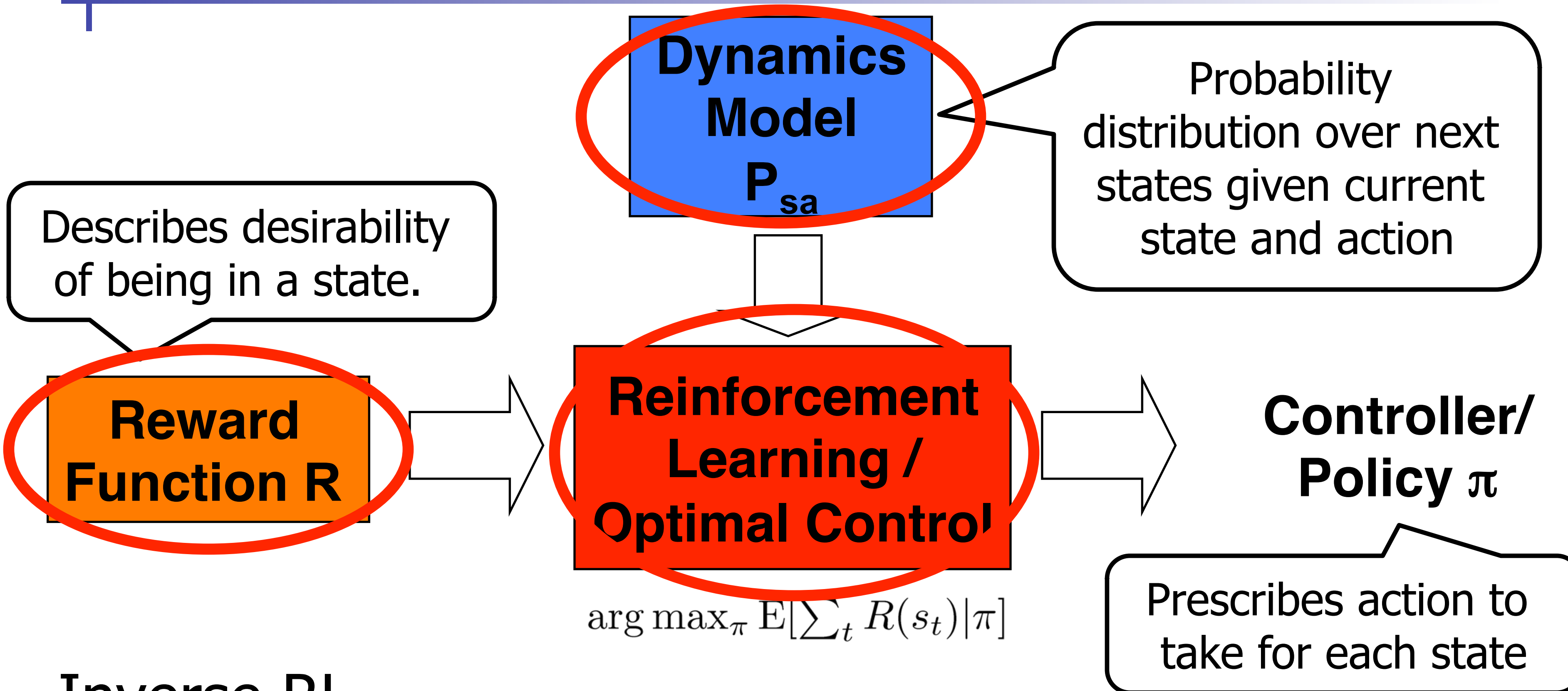
Apprenticeship Learning via Inverse Reinforcement Learning

Pieter Abbeel and Andrew Ng

Presented by Kathy Ge

Example task: driving

Big picture and key challenges



- Inverse RL
  - Can we recover R?

Overview

- Apprenticeship learning algorithms
  - Leverage expert demonstrations to learn to perform a desired task.

- Enabled us to solve highly challenging, previously unsolved, real-world control problems in
  - Quadruped locomotion
  - Simulated highway driving
  - Autonomous helicopter flight

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.

