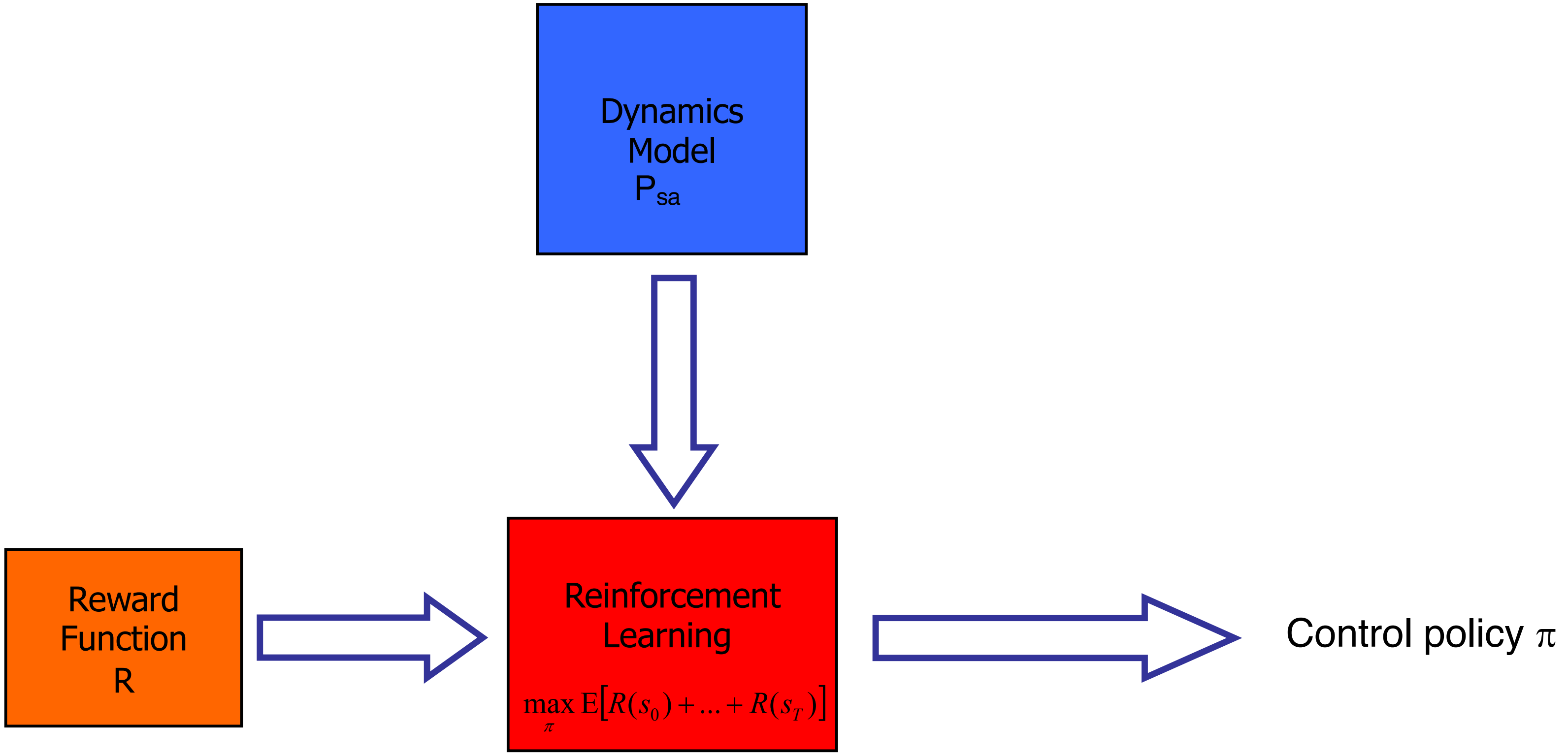
Learning by Demonstration: Imitation Learning, Inverse Reinforcement Learning, and Apprenticeship Learning

