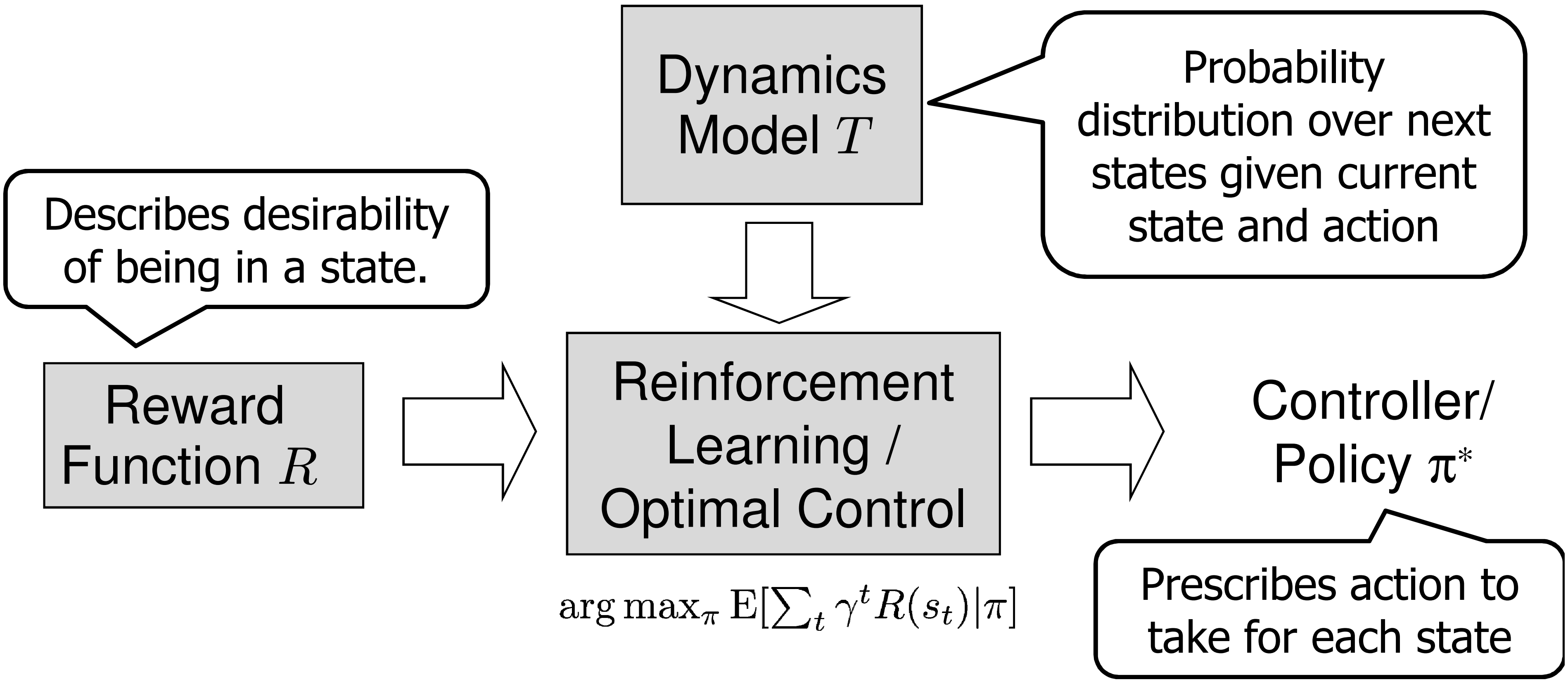
Inverse Reinforcement Learning

Pieter Abbeel
UC Berkeley EECS

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

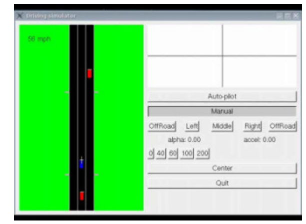
Lecture outline

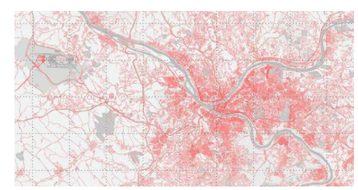
- Examples of apprenticeship learning via inverse RL
- Inverse RL vs. behavioral cloning
- Historical sketch of inverse RL
- Mathematical formulations for inverse RL
- Case studies

- Trajectory-based reward, with application to autonomous helicopter flight

Examples

- Simulated highway driving
  - Abbeel and Ng, ICML 2004,
  - Syed and Schapire, NIPS 2007
- Aerial imagery based navigation
  - Ratliff, Bagnell and Zinkevich, ICML 2006
- Parking lot navigation
  - Abbeel, Dolgov, Ng and Thrun, IROS 2008
- Urban navigation
  - Ziebart, Maas, Bagnell and Dey, AAAI 2008
- Quadruped locomotion
  - Ratliff, Bradley, Bagnell and Chestnutt, NIPS 2007
  - Kolter, Abbeel and Ng, NIPS 2008




Urban navigation

- Reward function for urban navigation?

→ destination prediction

Ziebart, Maas, Bagnell and Dey AAAI 2008

Lecture outline

- Examples of apprenticeship learning via inverse RL
- *Inverse RL vs. behavioral cloning*
- Historical sketch of inverse RL
- Mathematical formulations for inverse RL
- Case studies

- Trajectory-based reward, with application to autonomous helicopter flight

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.

Lecture outline

- Examples of apprenticeship learning via inverse RL
- Inverse RL vs. behavioral cloning
- *Historical sketch of inverse RL*
- Mathematical formulations for inverse RL
- Case studies

- Trajectory-based reward, with application to autonomous helicopter flight

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

Inverse RL history

- 2007, Ratliff+al: max margin with boosting---enables large vocabulary of reward features
- 2007, Ramachandran and Amir, and Neu and Szepesvari: reward function as characterization of policy class
- 2008, Kolter, Abbeel and Ng: hierarchical max-margin
- 2008, Syed and Schapire: feature matching + game theoretic formulation
- 2008, Ziebart+al: feature matching + max entropy
- 2008, Abbeel+al: feature matching -- application to learning parking lot navigation style
- Active inverse RL? Inverse RL w.r.t. minmax control, partial observability, learning stage (rather than observing optimal policy), … ?

Lecture outline

- Examples of apprenticeship learning via inverse RL
- Inverse RL vs. behavioral cloning
- Historical sketch of inverse RL
- *Mathematical formulations for inverse RL*
- Case studies

- Trajectory-based reward, with application to autonomous helicopter flight

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 LHS?
  - 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 $\phi : S \rightarrow \Re^n$.

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

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:

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

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

Recap of challenges

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

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

- Challenges:
  - Assumes we know the entire expert policy π* → assumes we can estimate expert feature expectations
  - R=0 is a solution (now: w=0), more generally: reward function ambiguity
  - Assumes the expert is indeed optimal---became even more of an issue with the more limited reward function expressiveness!
  - Computationally: assumes we can enumerate all policies

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:

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

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

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

Complete max-margin formulation

$$\min_w \|w\|_2^2 + C \sum_i \xi^{(i)}$$

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

[Ratliff, Zinkevich and Bagnell, 2006]

- Resolved: access to π*, ambiguity, expert suboptimality
- One challenge remains: very large number of constraints
  - Ratliff+al use subgradient methods.
  - In this lecture: constraint generation

Constraint generation

Initialize $\Pi^{(i)}$ = {} for all i and then iterate

- Solve

$$\min_w \|w\|_2^2 + C\sum_i \xi^{(i)}$$

$$\text{s.t.} \quad w^\top \mu(\pi^{(i)*}) \geq w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)}) - \xi^{(i)} \qquad \forall i, \forall \pi^{(i)} \in \Pi^{(i)}$$

- For current value of w, find the most violated constraint for all i by solving:

$$\max_{\pi^{(i)}} w^\top \mu(\pi^{(i)}) + m(\pi^{(i)*}, \pi^{(i)})$$

= find the optimal policy for the current estimate of the reward function (+ loss augmentation m)

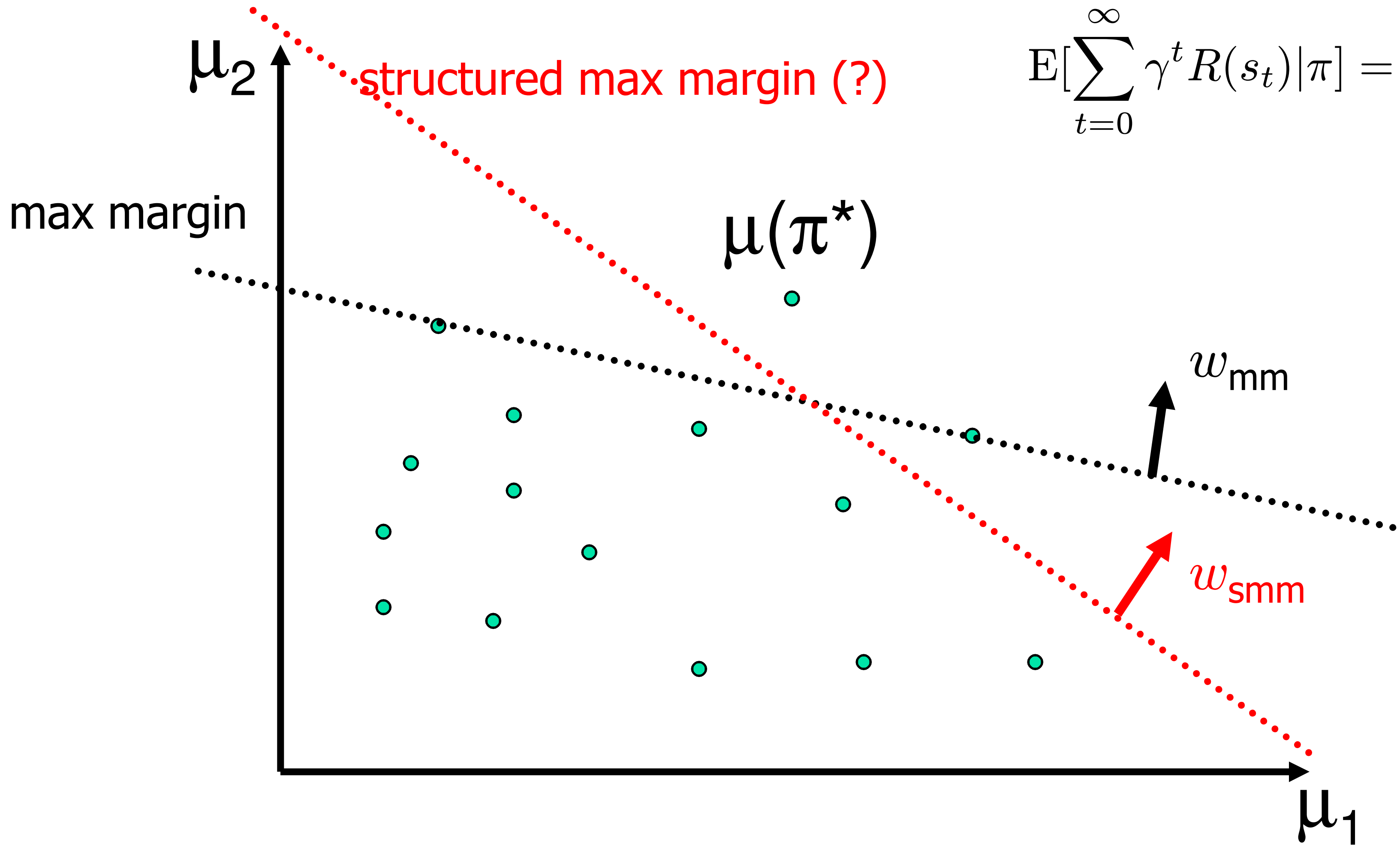
- For all i add $\pi^{(i)}$ to $\Pi^{(i)}$

- If no constraint violations were found, we are done.