Prior work: behavioral cloning

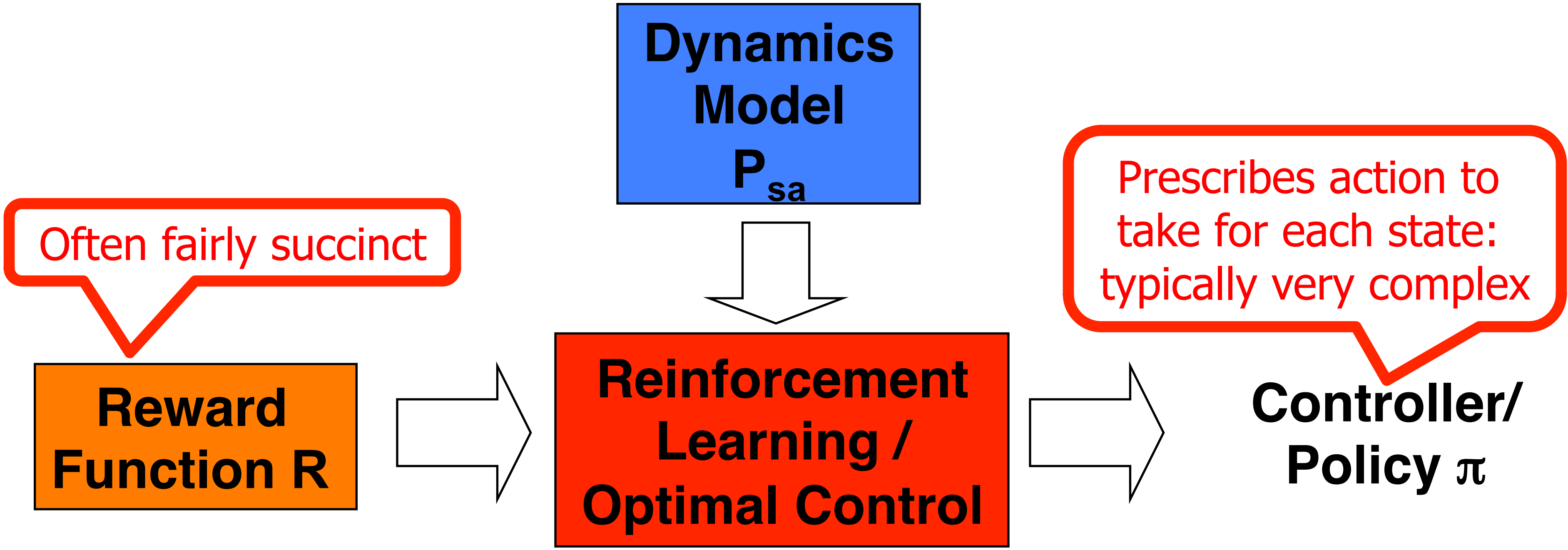
- Formulate as standard machine learning problem
  - Fix a policy class
    - E.g., support vector machine, neural network, decision tree, deep belief net, …
  - Estimate a policy from the training examples (s_0, a_0), (s_1, a_1), (s_2, a_2), …

- E.g., Pomerleau, 1989; Sammut et al., 1992; Kuniyoshi et al., 1994; Demiris & Hayes, 1994; Amit & Mataric, 2002.

Prior work: behavioral cloning

- **Limitations:**
  - Fails to provide strong performance guarantees
  - Underlying assumption: policy simplicity

Problem structure



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

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

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 LHS?
  - Assumes the expert is indeed optimal --- otherwise infeasible

Feature based reward function

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

$$\begin{aligned}
\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|\pi] &= \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t w^\top \phi(s_t)|\pi] \\
&= w^\top \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t)|\pi] \\
&= w^\top \underbrace{\mu(\pi)}_{\text{Expected cumulative discounted sum of feature values or “feature expectations”}}
\end{aligned}$$

- 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:

  Find w^* 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 $\phi : S \rightarrow \Re^n$.

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

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.

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

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

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

Formalization

- Standard max margin:

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

- "Structured prediction" max margin:

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

- 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

Formalization

- Structured prediction max margin with slack variables:

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

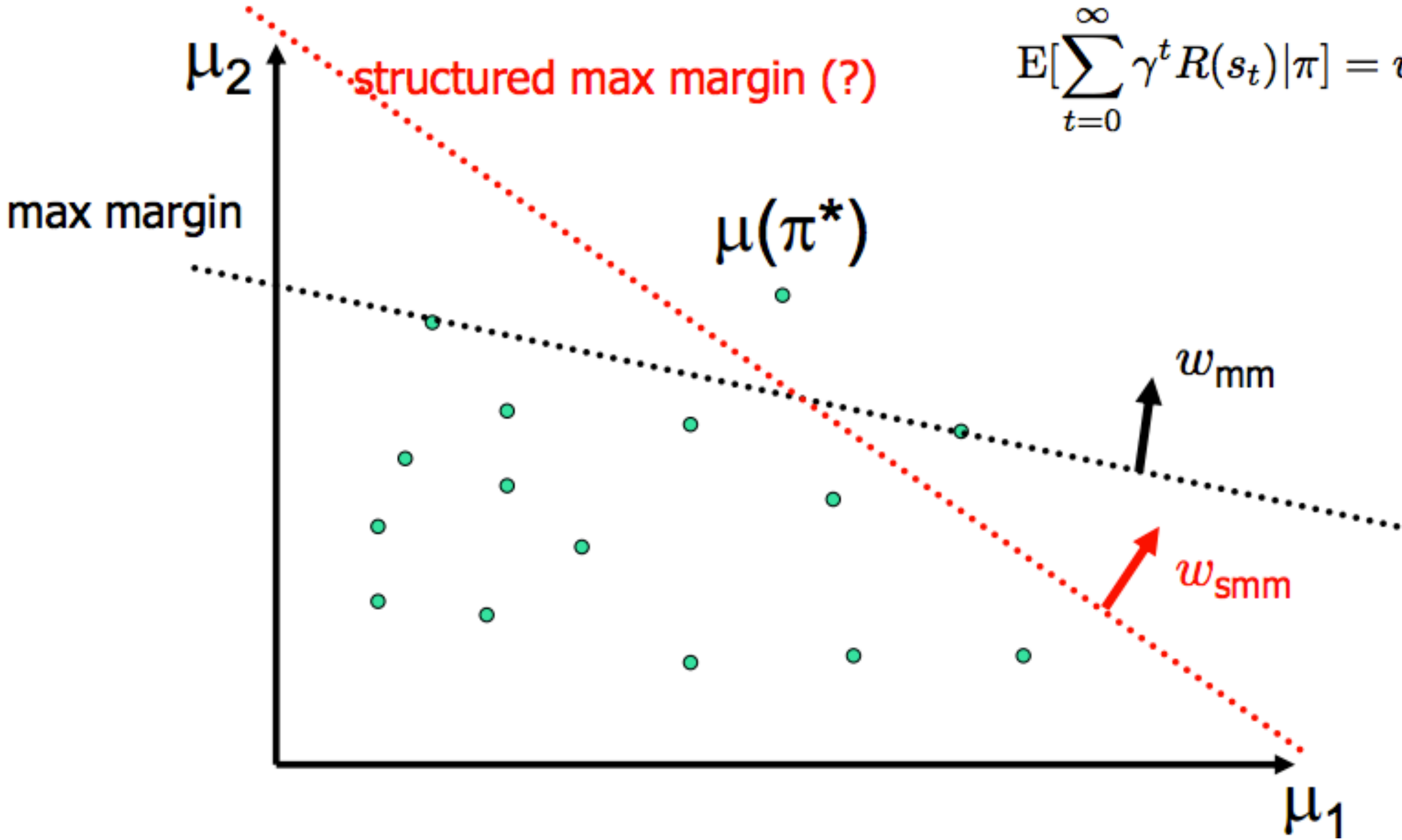
- Can be generalized to multiple MDPs (could also be same MDP with different initial state)