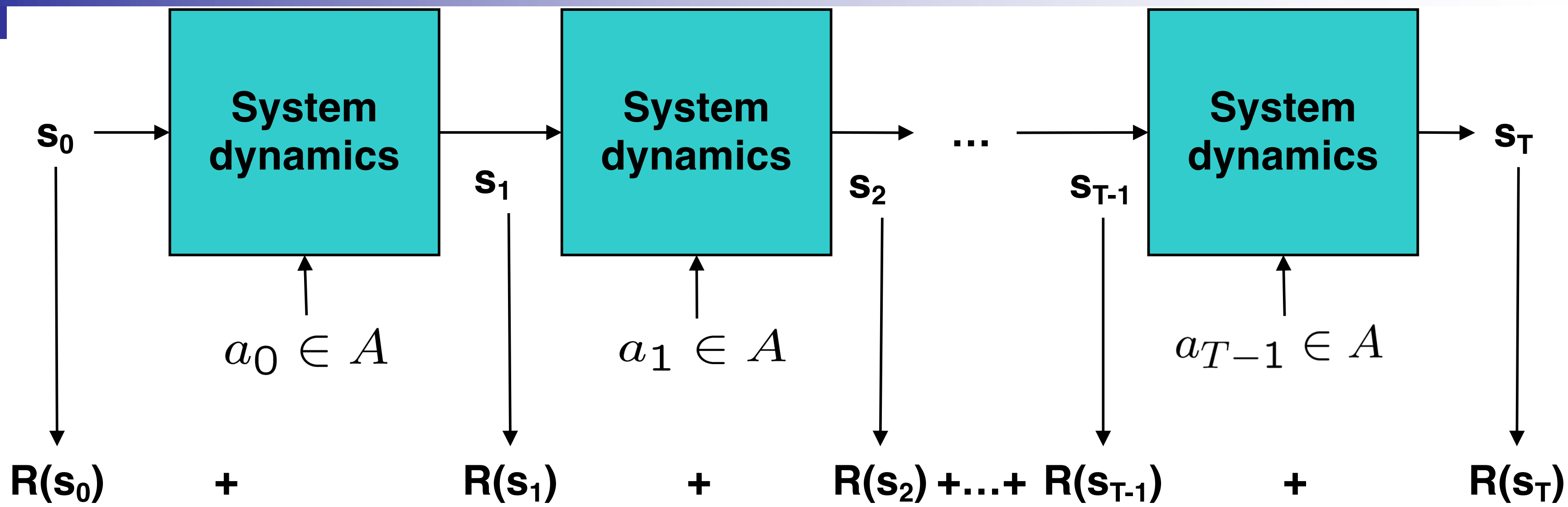
Luís Macedo
University of Coimbra

RL formalism



Pieter Abbeel and
Andrew Y. Ng

RL formalism



- Assume that at each time step, our system is in some state s_t.
- Upon taking an action a, our state randomly transitions to some new state s_{t+1}.
- We are also given a reward function R.
- The goal: Pick actions over time so as to maximize the expected score: $E[R(s_0) + R(s_1) + \ldots + R(s_T)]$.

RL formalism

- Markov Decision Process *(S,A,P_{sa},s_0,R)*

  $$R(s) = w^T \phi(s),$$

  $\phi : S \rightarrow [0,1]^k$: k-dimensional feature vector.