Visualization in feature expectation space

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

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

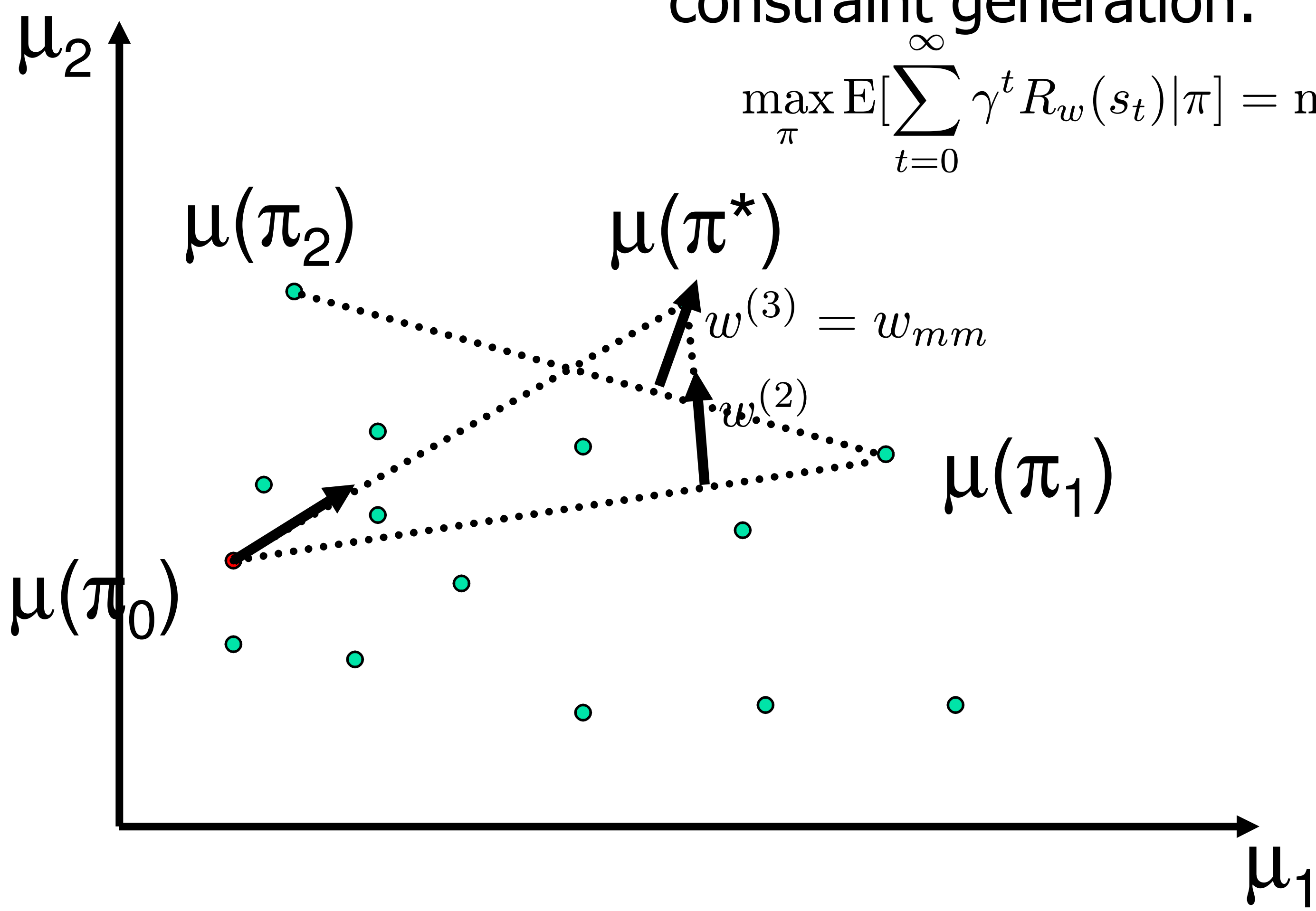


Constraint generation

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

constraint generation:

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



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)

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:

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

Holder's inequality:

$$\forall p, q \geq 1 \text{ if } \frac{1}{p} + \frac{1}{q} = 1, \text{ then } x^\top y \leq \|x\|_p \|y\|_q$$

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

$$\mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R_{w^*}(s_t)|\pi] = w^{*\top} \mu(\pi) \geq w^{*\top} \mu(\pi^*) - \epsilon = \mathrm{E}[\sum_{t=0}^{\infty} \gamma^t R_{w^*}(s_t)|\pi^*] - \epsilon$$

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.

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

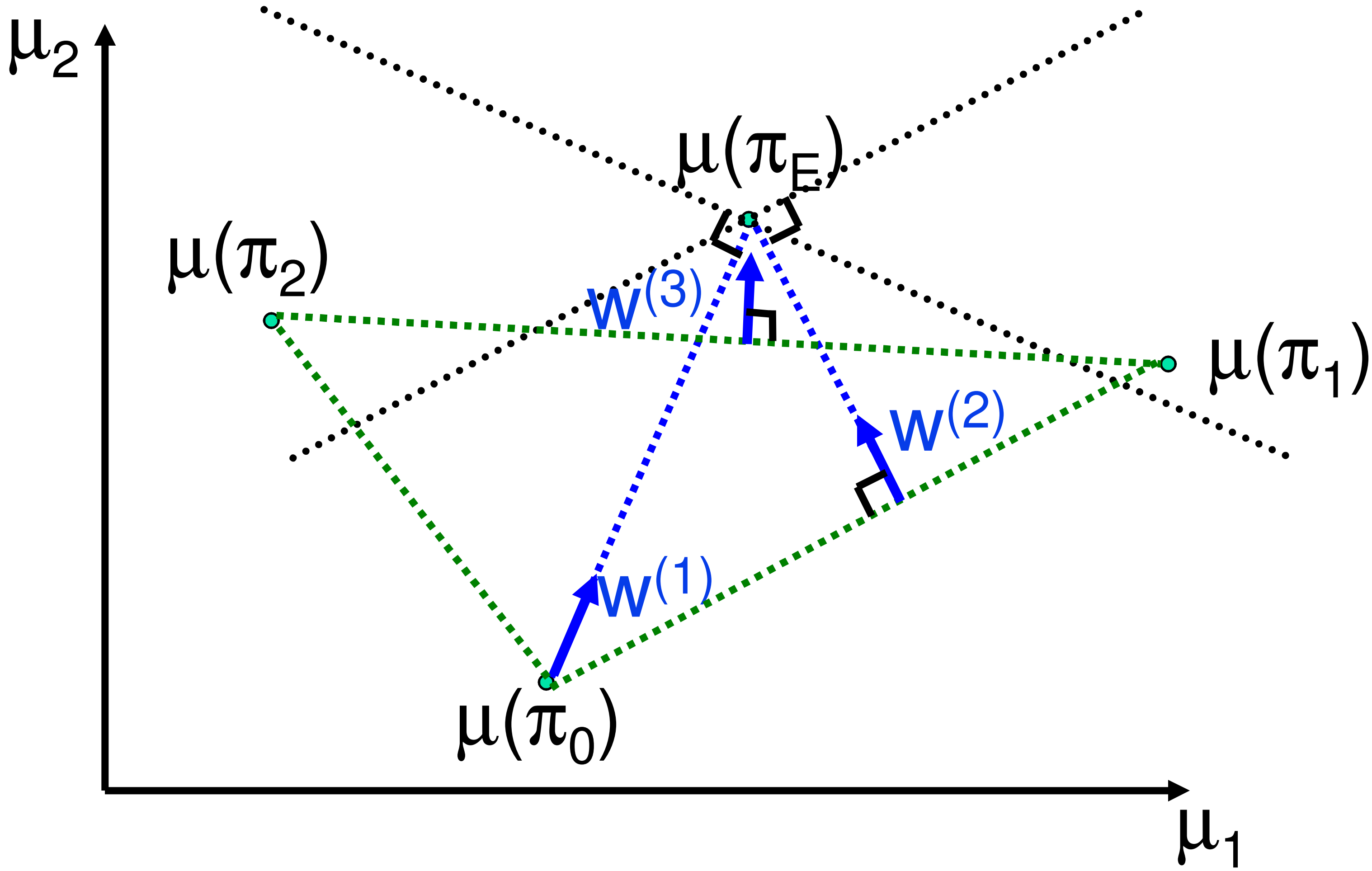
$$s.t. \quad \mathsf{E}[\sum_{t=0}^{T} R_w(s_t)|\pi^*] \geq \mathsf{E}[\sum_{t=0}^{T} R_w(s_t)|\pi] + \gamma \quad \forall \pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$

  - **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.

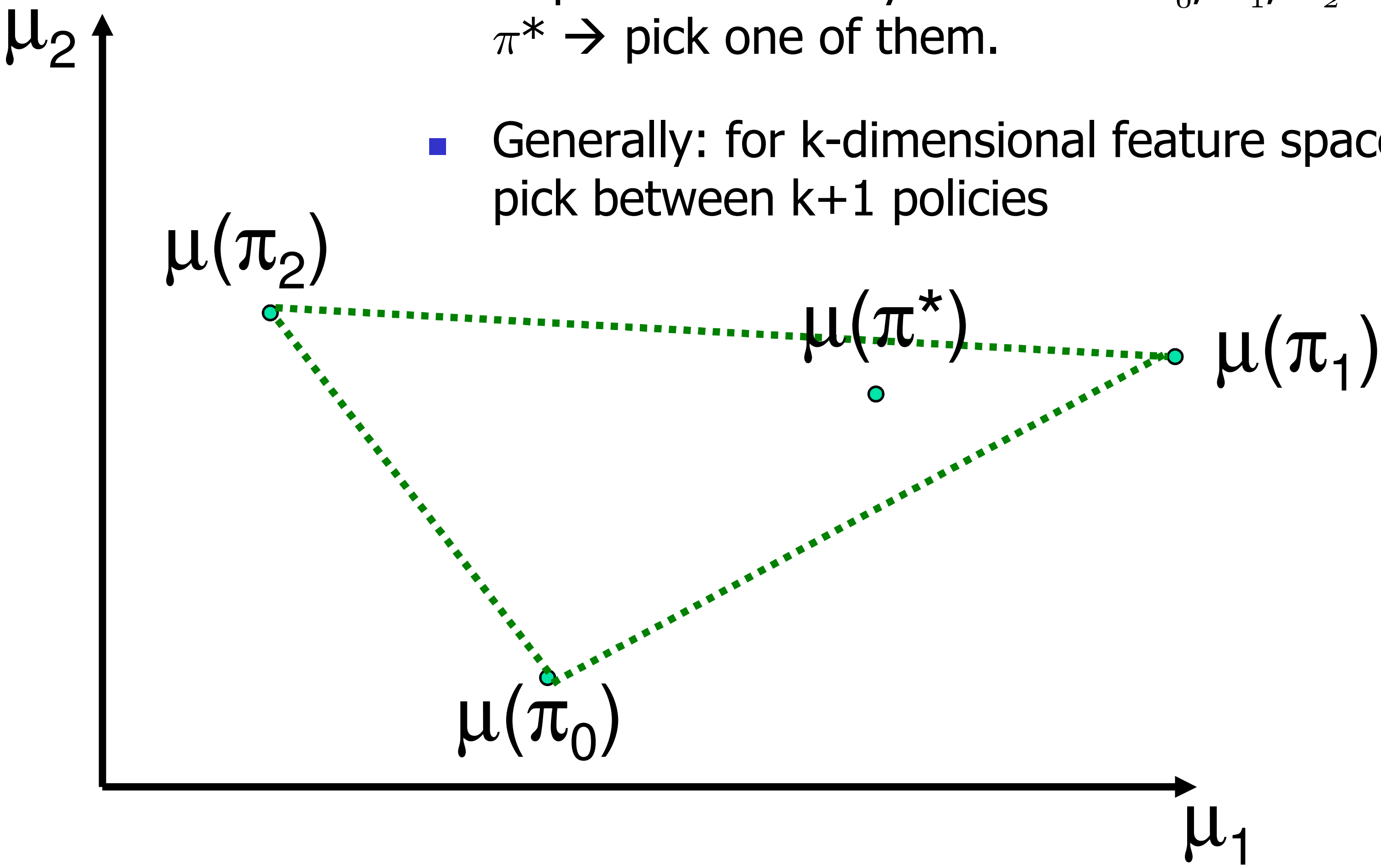
There is no reward function for which the teacher significantly outperforms thus-far found policies.

