrations

Types of Imitation Learning

Behavioral Cloning

$$\arg \min_{\theta} \mathbb{E}_{(s, a^*) \sim P^*} L(a^*, \pi_{\theta}(s))$$

Works well when P^* is close to P_{θ}

Direct Policy Learning (via Interactive Demonstrator)

Requires Interactive Demonstrator (BC is a 1-step special case)

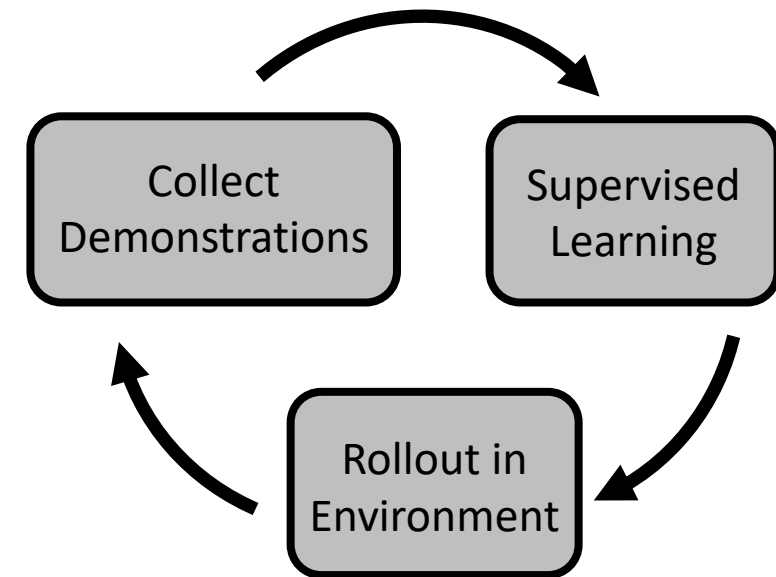
Inverse RL

Learn r such that:

$$\pi^* = \arg \max_{\theta} \mathbb{E}_{s \sim P(s|\theta)} r(s, \pi_{\theta}(s))$$

RL problem

Assume learning r is statistically easier than directly learning π^*



Today's itinerary

- Recap of Behavioral Cloning and Imitation Learning with Interactive Experts
- Inverse RL – Maximum Margin Planning
- Inverse RL – Maximum Entropy IRL
- Demonstration Quality
- Learning from other sources of data (preferences, physical feedback)

What can go wrong with policy learning?

Behavioral cloning: mimics the expert directly

- No reasoning about outcomes or dynamics
- No notion of intentions
- Expert can be suboptimal
- Expert might have different embodiments
- Safety and Robustness

History of Inverse Reinforcement Learning

- 1964: Kalman posed the inverse optimal control problem and solved it in 1D
- 1994: Boyd et al. A linear matrix inequality (LMI) characterization for the linear quadratic setting
- 2000: Ng, Russell. Proposed the first MDP formulation and issues around reward function ambiguity
- 2004: Abbeel, Ng. Inverse RL with feature matching for apprenticeship learning
- 2006: Ratliff et al. Max Margin Planning (MMP) Formulation
- 2008: Zeibart et al. Max Entropy Formulation
- Since then... Active Inverse RL, Integration with other types of data, Iterative approaches to update Reward and Policy (GAIL, etc.), Implicit BC, images as inputs, etc.

Apprenticeship Learning



[Abbeel, Ng, 2004]

Problem Setup: Behavioral Cloning

MDP with no reward functions:

- State space, S (sometimes partially observable)
- Actions space, A
- An expert policy π^* that maps states to distributions over actions: $\pi^*(s) \rightarrow P(s)$
- Transition model $P(s_{t+1}|s_t, a_t)$: simulator or environment

Goal: Learn an imitating policy $\pi_\theta(s)$ that imitates the expert demonstrations

Problem Setup: Inverse RL

MDP with no reward functions:

- State space, S (sometimes partially observable)
- Actions space, A
- An expert policy π^* that maps states to distributions over actions: $\pi^*(s) \rightarrow P(s)$
- Transition model $P(s_{t+1}|s_t, a_t)$: simulator or environment

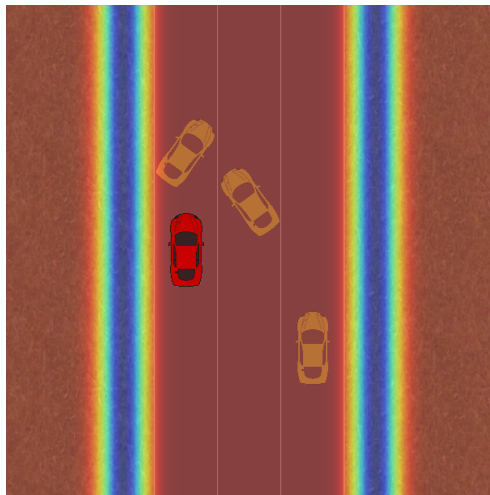
~~**Goal:** Learn an imitating policy $\pi_\theta(s)$ that imitates the expert demonstrations~~

Goal: Learn a reward function assuming the experts are optimal

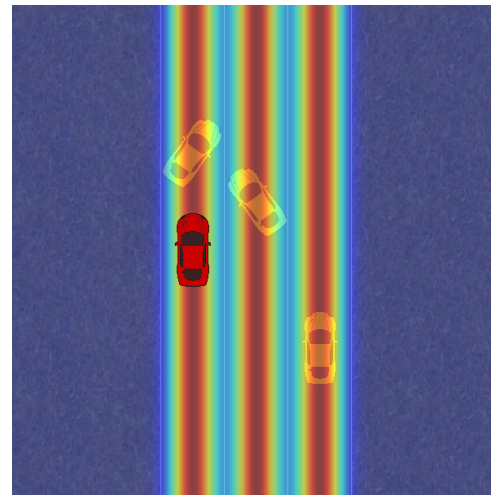
Inverse Reinforcement Learning

Assume the reward function is a linear combination of features:

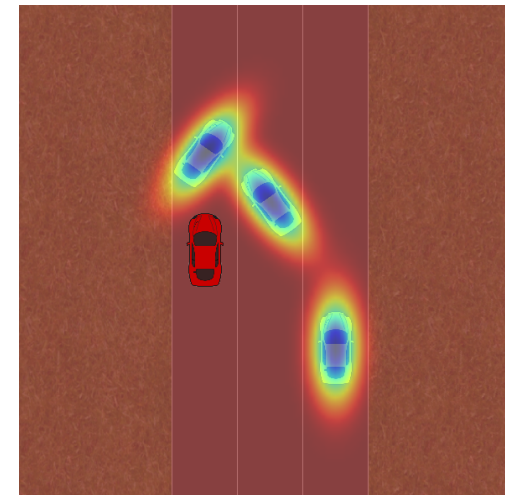
$$R(s) = w^T \varphi(s) \quad w \in \mathbb{R}^n \quad \varphi: S \rightarrow \mathbb{R}^n$$



(a) Features for the boundaries of the road



(b) Feature for staying inside the lanes.



(c) Features for avoiding other vehicles.

Inverse Reinforcement Learning

Assume the reward function is a linear combination of features:

$$R(s) = w^\top \varphi(s) \quad w \in \mathbb{R}^n \quad \varphi: S \rightarrow \mathbb{R}^n$$

The goal is to recover the weights: w

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \right] \\ &= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t w^\top \varphi(s_t) \right] = w^\top \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \varphi(s_t) \right] = w^\top \mu(\pi) \end{aligned}$$

Feature Expectations

How to deal with reward ambiguity?

Reward ambiguity: There are many reward functions under which the expert demonstrations are optimal!!