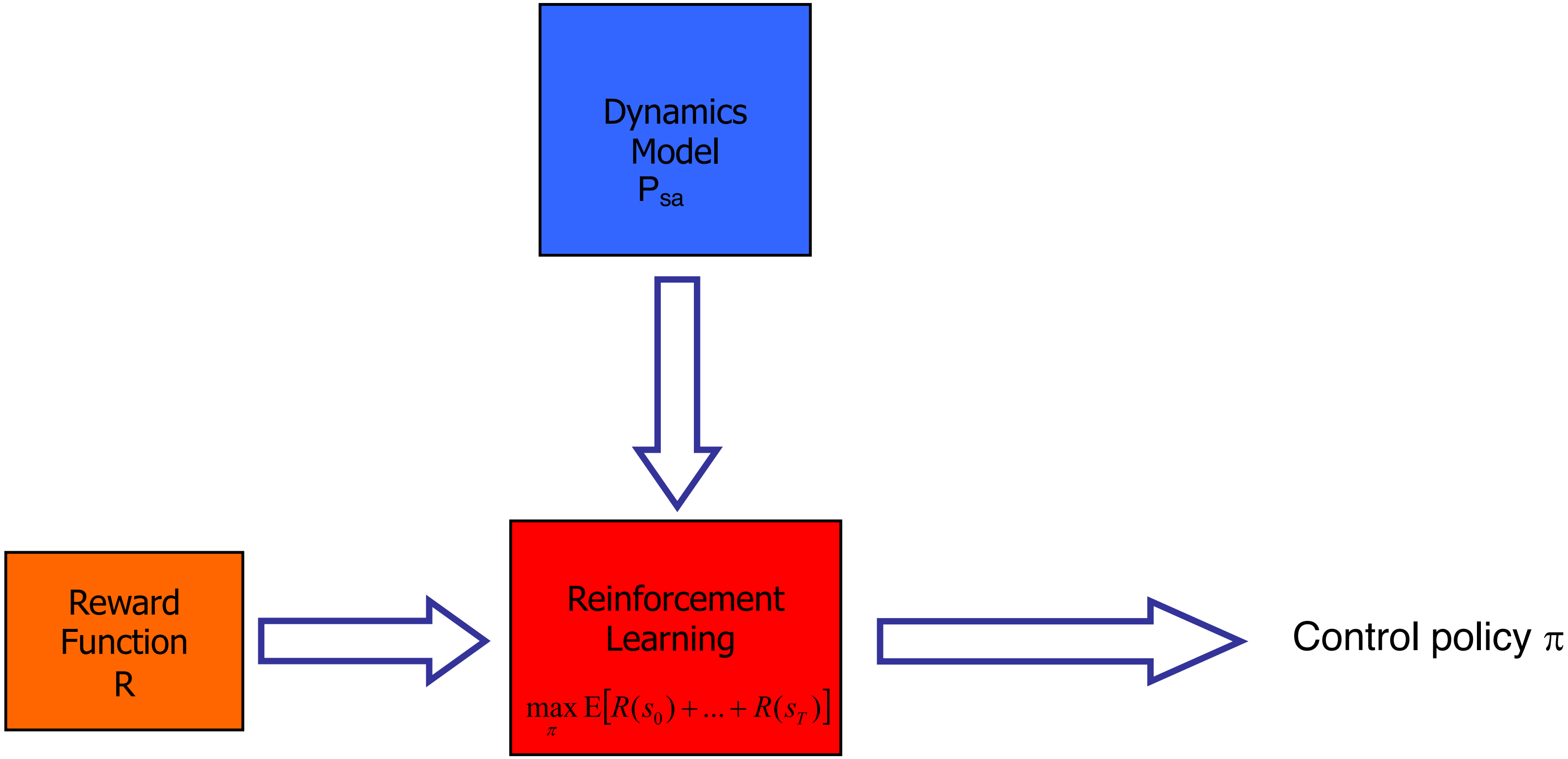
- W.l.o.g. we assume $\|w\|_2 \leq 1.$

- Policy $\pi : S \rightarrow A.$

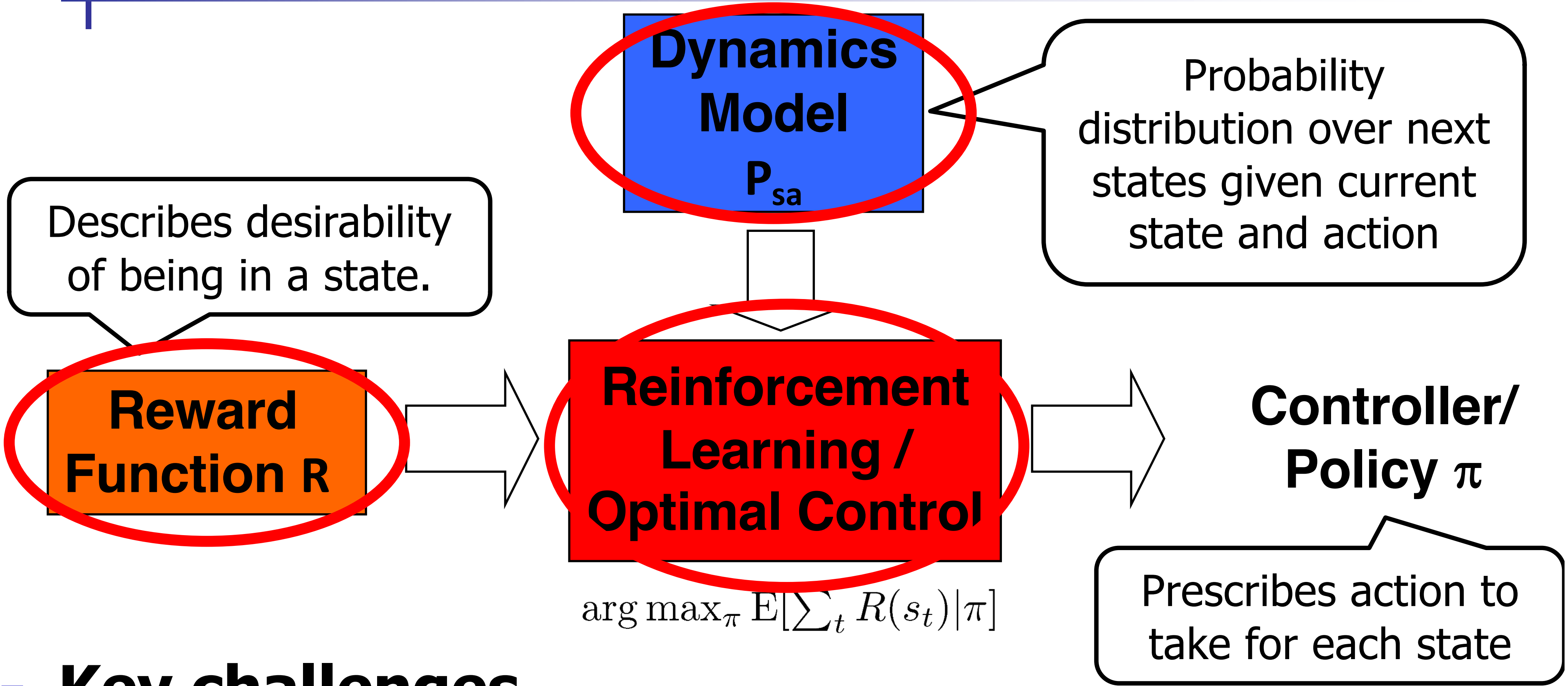
- Utility of a policy π for reward $R=w^T\phi$

- $$U_w(\pi) = E[\sum_{t=0}^{T} R(s_t)|\pi]$$

RL formalism



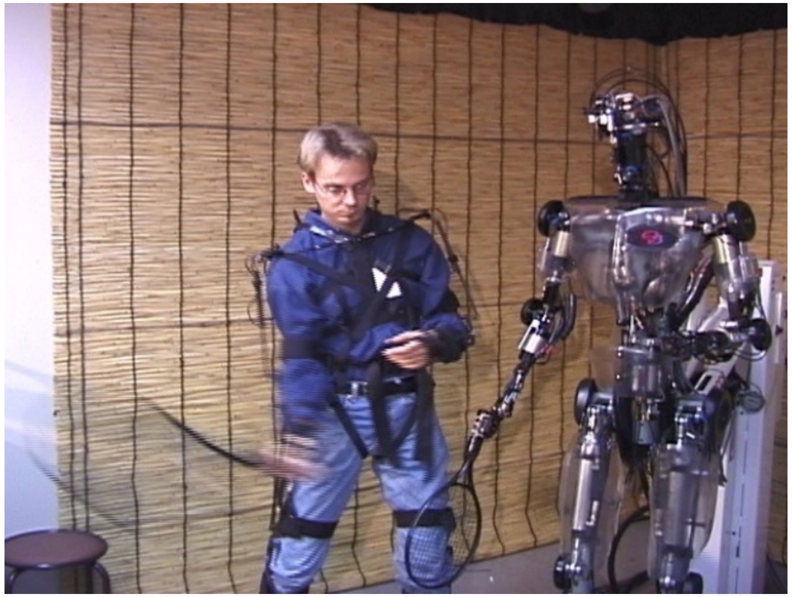
Big picture and key challenges

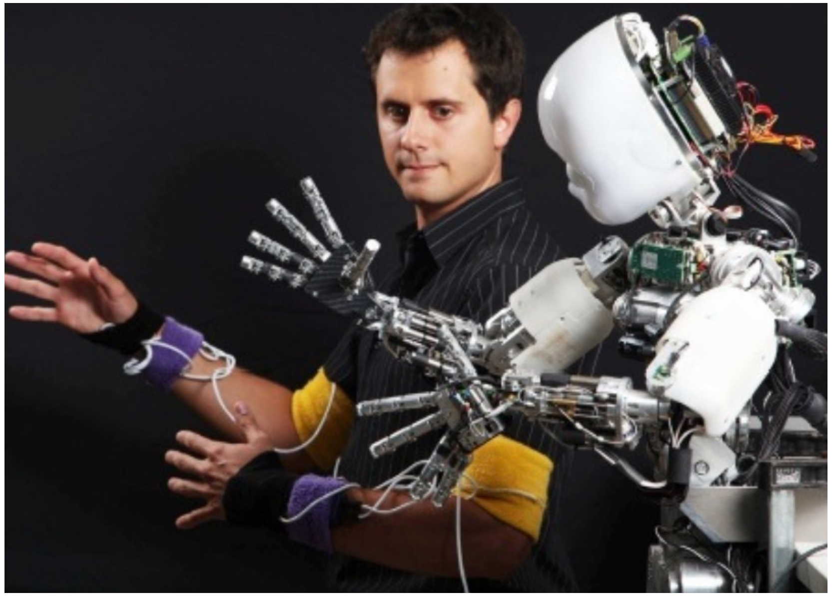


- **Key challenges**
  - Providing a formal specification of the control task.
  - Building a good dynamics model.
  - Finding closed-loop controllers.