Algorithm example run



Suboptimal expert case

- Can match expert by stochastically mixing between 3 policies
- In practice: for any w^* one of π_0, π_1, π_2 outperforms π^* → pick one of them.
- Generally: for k-dimensional feature space left to pick between k+1 policies



Feature expectation matching

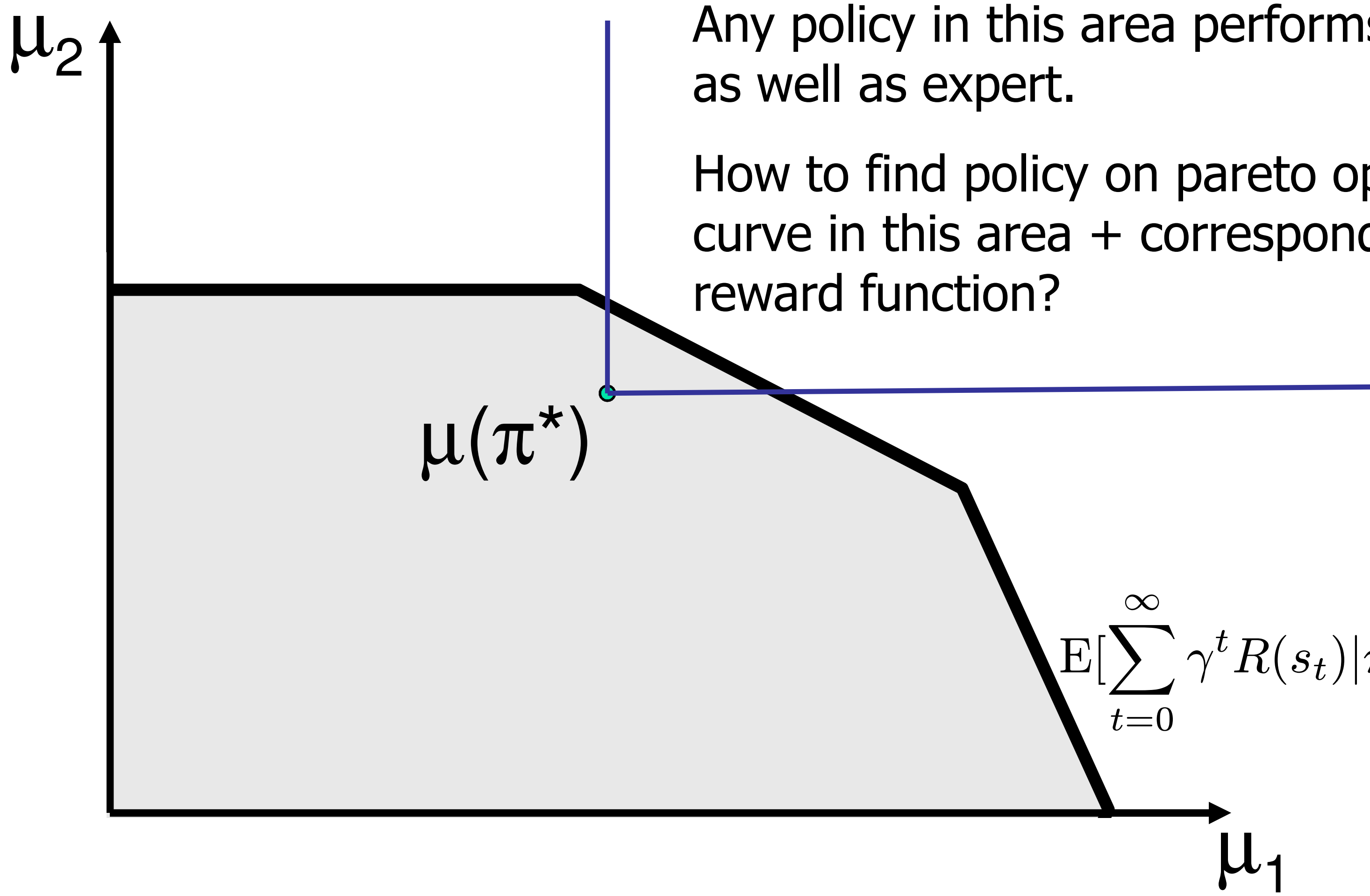
- If expert suboptimal then the resulting policy is a mixture of somewhat arbitrary policies which have expert in their convex hull.

- In practice: pick the best one of this set and pick the corresponding reward function.

- Next:
  - Syed and Schapire, 2008.
  - Ziebart+al, 2008.

Min-Max feature expectation matching
Syed and Schapire (2008)

Additional assumption: $w \geq 0$, $\sum_i w_i = 1$.



Any policy in this area performs at least as well as expert.

How to find policy on pareto optimal curve in this area + corresponding reward function?

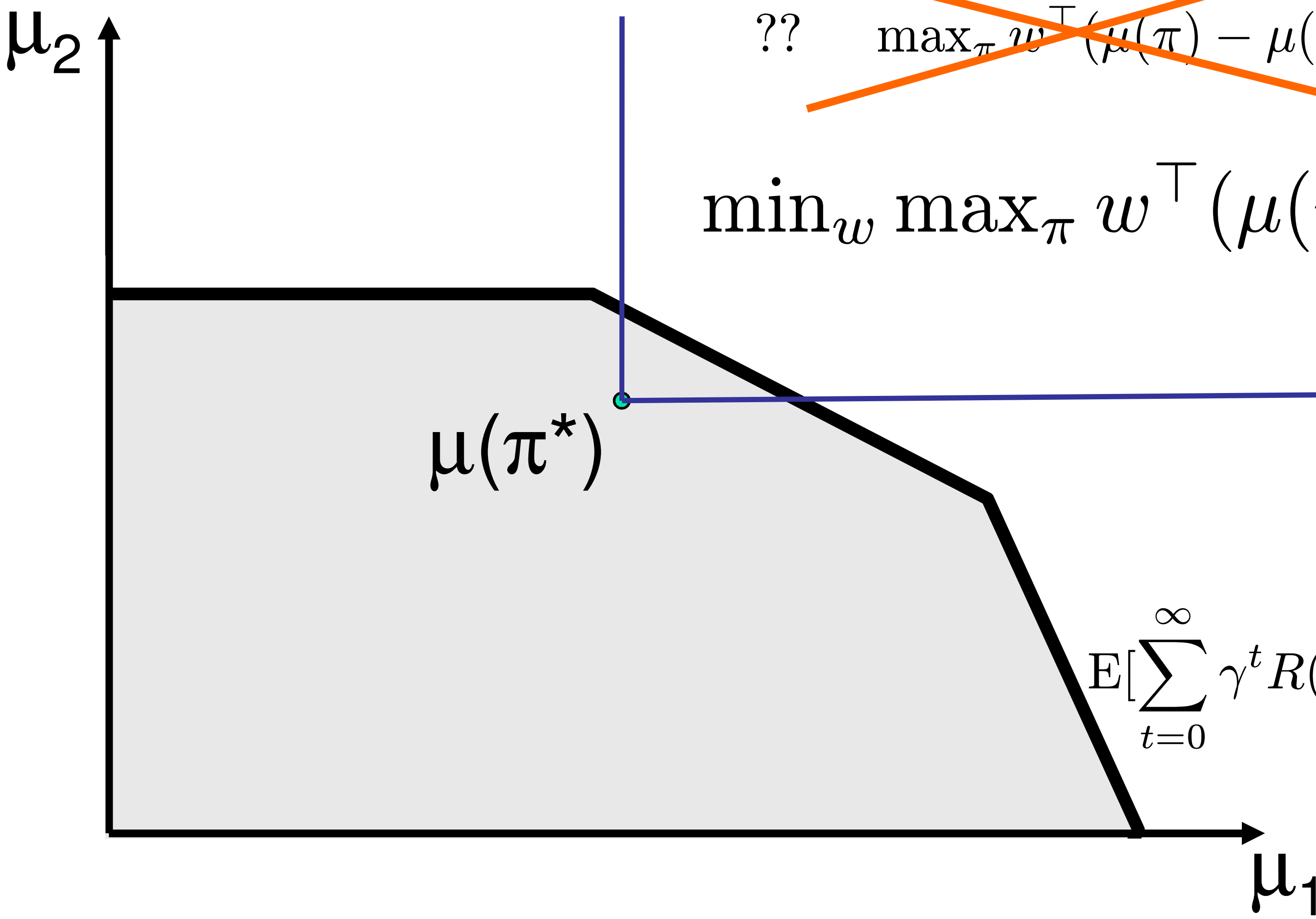
$$\mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi\right] = w^\top \mu(\pi)$$

Min-Max feature expectation matching Syed and Schapire (2008)

Additional assumption: $w \geq 0$, $\sum_i w_i = 1$.

?? $\cancel{\max_\pi w^\top(\mu(\pi) - \mu(\pi^*))}$

$$\min_w \max_\pi w^\top(\mu(\pi) - \mu(\pi^*))$$



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

Min max games

- Example of standard min-max game setting:

  rock-paper-scissors pay-off matrix:

		maximizer		
		rock	paper	scissors
minimizer	rock	0	1	-1
	paper	-1	0	1
	scissors	1	-1	0

pay-off matrix G

$$\boxed{\min_{w_m : w_m \geq 0, \|w_m\|_1 = 1} \; \max_{w_M : w_M \geq 0, \|w_M\|_1 = 1} \; w_m^\top G w_M}$$

Nash equilibrium solution is mixed strategy: (1/3,1/3,1/3) for both players

Min-Max feature expectation matching Syed and Schapire (2008)

- Standard min-max game:

$$\min_{w_m : w_m \geq 0, \|w_m\|_1 = 1} \; \max_{w_M : w_M \geq 0, \|w_M\|_1 = 1} w_m^\top G w_M$$

- Min-max inverse RL:

$$\min_{w : \|w\|_1 = 1, w \geq 0} \max_{\pi} w^\top (\mu(\pi) - \mu(\pi^*))$$

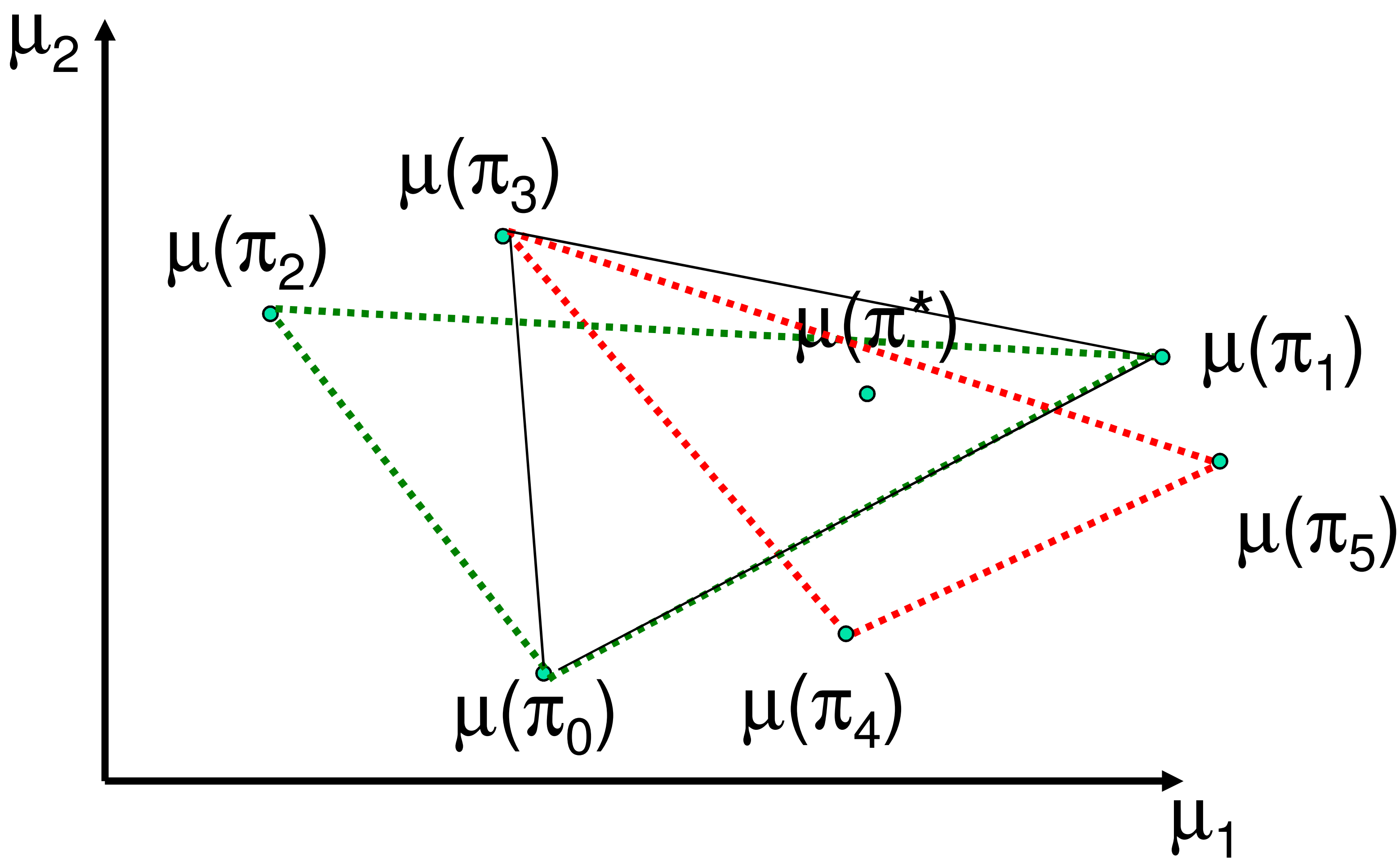
- Solution: maximize over weights λ which weigh the contribution of all policies π_1, π_2, …, π_N to the mixed policy.
- Formally:

$$\min_{w} \max_{\lambda} w^\top G \lambda \qquad\qquad G_{ij} = (\mu(\pi_j) - \mu(\pi^*))_i$$

- Remaining challenge: G very large! See paper for algorithm that only uses relevant parts of G. [Strong similarity with constraint generation schemes we have seen.]