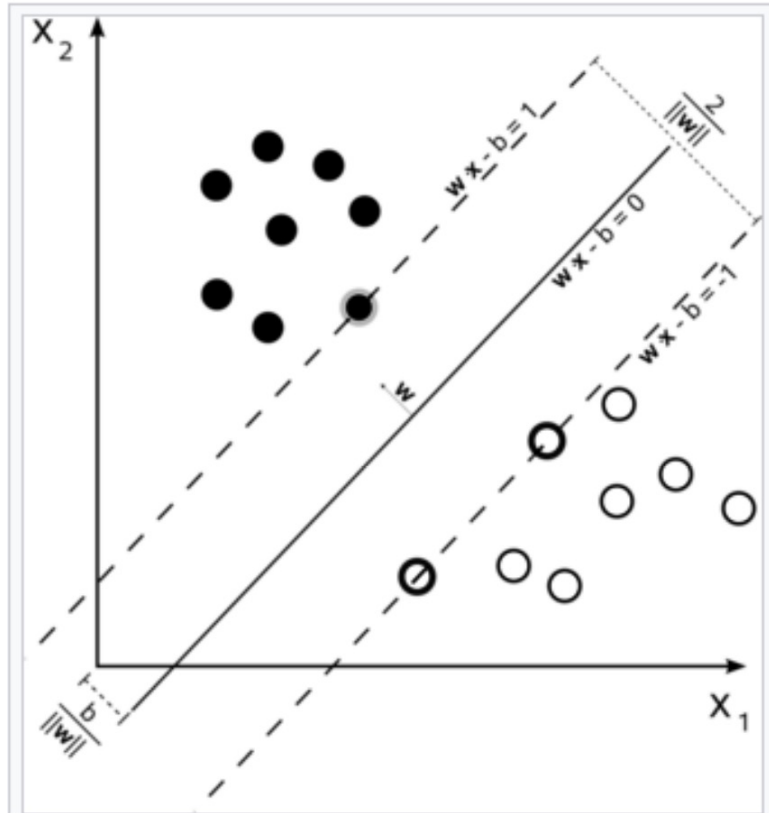
How to deal with reward ambiguity?

Reward ambiguity: There are many reward functions under which the expert demonstrations are optimal!!

Which reward function should we pick?

- **Maximum Margin Planning:** Looks for the one that separates the optimal policy best.

Aside: Maximum Margin Classifiers



Given a training dataset of $(x_1, y_1), \dots, (x_n, y_n)$, where y_i is either 1 or -1 identifying the class x_i is in. We want to find the maximum margin hyperplane that divides the points so the distance between the hyperplane and the nearest point from each class is maximized.

"Minimize $\|\vec{w}\|$ subject to $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$, for $i = 1, \dots, n$ "

Feature Matching

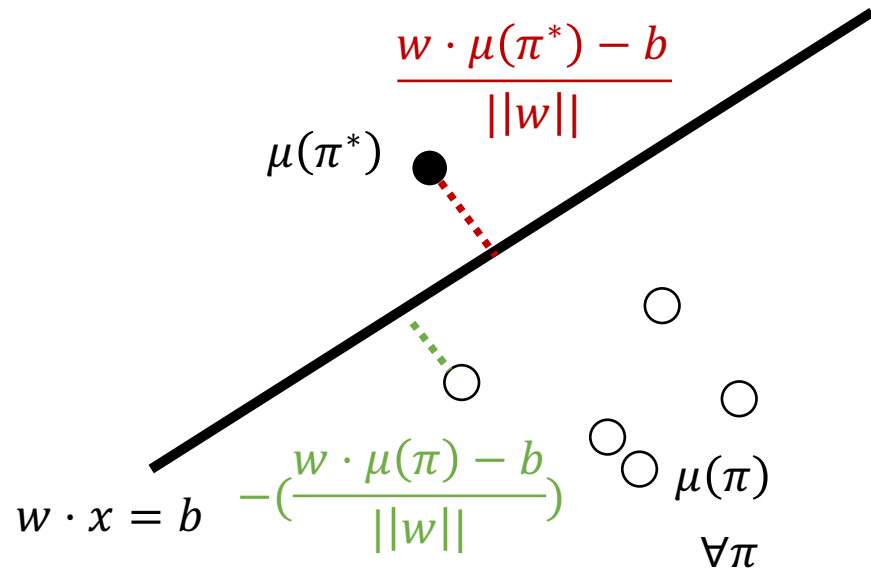
By definition, the value of optimal policy with respect to true reward is greater than the value of any other policy:

$$V^{\pi^*}(s) \geq V^{\pi}(s) \quad \forall \pi$$

$$\mathbb{E}_{\pi^*} \left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) \right] \geq \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) \right]$$

$$w^{*\top} \mu(\pi^*) \geq w^{*\top} \mu(\pi)$$

Maximum Margin Planning (MMP)



$$w \cdot \mu(\pi^*) \geq w \cdot \mu(\pi) \rightarrow \begin{aligned} w \cdot \mu(\pi^*) - b &\geq 0 \\ w \cdot \mu(\pi) - b &\leq 0 \end{aligned}$$

$$\begin{aligned} \max_w \min_{\pi} & \frac{\left(\frac{w \cdot \mu(\pi^*) - b}{\|w\|} \right) + \left(- \left(\frac{w \cdot \mu(\pi) - b}{\|w\|} \right) \right)}{2} \\ &= \max_w \min_{\pi} \frac{w \cdot \mu(\pi^*) - w \cdot \mu(\pi)}{\|w\|} \\ &= \min_w \|w\| \quad \text{s.t.} \quad w \cdot \mu(\pi^*) - w \cdot \mu(\pi) \geq 1 \end{aligned}$$

Maximally separate the policy induced by our learned reward functions from suboptimal policies.

Maximum Margin Planning (MMP)

Standard formulation:

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

More involved formulation:

$$\min_w \quad \left\| w \right\|_2^2 + C\nu$$

Add slack variables to incorporate expert suboptimality

$$\text{s.t.} \quad w^\top \mu(\pi^*) \geq w^\top \mu(\pi) + m(\pi^*, \pi) - \nu \quad \forall \pi$$

Give more margin if π and π^ are very different from each other.*

How to deal with reward ambiguity?

Reward ambiguity: There are many reward functions under which the expert demonstrations are optimal!!

Which reward function should we pick?

- **Maximum Margin Planning:** Looks for the one that separates the optimal policy best.

- **Maximum Entropy IRL:** Looks for the one where expert demonstrations are drawn from a high entropy distribution.

Max Entropy IRL

Let $\xi = \{(s_1, a_1), \dots, (s_T, a_T)\}$ be a sequence of state and actions.
We let $D = \{\xi_1, \dots, \xi_{|D|}\}$ to be the set of expert demonstrations.

$$f_D = \frac{1}{|D|} \sum_{\xi \in D} f(\xi)$$

Let's define a feature function over trajectories: $f: \Xi \rightarrow \mathbb{R}^n$

Empirical feature expectations

$$\mathbb{E}_{\xi \sim P(\xi)} \left[\sum_{t=1}^T \gamma^t R(s_t) \right] = \mathbb{E}_{\xi \sim P(\xi)} \left[\sum_{t=1}^T \gamma^t w^\top \varphi(s_t) \right] = w^\top \mathbb{E}_{\xi \sim P(\xi)} [f(\xi)]$$

Expected Return

Weighted Feature Expectations

In addition, let us define the reward using a noisily rational model assuming:

$$P(\xi) \propto \exp(R(\xi))$$

Max Entropy IRL

Selects the least committed distribution (maximizing entropy)

Goal: Find the distribution over the observations (expert trajectories) that matches empirical feature counts in expectation, and maximizes entropy.