Example task: driving

Imitation Learning





Problem setup

- Input:
  - Dynamics model / Simulator $P_{sa}(s_{t+1} \mid s_t, a_t)$
  - *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^*)

- Desired output:
  - Policy $\pi : S \rightarrow A$, which (ideally) has performance guarantees, i.e.,

$$\mathrm{E}[\frac{1}{T}\sum_t R^*(s_t)|\pi] \geq \mathrm{E}[\frac{1}{T}\sum_t R^*(s_t)|\pi^*] - \epsilon.$$

  - Note: R* is unknown.

Solutions

- Ideally we want dense in time rewards to closely guide the agent closely along the way. Who will supply those shaped rewards?
  - We will manually design them: “cost function design by hand remains one of the ’black arts’ of mobile robotics, and has been applied to untold numbers of robotic systems”
  - We will learn them from demonstrations: “rather than having a human expert tune a system to achieve desired behavior, the expert can demonstrate desired behavior and the robot can tune itself to match the demonstration”

Solution – Learning from Demonstration

- Learning from demonstrations a.k.a. Imitation Learning: Supervision through an expert (teacher) that provides a set of demonstration trajectories: sequences of states and actions.

- **Imitation learning** is useful when it is easier for the expert to demonstrate the desired behavior rather than:
  - coming up with a reward function that would generate such behavior
  - coding up with the desired policy directly

Solution – Learning from Demonstration

- Two broad approaches :
  - Direct: Supervised training of policy (mapping states to actions) using the demonstration trajectories as ground-truth (a.k.a. behavior cloning; especial case: it can be interactive, i.e., active behaviour cloning)
    - Can we directly learn the teacher's policy using supervised learning?
  - Indirect: Learn the unknown reward function/goal of the teacher, and derive the policy from these, a.k.a. Inverse Reinforcement Learning:
    - Inverse RL: Can we recover R?
    - Apprenticeship learning via inverse RL: Can we then use this R to find a good policy?

Solution – Learning from Demonstration

- Experts can be:
  - Humans
  - Optimal or near Optimal Planners/Controllers

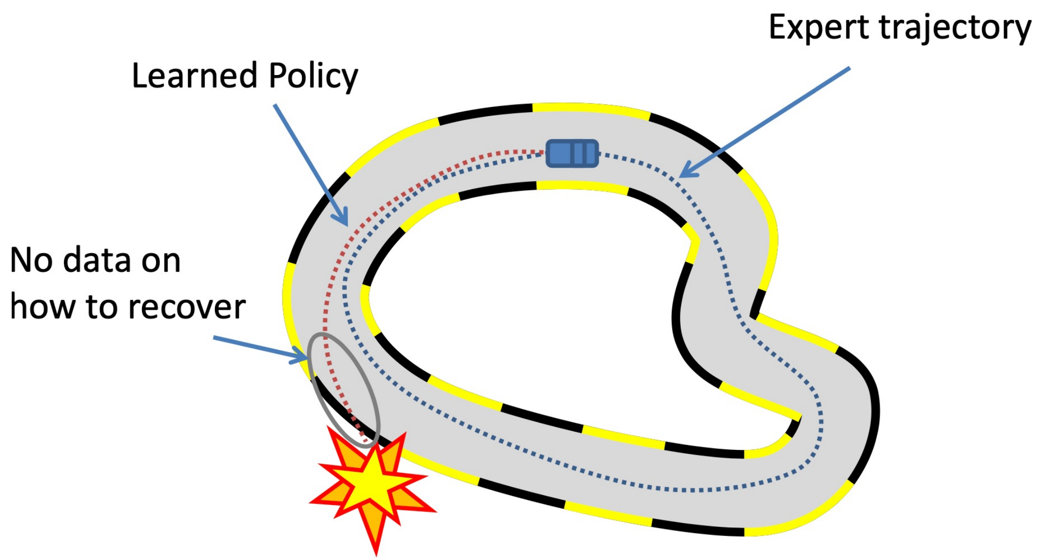
Behavioral cloning

- 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 from the training examples (s_0, a_0), (s_1, a_1), (s_2, a_2), …