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

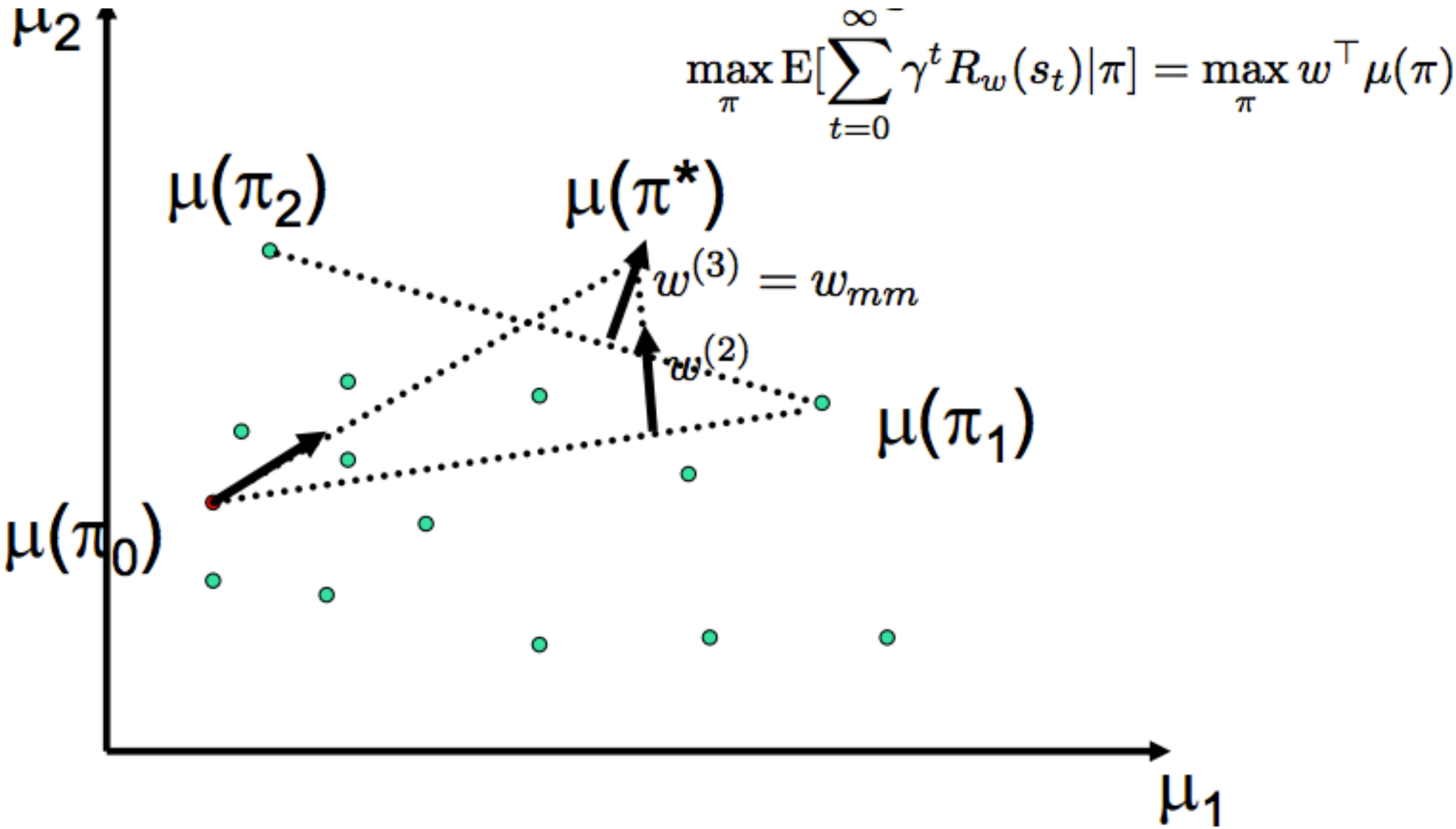
Visualization in Feature Space

- Every policy π has a corresponding feature expectation vector $\mu(\pi)$, which for visualization purposes we assume to be 2D