Maximum-entropy feature expectation matching --- Ziebart+al, 2008

- Recall feature matching in suboptimal expert case:



Maximum-entropy feature expectation matching --- Ziebart+al, 2008

- Maximize entropy of distributions over paths followed while satisfying the constraint of feature expectation matching:

$$\max_P \quad -\sum_{\zeta} P(\zeta) \log P(\zeta)$$
$$\text{s.t.} \quad \sum_{\eta} P(\zeta)\mu(\zeta) = \mu(\pi^*)$$

- This turns out to imply that P is of the form:

$$P(\zeta) = \frac{1}{Z(w)} \exp(w^\top \phi(\zeta))$$

- See paper for algorithmic details.

Feature expectation matching

- If expert suboptimal:
  - Abbeel and Ng, 2004: resulting policy is a mixture of policies which have expert in their convex hull---In practice: pick the best one of this set and pick the corresponding reward function.
  - Syed and Schapire, 2008 recast the same problem in game theoretic form which, at cost of adding in some prior knowledge, results in having a unique solution for policy and reward function.
  - Ziebart+al, 2008 assume the expert stochastically chooses between paths where each path’s log probability is given by its expected sum of rewards.

Lecture outline

- Examples of apprenticeship learning via inverse RL
- Inverse RL vs. behavioral cloning
- Historical sketch of inverse RL
- Mathematical formulations for inverse RL
  - Max-margin
  - Feature matching
  - *Reward function parameterizing the policy class*
- Case studies
- Trajectory-based reward, with application to autonomous helicopter flight

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

- 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:

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

- Ramachandran and Amir, AAAI2007: MCMC method to sample from this distribution

- Neu and Szepesvari, UAI2007: gradient method to optimize the likelihood

Lecture outline

- Examples of apprenticeship learning via inverse RL
- Inverse RL vs. behavioral cloning
- History of inverse RL
- Mathematical formulations for inverse RL
- *Case studies: (1) Highway driving, (2) Crusher, (3) Parking lot navigation, (4) Route inference, (5) Quadruped locomotion*
- Trajectory-based reward, with application to autonomous helicopter flight

Simulated highway driving

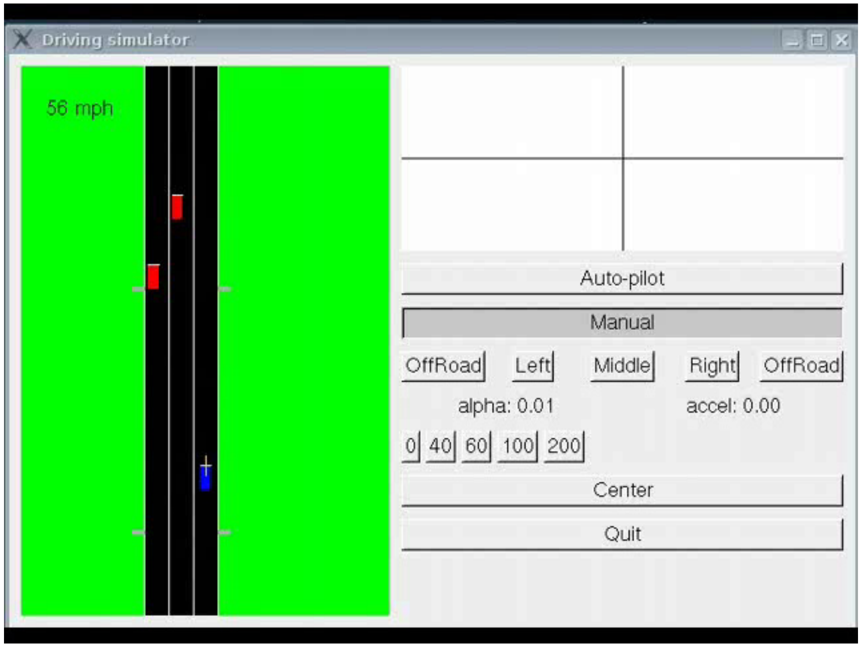
Abbeel and Ng, ICML 2004; Syed and Schapire, NIPS 2007

Highway driving

[Abbeel and Ng 2004]

Teacher in Training World

Learned Policy in Testing World

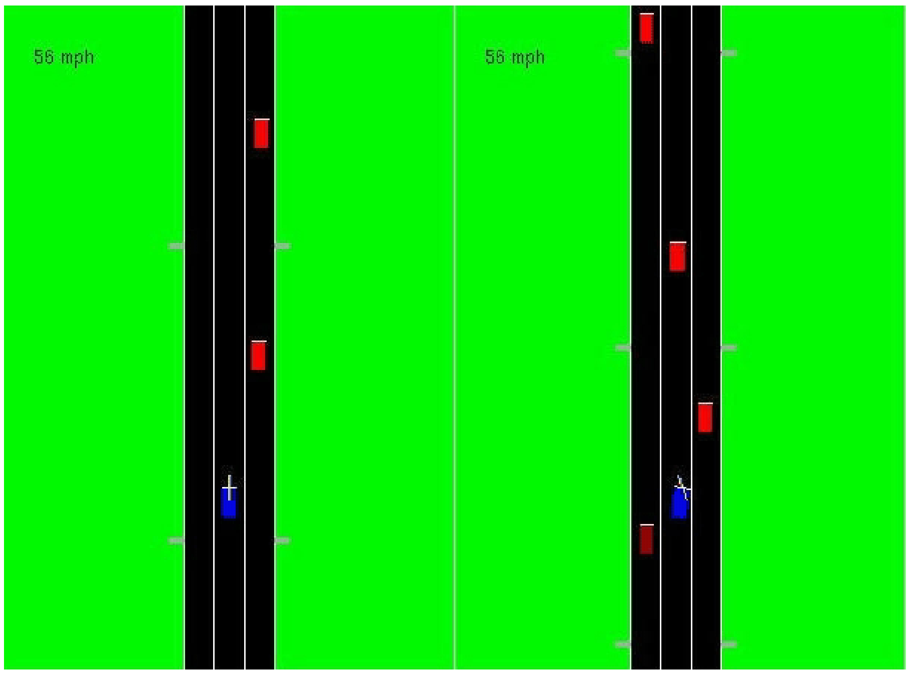


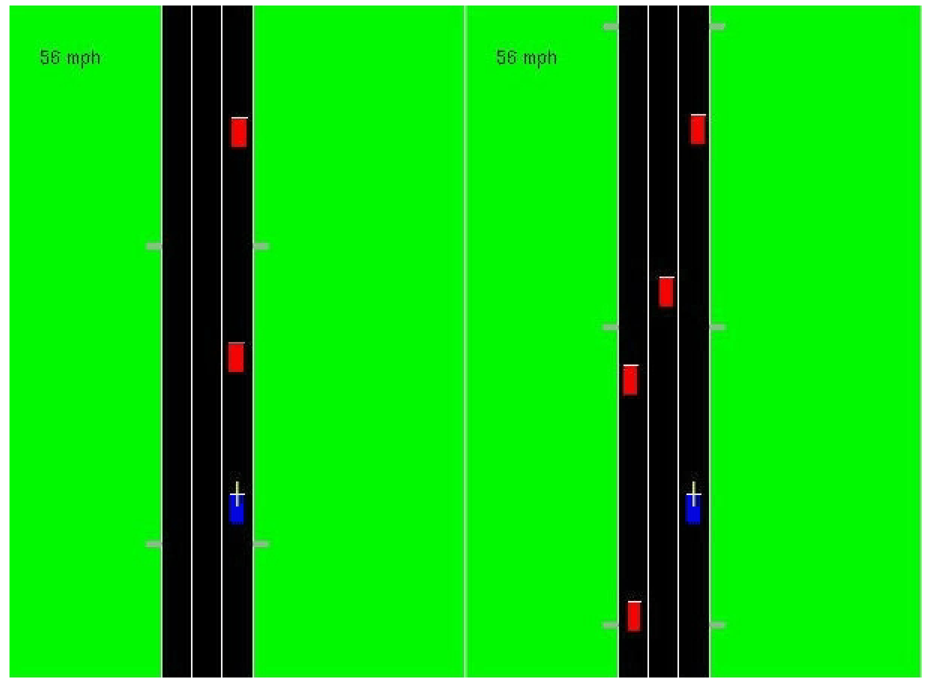
- Input:
  - Dynamics model / Simulator $P_{sa}(s_{t+1} \mid s_t, a_t)$
  - Teacher's demonstration: 1 minute in "training world"
  - Note: R* is unknown.
  - Reward features: 5 features corresponding to lanes/shoulders; 10 features corresponding to presence of other car in current lane at different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

More driving examples

[Abbeel and Ng 2004]

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.

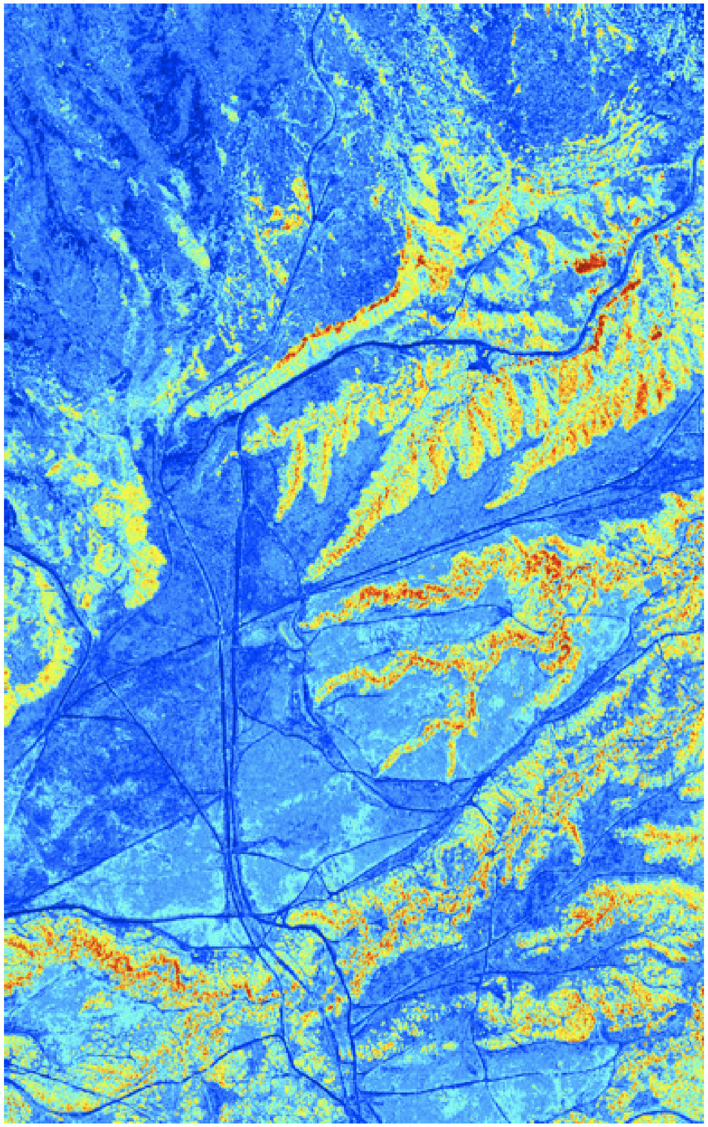

RSS 2008: Dave Silver and Drew Bagnell

[redacted]