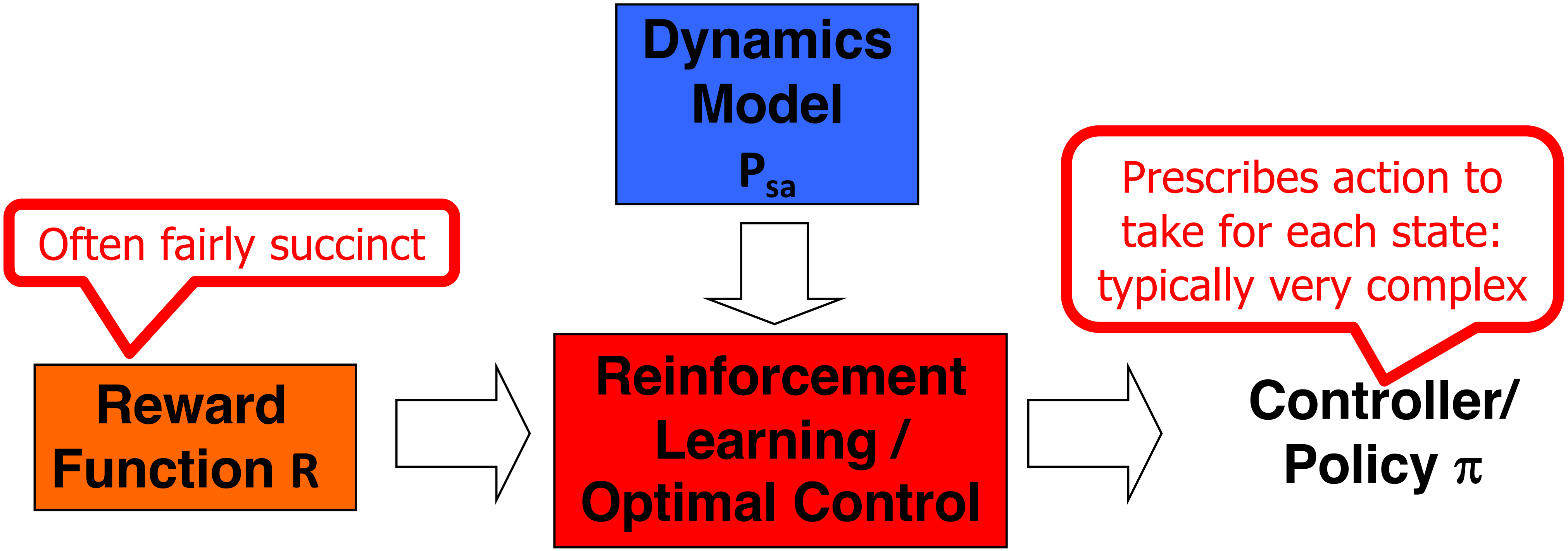
- E.g.: http://robotwhisperer.org/bird-muri/
  - https://youtu.be/hNsP6-K3Hn4
- E.g., Pomerleau, 1989; Sammut et al., 1992; Kuniyoshi et al., 1994; Demiris & Hayes, 1994; Amit & Mataric, 2002.

Behavioral cloning

- Limitations:
  - Underlying assumption: policy simplicity
  - makes mistakes: enters new states from which it cannot recover



Inverse RL

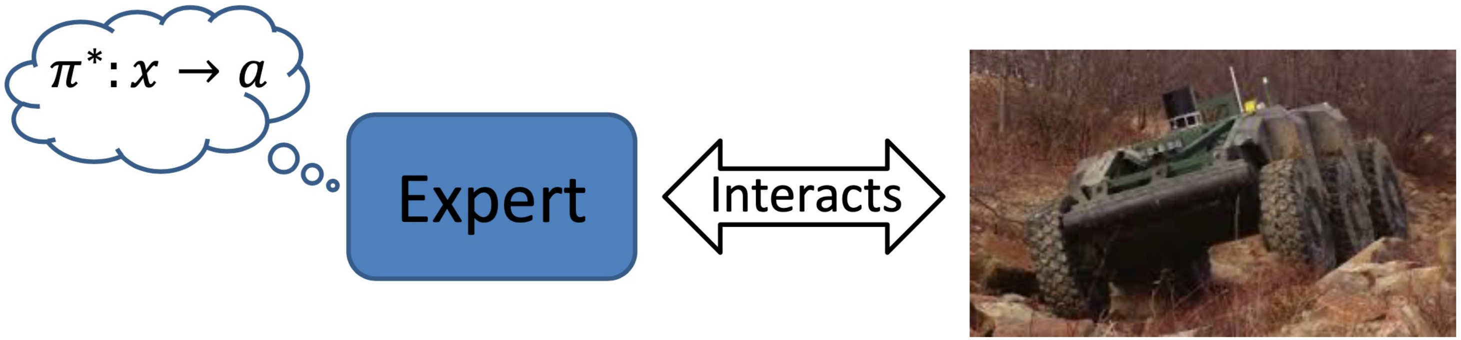


$$\arg\max_\pi \mathrm{E}[\textstyle\sum_t R(s_t)|\pi]$$

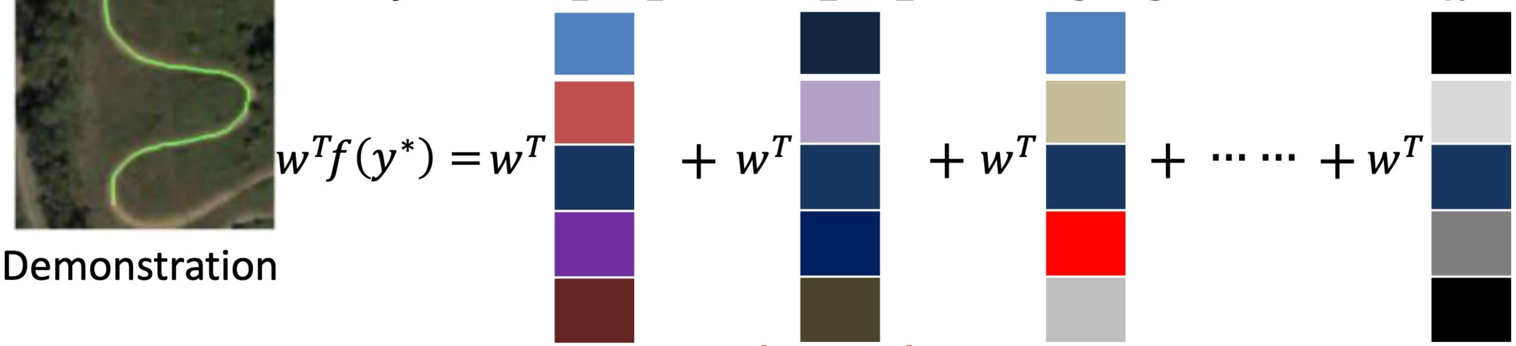
E.g., $R^* = w_1^* \mathbf{1}\{''in\ right\ lane''\} + w_2^* \mathbf{1}\{''safe\ distance''\}$