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



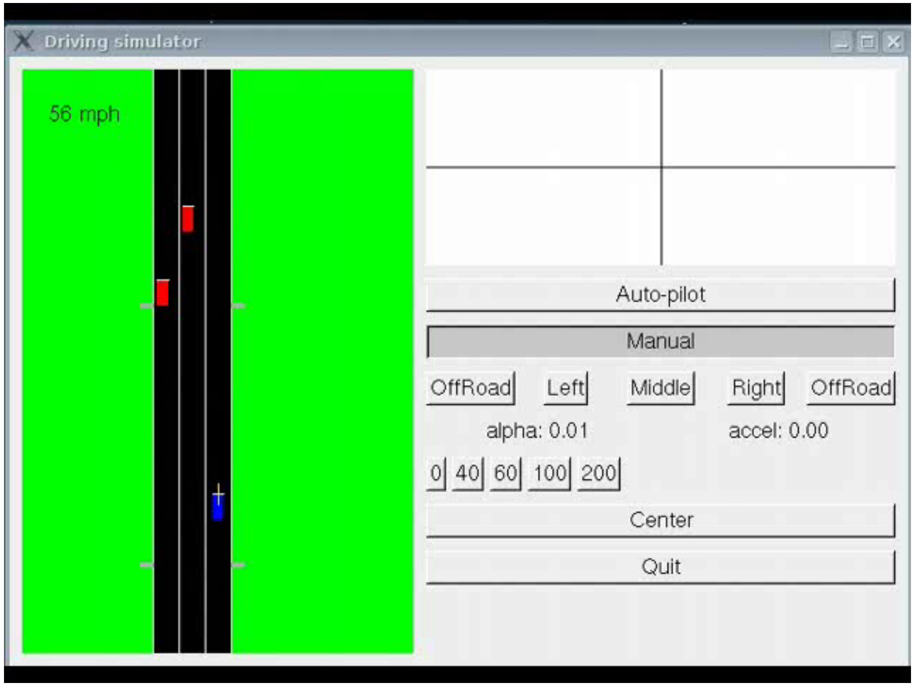
Visualization in Feature Space



Highway driving

Teacher in Training World

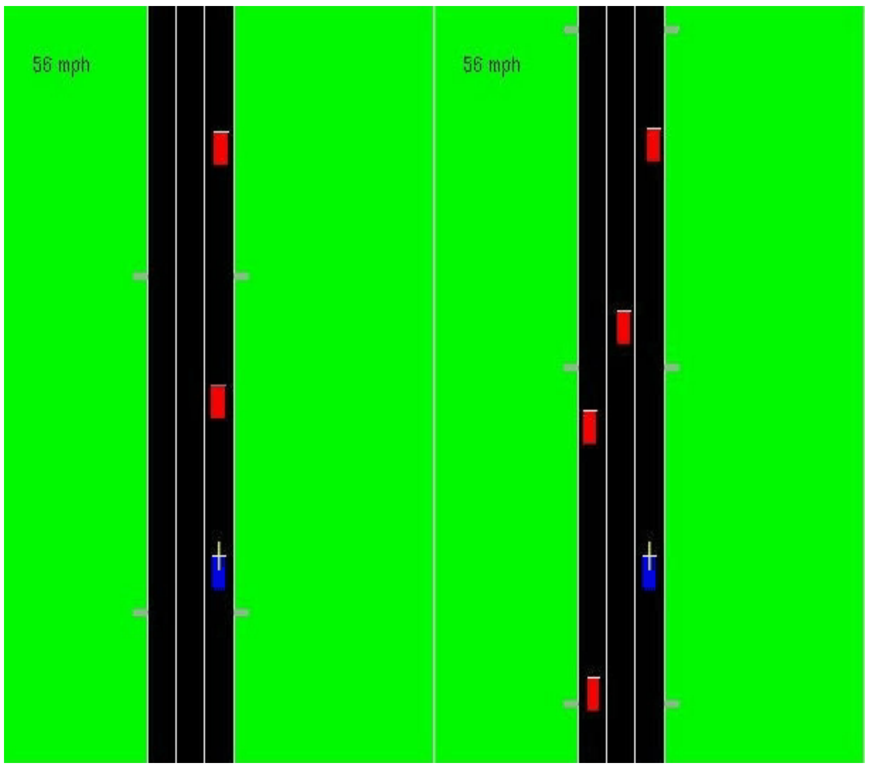
Learned Policy in Testing World



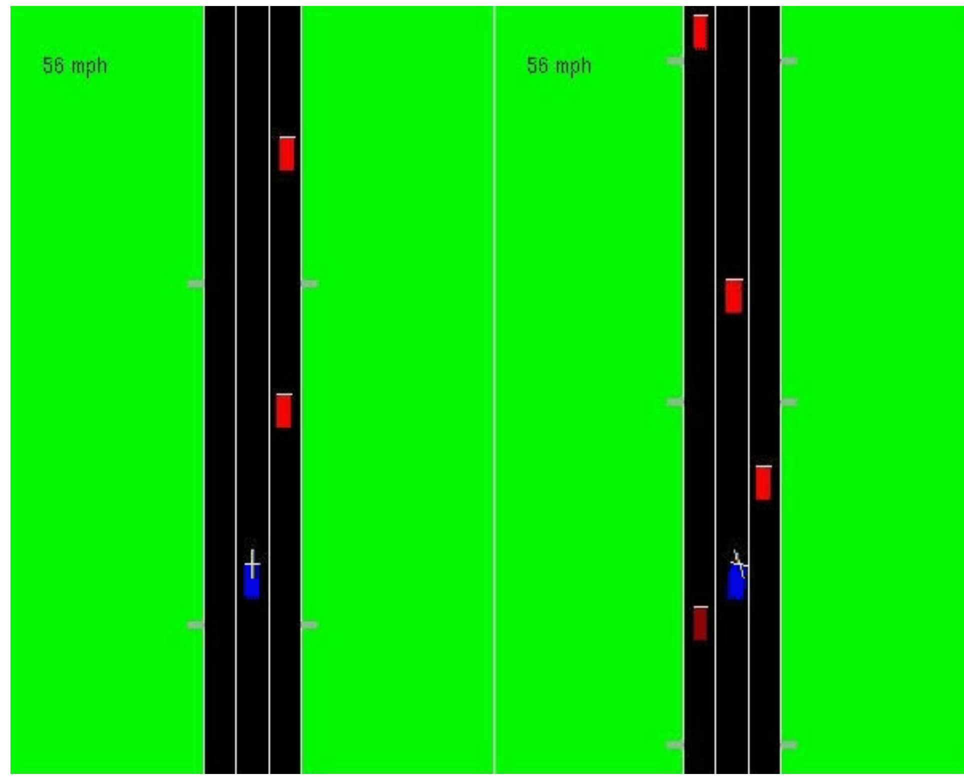
- Input:
  - Dynamics model / Simulator $P_{sa}(s_{t+1} \mid s_t, a_t)$