$$\begin{aligned} & \max_P \int -P(\xi) \log P(\xi) d\xi \\ \text{s.t.} \quad & \mathbb{E}_{\xi \sim P(\xi)} [f(\xi)] = \int P(\xi) f(\xi) d\xi = f_D \\ & \int P(\xi) d\xi = 1 \\ & P(\xi) \geq 0, \quad \forall \xi \in \Xi \end{aligned}$$

Write the Lagrangian of this optimization:

$$\mathcal{L}(P; \lambda, \nu) = \int -P(\xi) \log P(\xi) d\xi + \lambda^\top \left(\int P(\xi) f(\xi) d\xi - f_D \right) + \nu \left(\int P(\xi) d\xi - 1 \right)$$

$$\mathcal{L}(P; \lambda, \nu) = \underbrace{\int (-P(\xi) \log P(\xi) + \lambda^\top P(\xi) f(\xi) + \nu P(\xi)) d\xi}_{F(P(\xi))} + \nu - \lambda^\top f_D$$

$$\mathcal{L}(P; \lambda, \nu) = \int F(P(\xi)) d\xi + \nu - \lambda^\top f_D$$

Set the gradient of \mathcal{L} with respect to P to be zero

$$\mathcal{L}(P; \lambda, \nu) = \underbrace{\int (-P(\xi) \log P(\xi) + \lambda^\top P(\xi) f(\xi) + \nu P(\xi)) d\xi}_{F(P(\xi))} + \nu - \lambda^\top f_D$$

$$\mathcal{L}(P; \lambda, \nu) = \int F(P(\xi)) d\xi + \nu - \lambda^\top f_D$$

$$\nabla_P \mathcal{L}(P; \lambda, \nu) = 0$$

$$\frac{\partial F(P(\xi))}{\partial P(\xi)} = -\log(P(\xi)) - 1 + \lambda^\top f(\xi) + \nu = 0$$

Using Euler-Lagrange Equation, we only need to set the gradient of each term in the integral equal to zero ($\frac{\partial F}{\partial P} = 0$)

$$-\log(P(\xi)) - 1 + \lambda^\top f(\xi) + \nu = 0$$

$$P^*(\xi) = \exp(\lambda^\top f(\xi) + \nu - 1)$$

Replace P^* in the Lagrangian \mathcal{L} :

$$P^*(\xi) = \exp(\lambda^\top f(\xi) + \nu - 1)$$

$$\mathcal{L}(P^*; \lambda, \nu) = \int (-P^*(\xi) \log P^*(\xi) + \lambda^\top P^*(\xi) f(\xi) + \nu P^*(\xi)) d\xi + \nu - \lambda^\top f_D$$

$$\begin{aligned} \mathcal{L}(P^*; \lambda, \nu) = \int & (-\exp(\lambda^\top f(\xi) + \nu - 1) \log(\exp(\lambda^\top f(\xi) + \nu - 1)) + \\ & \lambda^\top (\exp(\lambda^\top f(\xi) + \nu - 1)) f(\xi) + \\ & \nu \exp(\lambda^\top f(\xi) + \nu - 1)) d\xi \\ & + \nu - \lambda^\top f_D \end{aligned}$$

$$\begin{aligned} \mathcal{L}(P^*; \lambda, \nu) = \int & (-\lambda^\top f(\xi) \exp(\lambda^\top f(\xi) + \nu - 1) - \nu \exp(\lambda^\top f(\xi) + \nu - 1) + \exp(\lambda^\top f(\xi) + \nu - 1) \\ & + \lambda^\top (\exp(\lambda^\top f(\xi) + \nu - 1)) f(\xi) + \nu \exp(\lambda^\top f(\xi) + \nu - 1)) d\xi \\ & + \nu - \lambda^\top f_D \end{aligned}$$

$$\mathcal{L}(P^*; \lambda, \nu) = \int \exp(\lambda^\top f(\xi) + \nu - 1) d\xi - \lambda^\top f_D - \nu$$

Recap:

$$P^*(\xi) = \exp(\lambda^\top f(\xi) + \nu - 1)$$

$$\mathcal{L}(P^*; \lambda, \nu) = \int \exp(\lambda^\top f(\xi) + \nu - 1) d\xi - \lambda^\top f_D - \nu$$