Inverse RL

- Assumption: reward as a linear function of state features



$$y^* = (x_1, a_1) \to (x_2, a_2) \to (x_3, a_3) \to \cdots \to (x_n, a_n)$$



Demonstration

Expert trajectory reward/cost

Inverse RL

- Assumption: reward as a linear function of state features

Find a reward function R^* which explains the expert behavior

i.e., assume expert follows optimal policy, given her R^*

Find R^* such that

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

Inverse RL

- Idea:

given:

states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$

(sometimes) transitions $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$

samples $\{\tau_i\}$ sampled from $\pi^\star(\tau)$

learn $r_\psi(\mathbf{s}, \mathbf{a})$

reward parameters

...and then use it to learn $\pi^\star(\mathbf{a}|\mathbf{s})$

RL vs. Inverse RL

given:

states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$

(sometimes) transitions $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$

reward function $r(\mathbf{s}, \mathbf{a})$

learn $\pi^\star(\mathbf{a}|\mathbf{s})$

given:

states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$

(sometimes) transitions $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$

samples $\{\tau_i\}$ sampled from $\pi^\star(\tau)$

learn $r_\psi(\mathbf{s}, \mathbf{a})$

reward parameters

...and then use it to learn $\pi^\star(\mathbf{a}|\mathbf{s})$

Inverse RL

- Idea:

  1. Guess an initial reward function $R(s)$
  2. Learn policy $\pi(s)$ that optimizes $R(s)$
  3. Whenever $\pi(s)$ chooses action different from expert $\pi^*(s)$
     - Update estimate of $R(s)$ to assure value of $\pi^*(s)$ > value of $\pi(s)$
  4. Go to 2

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.

$$\max_{\gamma, w : \|w\|_2 \leq 1} \gamma$$

$$s.t. \quad \mathrm{E}[\sum_{t=0}^{T} R_w(s_t)|\pi^*] \geq \mathrm{E}[\sum_{t=0}^{T} R_w(s_t)|\pi] + \gamma \quad \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$

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

Learning through reward functions rather than directly learning policies.

There is no reward function for which the teacher significantly outperforms thus-far found policies.

Apprenticeship learning [Abbeel & Ng, 2004]

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

Apprenticeship learning [Abbeel & Ng, 2004] - Theoretical guarantees

Theorem.
To ensure with probability at least $1-\delta$ that our algorithm returns a policy π such that

$$\mathrm{E}[\frac{1}{T}\sum_t R^*_w(s_t)|\pi] \geq \mathrm{E}[\frac{1}{T}\sum_t R^*_w(s_t)|\pi^*] - \epsilon.$$

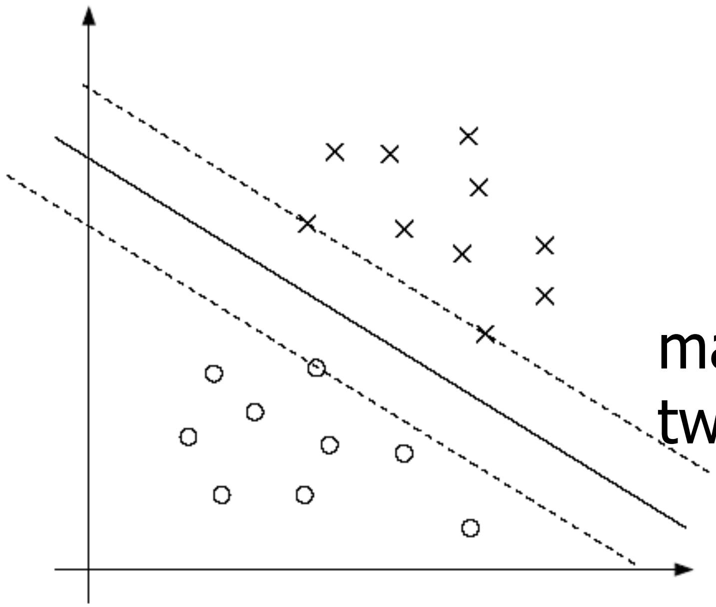
it suffices that
we run for $\frac{4n}{\epsilon^2}$ iterations,
we have $m \geq \frac{2n}{\epsilon^2}\log\frac{2n}{\delta}$ demonstrations.

- Guarantee w.r.t. unrecoverable reward function of teacher.
- Sample complexity does *not* depend on complexity of teacher's policy π^*.

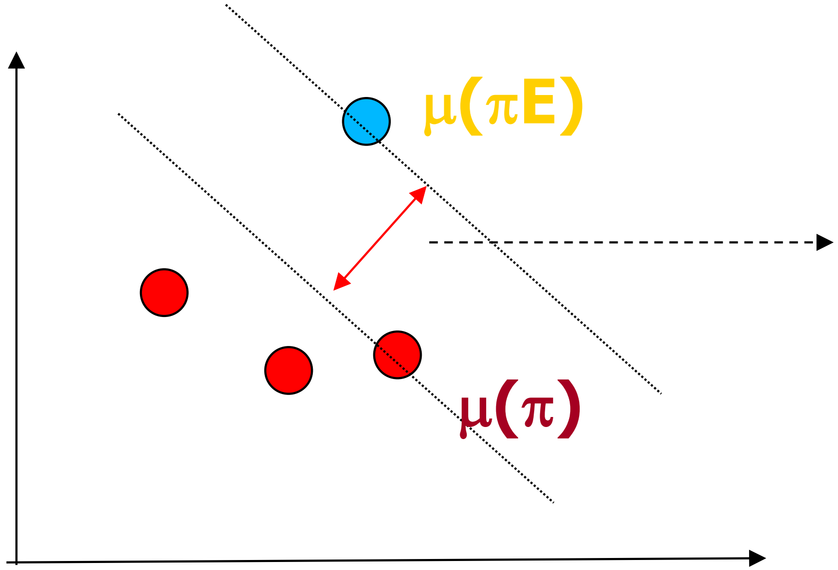
Apprenticeship learning [Abbeel & Ng, 2004]