example path

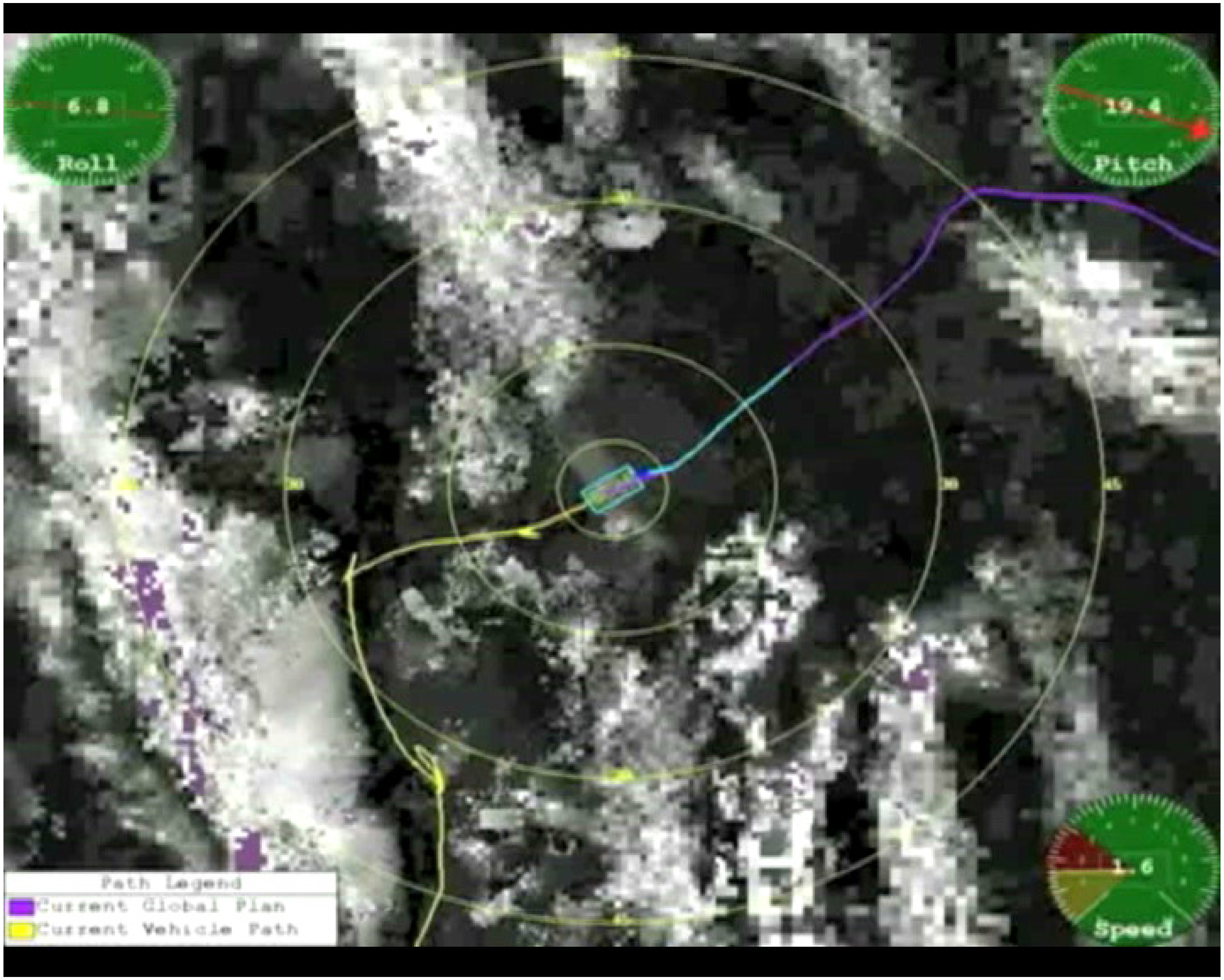
[redacted]

Max margin



[Ratliff + al, 2006/7/8]

Max-margin



[Ratliff + al, 2006/7/8]

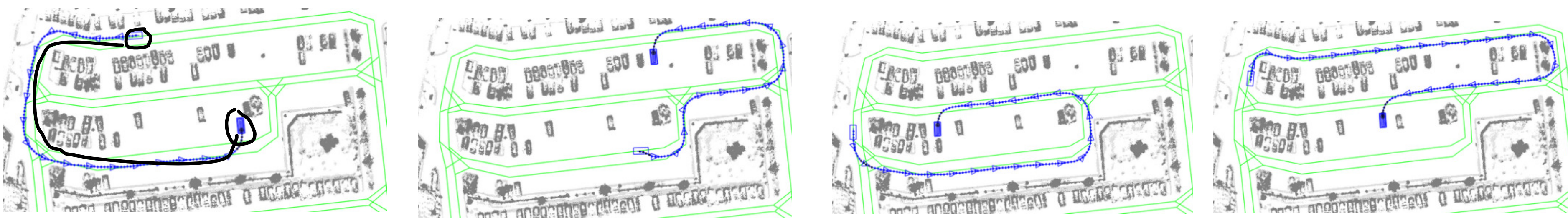
Parking lot navigation

- 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.”



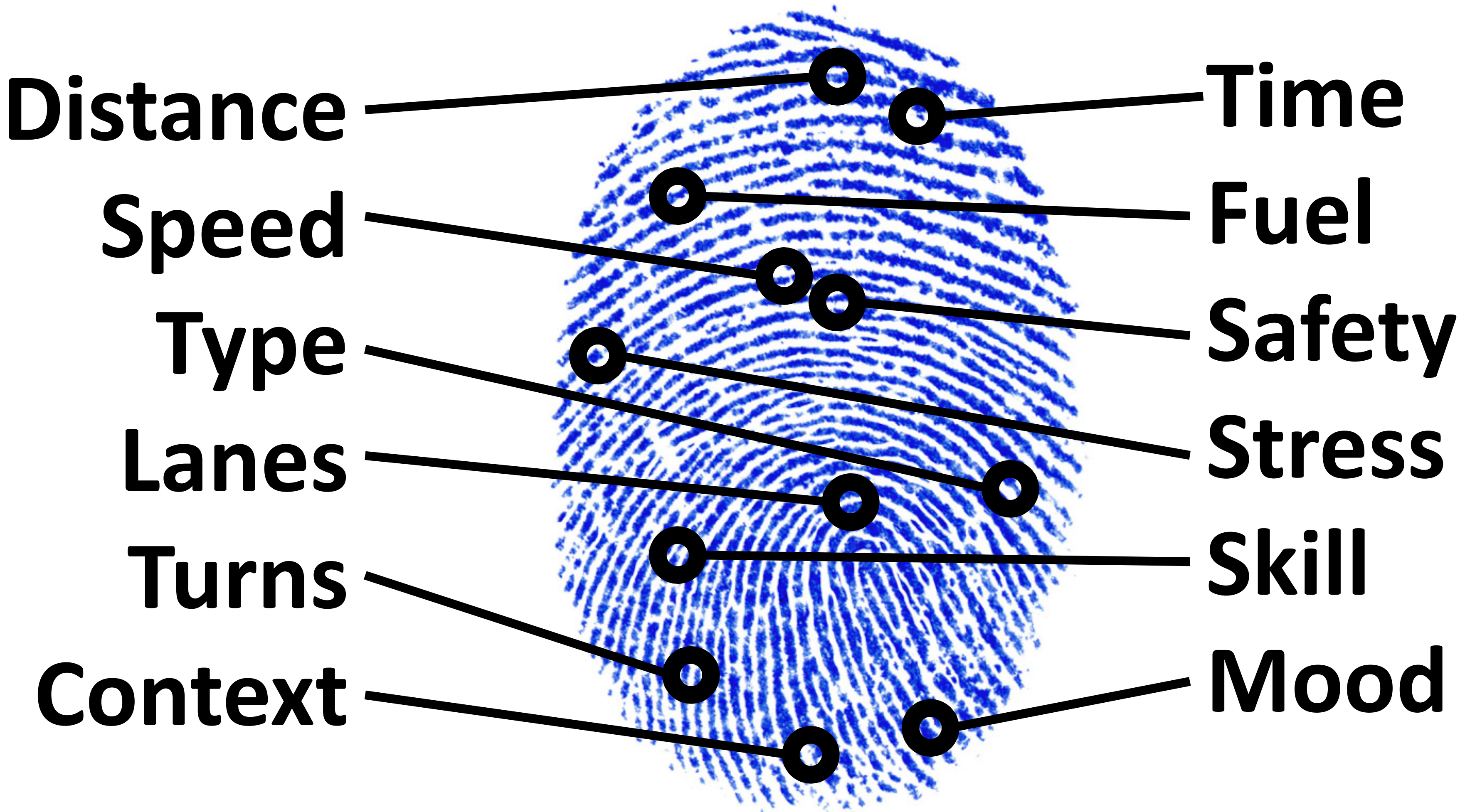
- Run our apprenticeship learning algorithm to find the reward function.
- Receive “test parking lot” map + starting point and destination.
- Find the trajectory that maximizes the *learned reward function* for navigating the test parking lot.

Nice driving style

Sloppy driving-style

Only 35% of routes are "fastest" (Letchner, Krumm, & Horvitz 2006)