  - Teacher's demonstration: 1 minute in "training world"
  - Note: R* is unknown.
  - Reward features: 5 features corresponding to lanes/shoulders; 10 features corresponding to presence of other car in current lane at different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

More driving examples

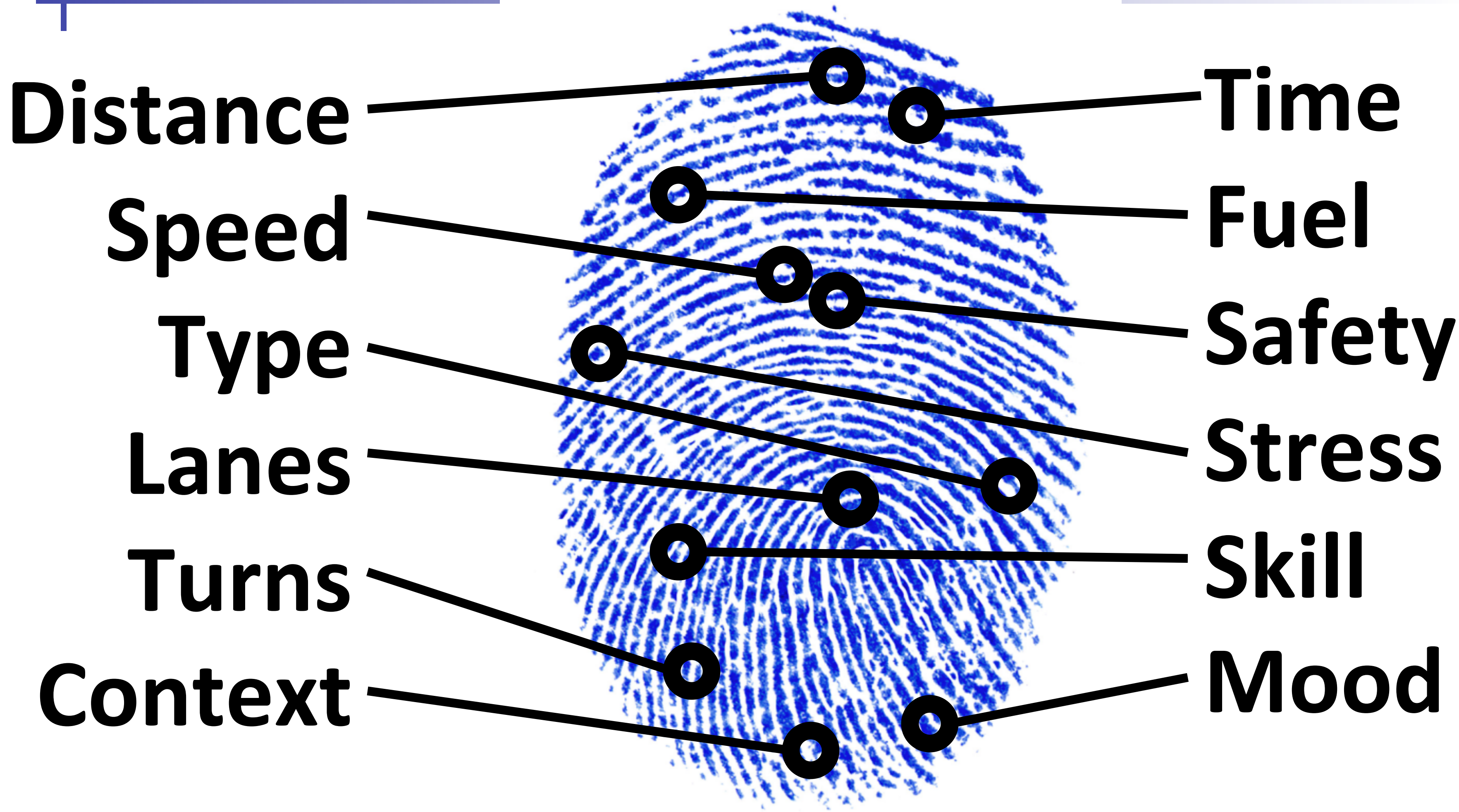
Driving demonstration | Learned behavior