- **For $t = 1,2,\ldots$**
  - **Inverse RL step:**
    - **Estimate expert's reward function $R(s)= w^T\phi(s)$ such that under $R(s)$ the expert performs better than all previously found policies $\{\pi_i\}$.**
  - **RL step:**
    - **Compute optimal policy π_t for the estimated reward w.**

Apprenticeship learning [Abbeel & Ng, 2004]



maximum margin hyperplane seperating two sets of points

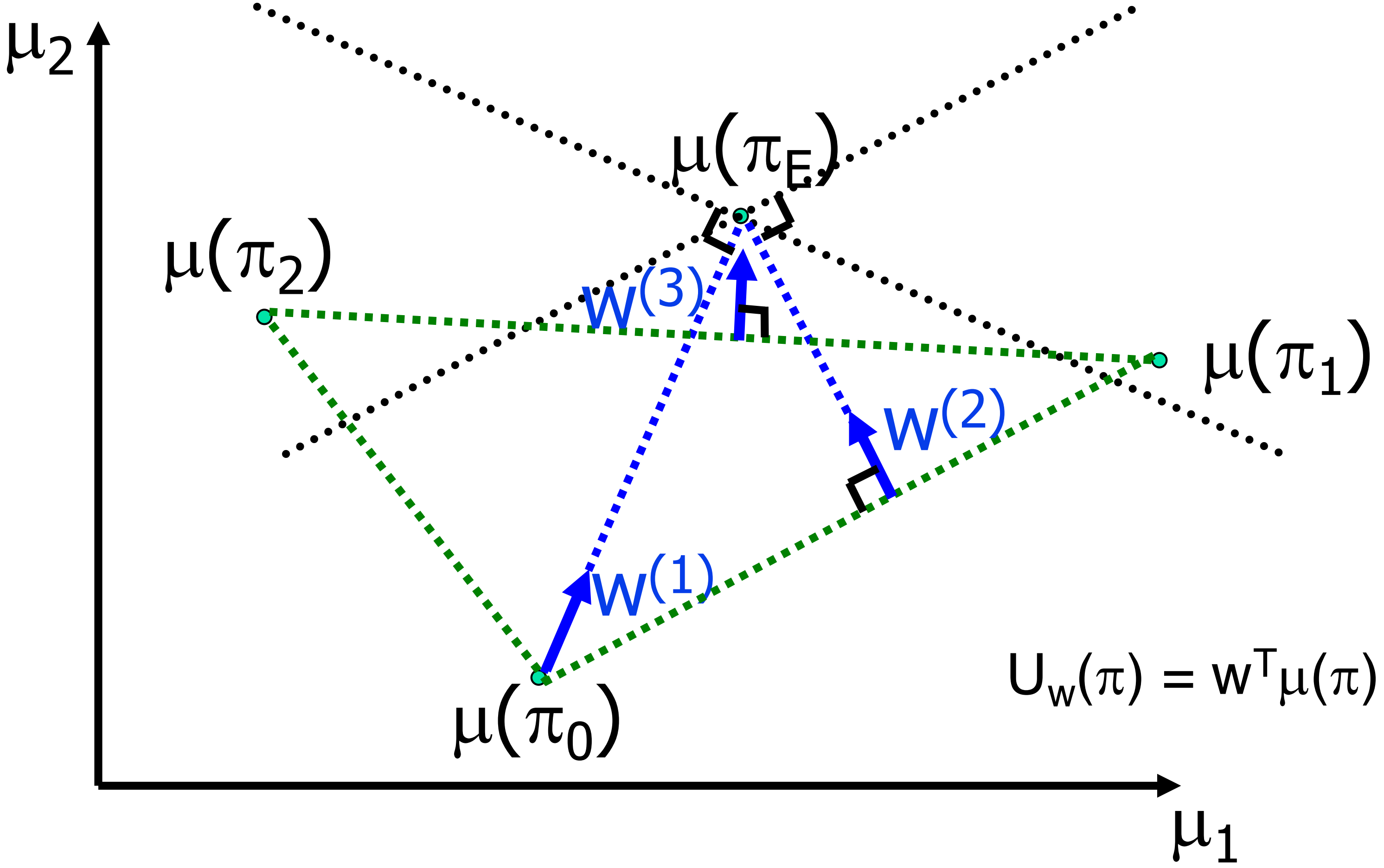


|w*T μ(πE) - w*T μ(π)|

= |Vw*(πE) - Vw*(π)|

= maximal difference between expert policy's value function and 2nd to the optimal policy's value function

Apprenticeship learning [Abbeel & Ng, 2004]



Courtesy of Pieter Abb

Gridworld Experiment

- 128 x 128 grid world divided into 64 regions, each of size 16 x 16 (“macrocells”).
- A small number of macrocells have positive rewards.
- For each macrocell, there is one feature $\Phi_i(s)$ indicating whether that state s is in macrocell i
- *Algorithm was also run on the subset of features $\Phi_i(s)$ that correspond to non-zero rewards.*