Time
Money
Stress
Skill
Ziebart+al, 2007/8/9



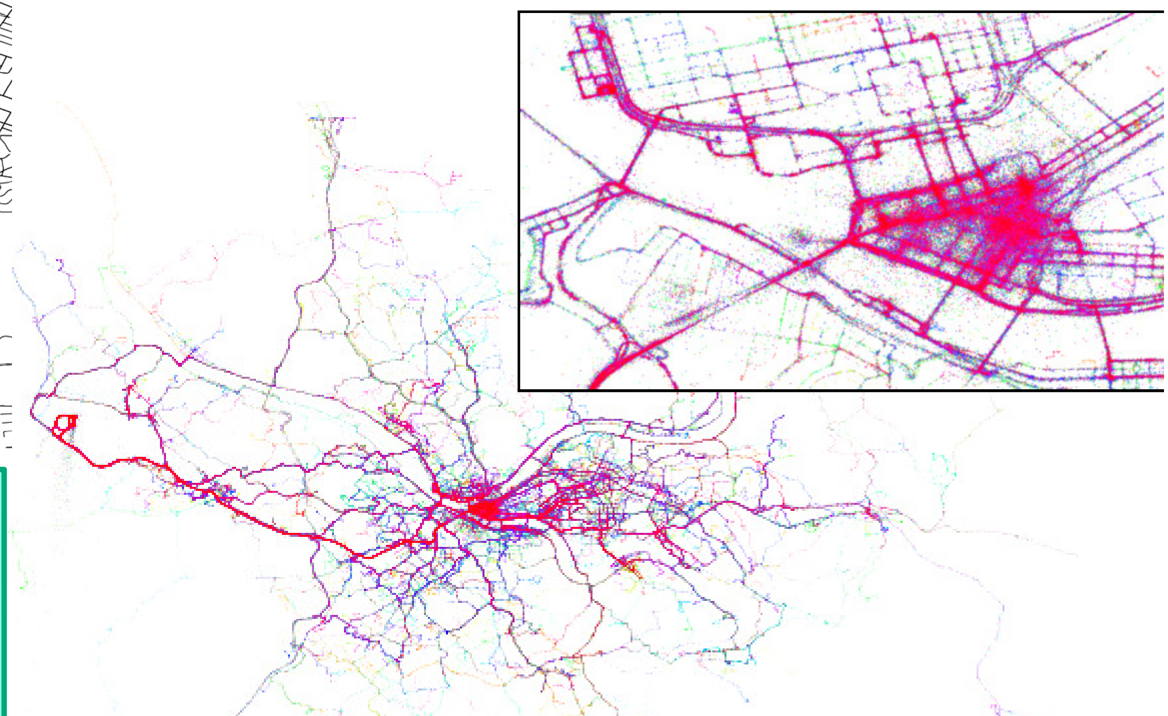
Ziebart+al, 2007/8/9

Data Collection

Length
Speed
Road
Type
Lanes

Accidents
Construction
Congestion
Time of day

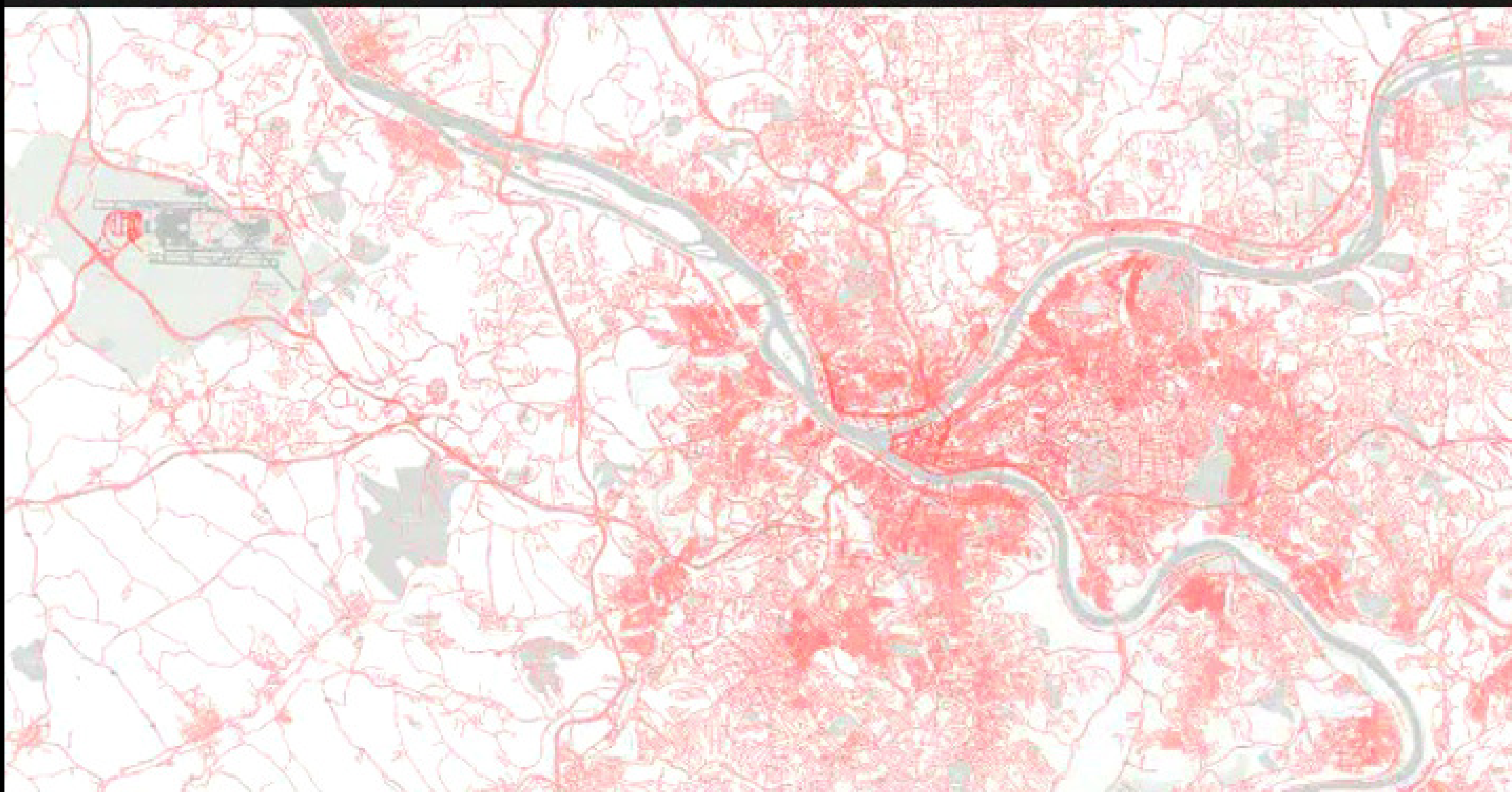
25 Taxi Drivers



Over 100,000 miles

Ziebart+al, 2007/8/9

Destination Prediction



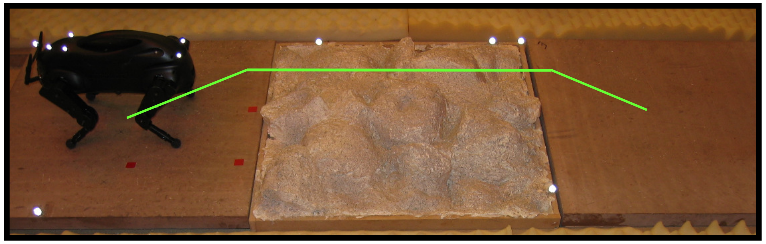
Quadruped

- Reward function trades off 25 features.

Hierarchical max margin [Kolter, Abbeel & Ng, 2008]

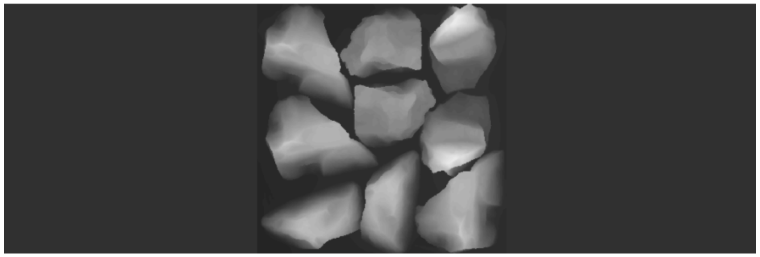
Experimental setup

- Demonstrate path across the "training terrain"



- Run our apprenticeship learning algorithm to find the reward function

- Receive "testing terrain"---height map.



- Find the optimal policy with respect to the *learned reward function* for crossing the testing terrain.

Hierarchical max margin [Kolter, Abbeel & Ng, 2008]

Without learning

With learned reward function

Quadruped: Ratliff + al, 2007

- Run footstep planner as expert (slow!)
- Run boosted max margin to find a reward function that explains the center of gravity path of the robot (smaller state space)

- At control time: use the learned reward function as a heuristic for A* search when performing footstep-level planning

Lecture outline

- Example of apprenticeship learning via inverse RL
- Inverse RL vs. behavioral cloning
- Sketch of history of inverse RL
- Mathematical formulations for inverse RL
- Case studies

- *Trajectory-based reward, with application to autonomous helicopter flight*

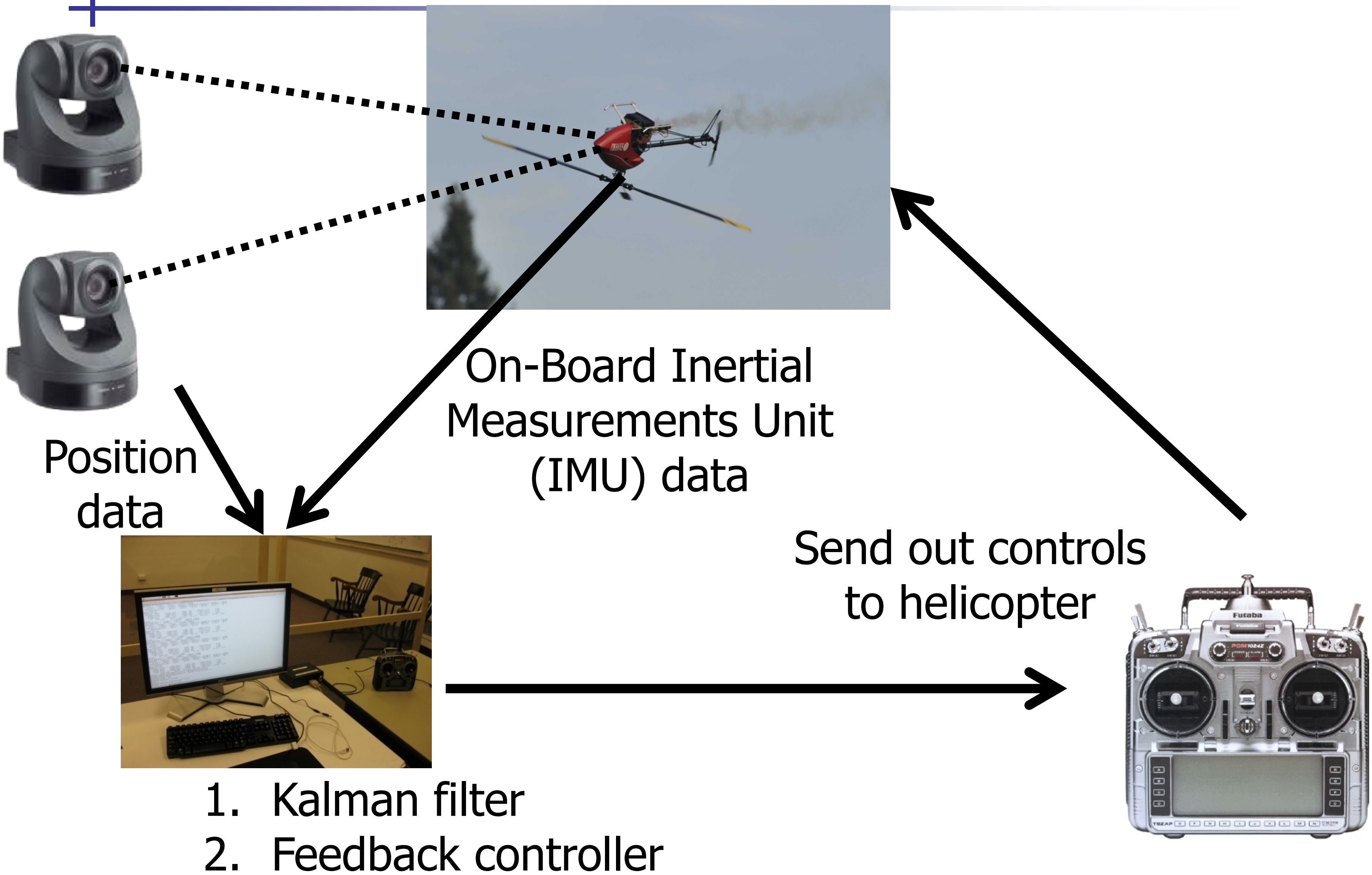
Remainder of lecture: extreme helicopter flight

- How does helicopter dynamics work
- Autonomous helicopter setup
- Application of inverse RL to autonomous helicopter flight

Helicopter dynamics

- 4 control inputs:
  - Main rotor collective pitch
  - Main rotor cyclic pitch (roll and pitch)
  - Tail rotor collective pitch