Solving for ν^* :

$$\frac{\partial \mathcal{L}(P^*; \lambda, \nu)}{\partial \nu} = 0$$

$$e^{\nu-1} \int \exp(\lambda^\top f(\xi)) d\xi - 1 = 0$$

$$e^{-\nu} = \int \exp(\lambda^\top f(\xi) - 1) d\xi$$

$$\nu^* = -\log\left(\int \exp(\lambda^\top f(\xi) - 1) d\xi\right)$$

Replace ν^* in P^* :

$$P^*(\xi) = \exp(\lambda^\top f(\xi) + \nu - 1)$$

$$P^*(\xi) = \exp(\lambda^\top f(\xi) - \log\left(\int \exp(\lambda^\top f(\xi) - 1)d\xi\right) - 1)$$

$$P^*(\xi) = \frac{\exp(\lambda^\top f(\xi) - 1)}{\int \exp(\lambda^\top f(\xi) - 1)d\xi}$$

$$P^*(\xi) = \frac{\exp(\lambda^\top f(\xi))}{\int \exp(\lambda^\top f(\xi))d\xi}$$

$$P^*(\xi) = \exp(\lambda^\top f(\xi) + \nu - 1)$$

$$\nu^* = -\log\left(\int \exp(\lambda^\top f(\xi) - 1)d\xi\right)$$

This nicely matches our noisily rational reward formulation:
 $P(\xi) \propto \exp(R(\xi))$

Let us define the reward as $R(\xi) = \lambda^\top f(\xi)$, where the reward weights are the dual variables λ

Finding the reward weights λ

By solving the optimization, we have: $P^*(\xi; \lambda) = \frac{\exp(\lambda^\top f(\xi))}{\int \exp(\lambda^\top f(\xi)) d\xi}$

We look for λ parameters that maximize the log-likelihood of observing expert trajectories

$$\begin{aligned}\lambda^* &= \arg \max_{\lambda} P(\xi_{1..D}; \lambda) \\ &= \arg \max_{\lambda} \underbrace{\lambda^\top f_D - \log\left(\int \exp(\lambda^\top f(\xi)) d\xi\right)}_M\end{aligned}$$

$$\nabla_{\lambda} M = f_D - \mathbb{E}_{\xi \sim P(\xi; \lambda)}[f(\xi)]$$

$$\lambda_{i+1} \leftarrow \lambda_i + \alpha(f_D - \mathbb{E}_{\xi \sim P(\xi; \lambda)}[f(\xi)])$$

Max Entropy IRL

- 1) Initialize λ and collect expert demonstrations D .
- 2) Solve for the optimal policy $\pi_\lambda(a|s)$ with respect to λ . (RL Loop)
- 3) Solve for state visitation frequencies $P_\lambda(s)$.
- 4) Compute the gradient $\nabla_\lambda M$.
- 5) Update λ with one gradient step.

This assumes access to the dynamics (transition function) and having low dimensional systems to be able to solve for the policy using RL.

End-to-end driving via conditional imitation Learning

End-to-end Driving via Conditional Imitation Learning

Felipe Codevilla, Antonio López - Computer Vision Center (CVC)
Matthias Müller - King Abdullah University of Science and Technology (KAUST)
Vladlen Koltun, Alexey Dosovitskiy - Intel Visual Computing Lab

We propose conditional imitation learning which allows an autonomous vehicle trained end-to-end to be directed by high-level commands.

Experiments in simulation and on a physical vehicle show that the method allows for goal-directed navigation guided by a topological planner or a user.



Today's itinerary

- Recap of Behavioral Cloning and Imitation Learning with Interactive Experts
- Inverse RL – Maximum Margin Planning
- Inverse RL – Maximum Entropy IRL
- Demonstration Quality
- Learning from other sources of data (preferences, physical feedback)