Driving demonstration | Learned behavior



In each video, the left sub-panel shows a demonstration of a different driving "style", and the right sub-panel shows the behavior learned from watching the demonstration.



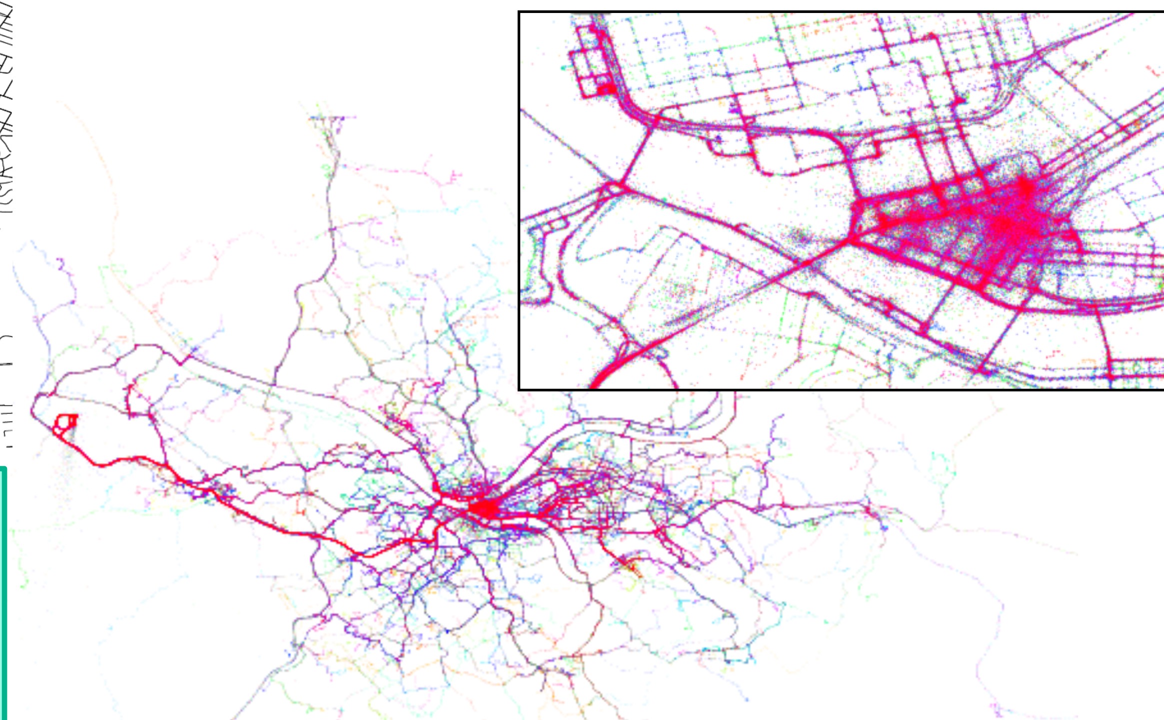
Ziebart+al, 2007/8/9

Data Collection

Length
Speed
Road
Type
Lanes

Accidents
Construction
Congestion
Time of day

25 Taxi Drivers



Over 100,000 miles

Ziebart+al, 2007/8/9

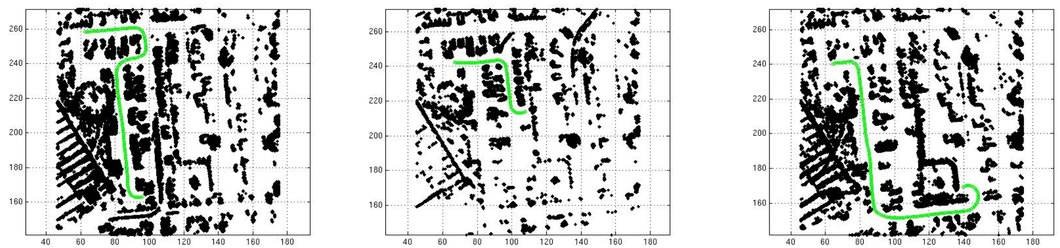
Parking lot navigation

- Reward function trades off:
  - curvature
  - smoothness,
  - distance to obstacles,
  - alignment with principal directions.

[Abbeel et al., submitted]

Experimental setup

- Demonstrate parking lot navigation on “train parking lot.”



- Run our apprenticeship learning algorithm to find a set of reward weights w.
- Receive “test parking lot” map + starting point and destination.
- Find a policy $\pi \approx \arg\max_\pi \mathrm{E}[\sum_t R_w(s_t)|\pi]$ for navigating the test parking lot.

Learned reward weights

Nice driving style

Sloppy driving-style

“Don’t mind reverse” driving-style

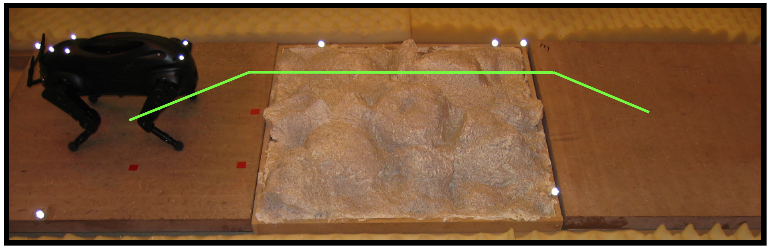
Quadruped

- Reward function trades off 25 features.

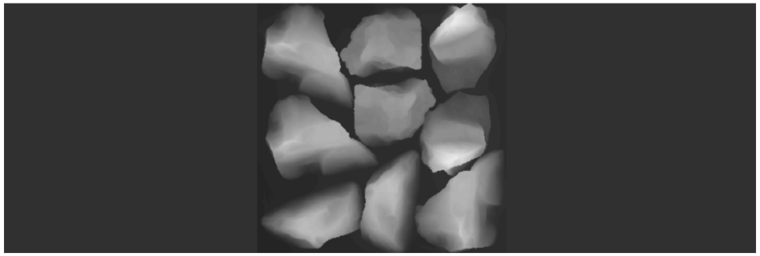