Autonomous helicopter setup



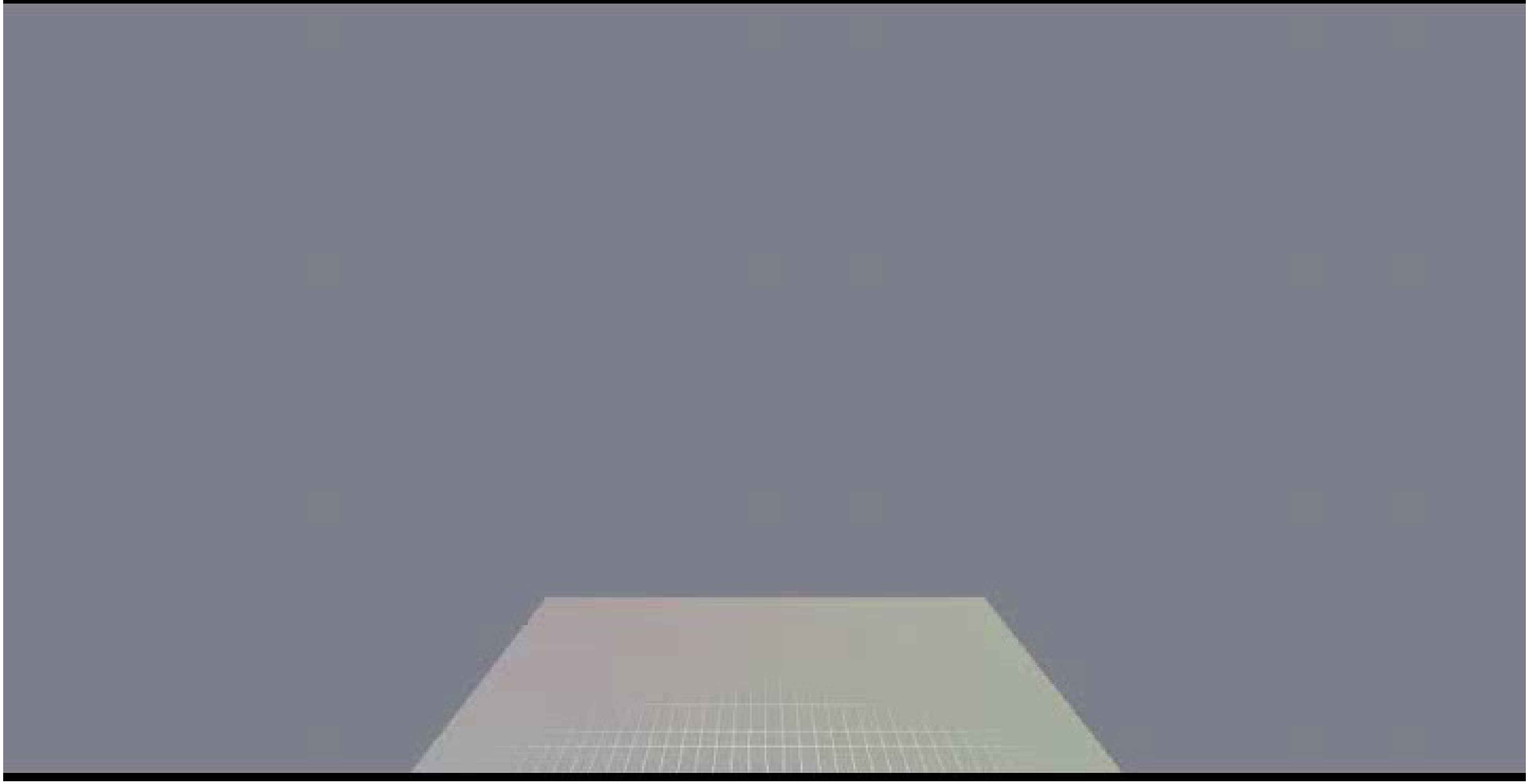
Motivating example

- How do we specify a task like this???

Key difficulties

- Often very difficult to specify trajectory by hand.
  - Difficult to articulate exactly how a task is performed.
  - The trajectory should obey the system dynamics.
- Use an expert *demonstration* as trajectory.
  - But, getting perfect demonstrations is hard.
- Use multiple suboptimal demonstrations.
  - ~~Simply average them?~~
  - **How to find the experts "intended" trajectory from suboptimal demonstrations?**

Expert demonstrations: Airshow



Graphical model

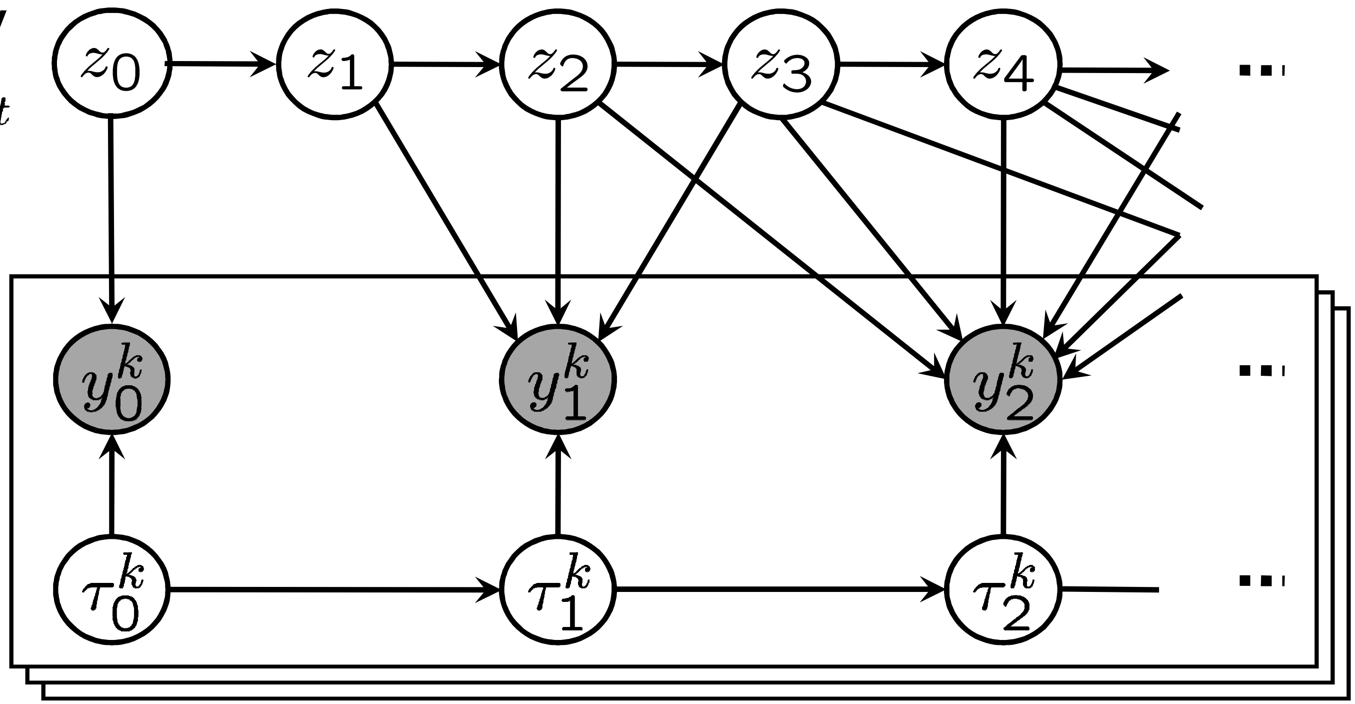
Intended trajectory

$$z_{t+1} = f(z_t) + \omega_t$$

Expert demonstrations

$$y_j = z_{\tau_j} + \nu_j$$

Time indices



- Intended trajectory satisfies dynamics.
- Expert trajectory is a noisy observation of one of the hidden states.
  - But we don't know exactly which one.

[ICML 2008]

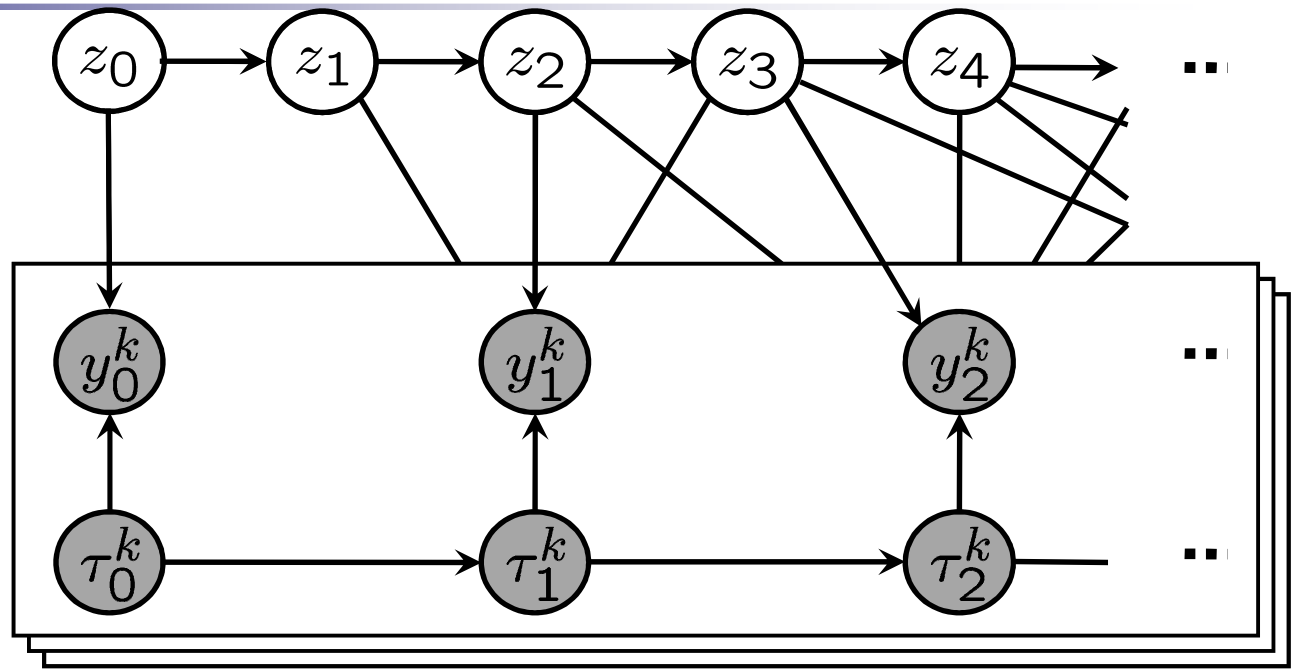
Learning algorithm

- Similar models appear in speech processing, genetic sequence alignment.
  - See, e.g., Listgarten et. al., 2005
- Maximize likelihood of the demonstration data over:
  - Intended trajectory states
  - Time index values
  - Variance parameters for noise terms
  - Time index distribution parameters

Learning algorithm

If τ is unknown, inference is hard.

If τ is known, we have a standard HMM.



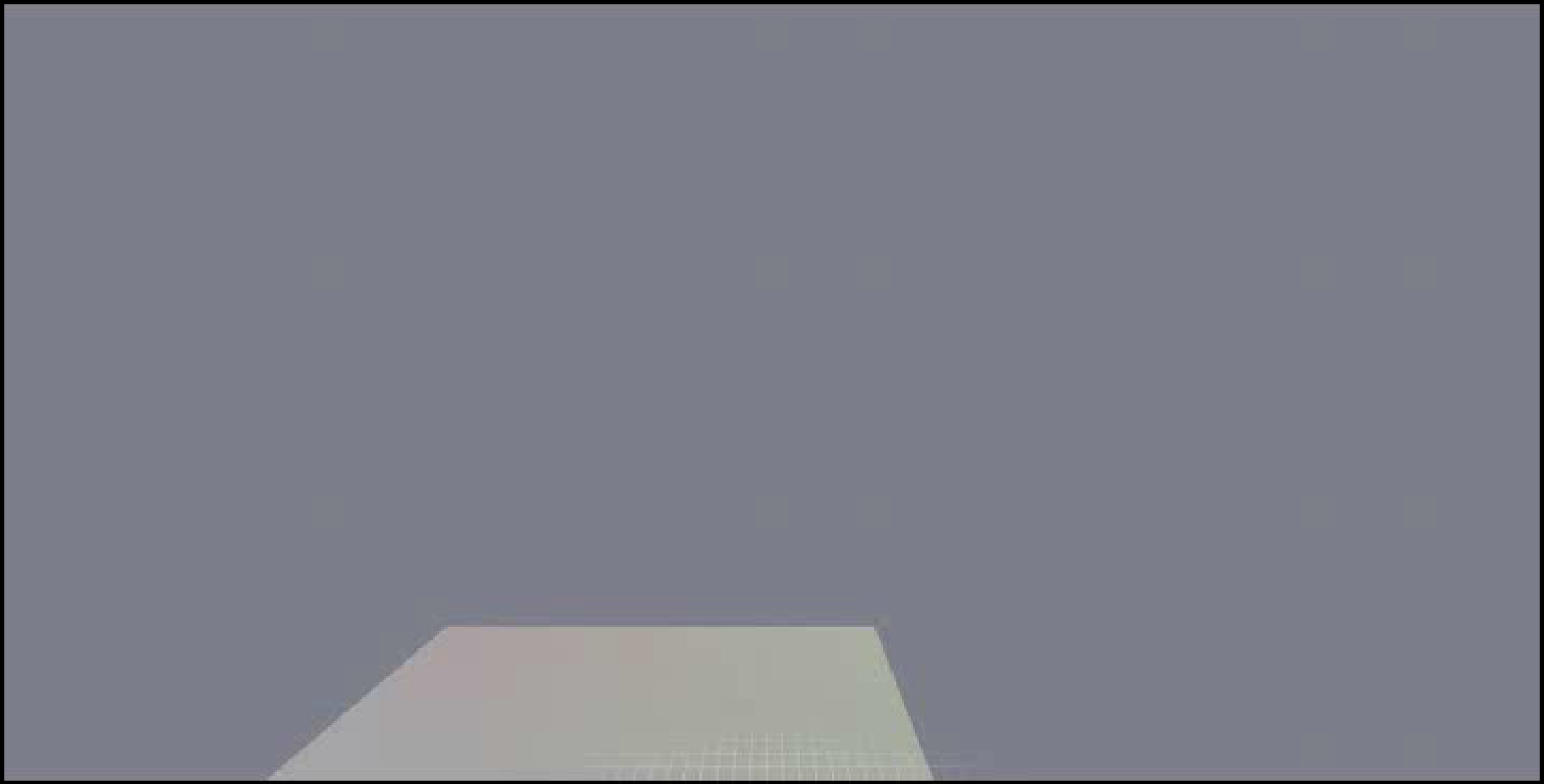
- Make an initial guess for τ.
- Alternate between:
  - Fix τ. Run EM on resulting HMM.
  - Choose new τ using dynamic programming.