Gridworld Results

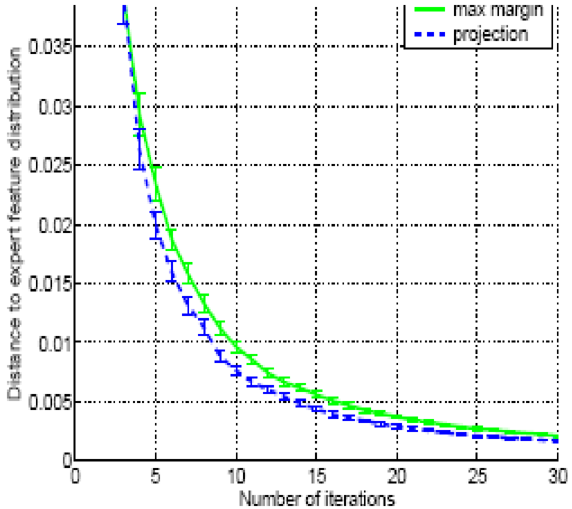
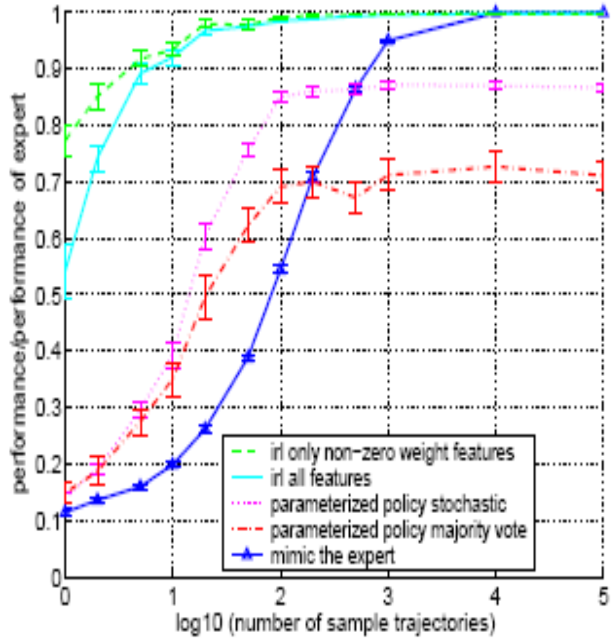


Figure 3. A comparison of the convergence speeds of the max-margin and projection versions of the algorithm on a 128x128 grid. Euclidean distance to the expert's feature expectations is plotted as a function of the number of iterations. We rescaled the feature expectations by $(1-\gamma)$ such that they are in $[0,1]^k$. The plot shows averages over 40 runs, with 1 s.e. errorbars.



Distance to expert vs. # Iterations

Performance vs. # Trajectories

Car Driving Experiment

- No explict reward function at all!
- Expert demonstrates proper policy via 2 min. of driving time on simulator (1200 data points).
- 5 different "driver types" tried.
- Features: which lane the car is in, distance to closest car in current lane.
- Algorithm run for 30 iterations, policy hand-picked.
- Movie Time! (Expert left, IRL right)

Examples

- **Learning for Control from Multiple Demonstrations** Adam Coates, Pieter Abbeel, Andrew Ng, ICML 2008
- **An Application of Reinforcement Learning to Aerobatic Helicopter Flight** Pieter Abbeel, Adam Coates, Morgan Quigley, Andrew Y. Ng, NIPS 2006
- https://www.youtube.com/watch?v=0JL04JJjocc

Examples

- **Planning-based Prediction for Pedestrians** Brian Ziebart et al., IROS 2009
- https://www.youtube.com/watch?v=hjOteEd7qwE

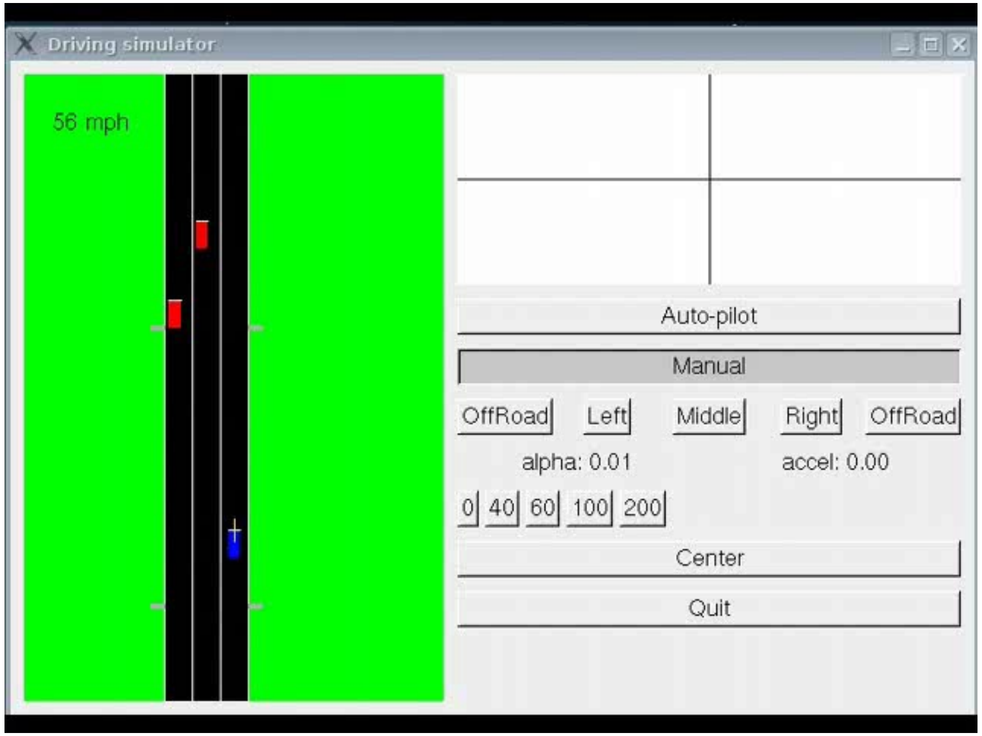
Examples

- **Data Driven Ghosting using Deep Imitation Learning** Hoang M. Le et al., SSAC 2017
- https://www.youtube.com/watch?v=WI-WL2cj0CA

Example: Highway driving

Teacher in Training World

Learned Policy in Testing World



- Input:
  - Dynamics model / Simulator $P_{sa}(s_{t+1} \mid s_t, a_t)$
  - Teacher's demonstration: 1 minute in "training world"
  - Note: R* is unknown.
  - Reward features: 5 features corresponding to lanes/shoulders; 10 features corresponding to presence of other car in current lane at different distances