[Kolter, Abbeel & Ng, 2008]

Experimental setup

- Demonstrate path across the "training terrain"



- Run our apprenticeship learning algorithm to find a set of reward weights w.
- Receive "testing terrain"---height map.



- Find a policy $\pi \approx \arg\max_\pi \mathrm{E}[\sum_t R_w(s_t)|\pi]$ for crossing the testing terrain.

Learned reward weights

Challenging Terrain

Stairs

Teacher demonstration for quadruped

- Full teacher demonstration = sequence of footsteps.
- Much simpler to “teach hierarchically”:
  - Specify a body path.
  - Specify best footstep in a small area.

Experimental setup

- Training:
  - Have quadruped walk straight across a fairly simple board with fixed-spaced foot placements.
  - Around each foot placement: label the best foot placement. (about 20 labels)
  - Label the best body-path for the training board.
- Use our *hierarchical* inverse RL algorithm to learn a reward function from the footstep and path labels.
- Test on hold-out terrains:
  - Plan a path across the test-board.

Apprenticeship learning

Teacher's flight

$(s_0, a_0, s_1, a_1, \ldots)$

Learn
R

Reward Function R

Dynamics Model P_{sa}

Reinforcement Learning / Optimal Control

Controller π

Chaos

Flips

Nose-in funnel

Tail-in funnel

Thank you.