Details: Incorporating prior knowledge

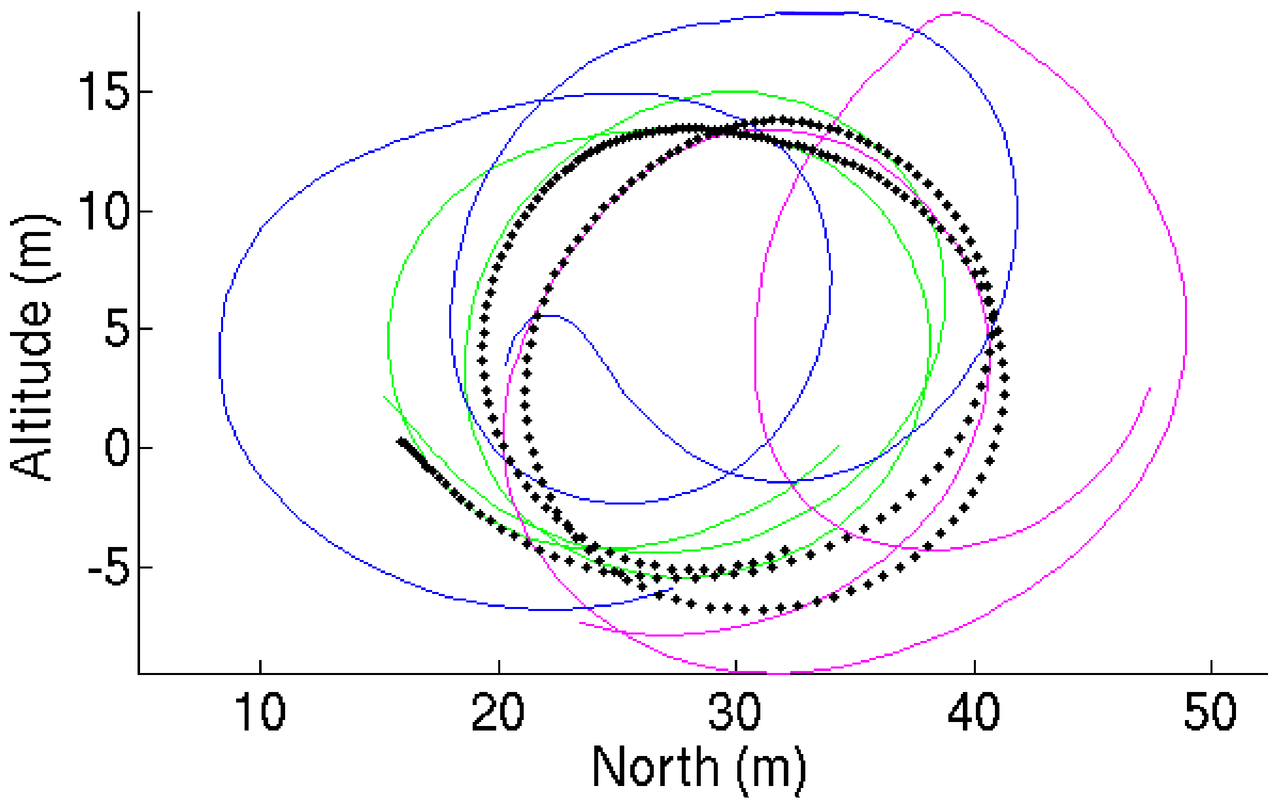
- Might have some limited knowledge about how the trajectory should look.
  - Flips and rolls should stay in place.
  - Vertical loops should lie in a vertical plane.
  - Pilot tends to “drift” away from intended trajectory.

Results: Time-aligned demonstrations

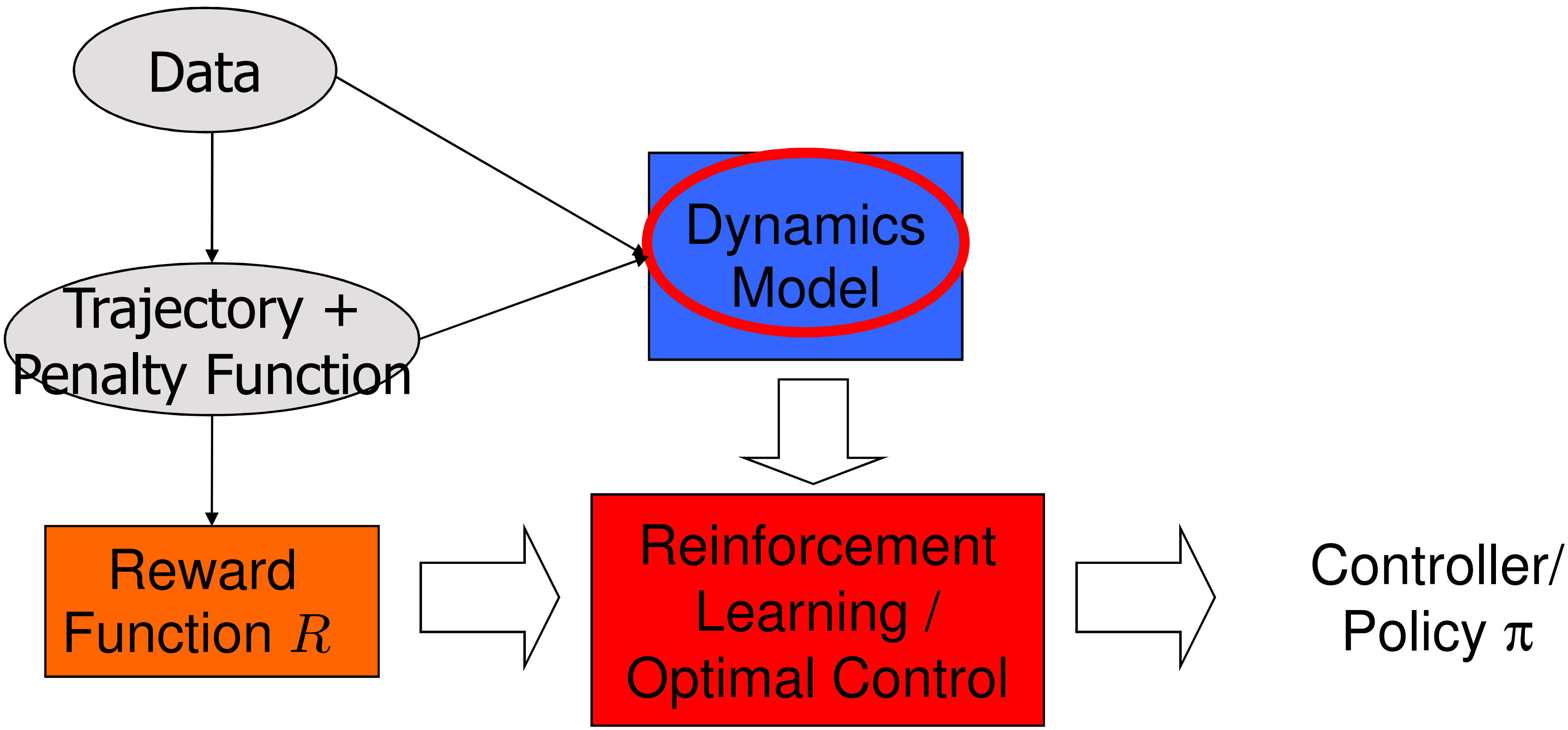
- White helicopter is inferred “intended” trajectory.



Results: Loops

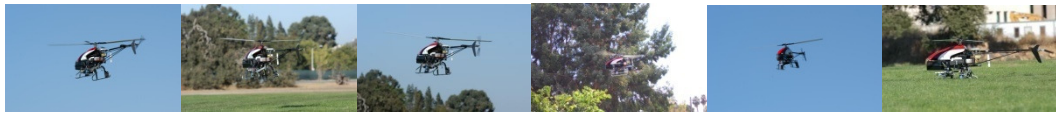


High-level picture



Experimental setup for helicopter

1. Our expert pilot demonstrates the airshow several times.



2. Learn (by solving a joint optimization problem):

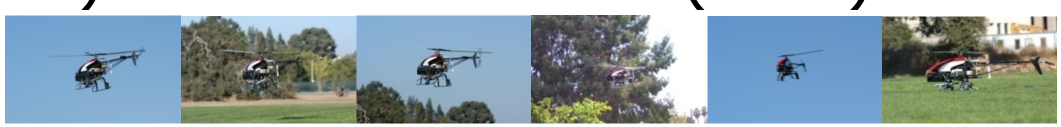
   Reward function---trajectory.

   Dynamics model---trajectory-specific local model.

3. Fly autonomously:

   Inertial sensing + vision-based position sensing → (extended) Kalman filter

   Receding horizon differential dynamic programming (DDP) feedback controller (20Hz)



Learning to fly new aerobatics takes < 1 hour

Related work

- Bagnell & Schneider, 2001; LaCivita, Papageorgiou, Messner & Kanade, 2002; Ng, Kim, Jordan & Sastry 2004a (2001);
- Roberts, Corke & Buskey, 2003; Saripalli, Montgomery & Sukhatme, 2003; Shim, Chung, Kim & Sastry, 2003; Doherty et al., 2004.
- Gavrilets, Martinos, Mettler and Feron, 2002; Ng et al., 2004b.

- *Maneuvers presented here are significantly* ***more challenging*** *and* ***more diverse*** *than those performed by any other autonomous helicopter.*

Autonomous aerobatic flips (attempt) before apprenticeship learning

Task description: meticulously hand-engineered
Model: learned from (commonly used) frequency sweeps data

Results: Autonomous airshow

Summary

- Example of apprenticeship learning via inverse RL
- Inverse RL vs. behavioral cloning
- Sketch of history of inverse RL
- Mathematical formulations for inverse RL
- Case studies
- Trajectory-based reward, with application to autonomous helicopter flight

[redacted]

Thank you.

Questions?

More materials

- Lecture based upon these slides (by Pieter Abbeel) to appear on VideoLectures.net [part of Robot Learning Summer School, 2009]

- Helicopter work:
  - http://heli.stanford.edu for helicopter videos and video lecture (ICML 2008, Coates, Abbeel and Ng) and thesis defense (Abbeel)

More materials / References

- Abbeel and Ng. Apprenticeship learning via inverse reinforcement learning. (ICML 2004)
- Abbeel, Dolgov, Ng and Thrun. Apprenticeship learning for motion planning with application to parking lot navigation.
- Boyd, Ghaoui, Feron and Balakrishnan. (1994). Linear matrix inequalities in system and control theory. SIAM.
- Coates, Abbeel and Ng. Learning for control from multiple demonstrations. (ICML 2008)
- Kolter, Abbeel and Ng. Hierarchical apprenticeship learning with application to quadruped locomotion. (NIPS 2008)
- Neu and Szepesvari. Apprenticeship learning using inverse reinforcement learning and gradient methods. (UAI 2007)
- Ng and Russell. Algorithms for inverse reinforcement learning. (ICML 2000)
- Ratliff, Bagnell and Zinkevich. Maximum margin planning. (ICML 2006)
- Ratliff, Bradley, Bagnell and Chestnutt. Boosting structured prediction for imitation learning. (NIPS 2007)
- Ratliff, Bagnell and Srinivasa. Imitation learning for locomotoin and manipulation. (I. Conf. on Humanoid Robotics, 2007)
- Ramachandran and Amir. Bayesian inverse reinforcement learning. (IJCAI 2007)
- Syed and Schapire. A game-theoretic approach to apprenticeship learning. (NIPS 2008)
- Ziebart, Maas, Bagnell and Dey. Maximum entropy inverse reinforcement learning. (AAAI 2008)