CS 343: Artificial Intelligence

Advanced Applications: Robotics

Prof. Scott Niekum — The University of Texas at Austin

[These slides based on those of Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]
Robotic Helicopters
Motivating Example

• How do we execute a task like this?
Autonomous Helicopter Flight

Key challenges:

- Track helicopter position and orientation during flight
- Decide on control inputs to send to helicopter
Autonomous Helicopter Setup

- On-board inertial measurement unit (IMU)
- Send out controls to helicopter
- Position

Diagram shows a setup for autonomous helicopter control with a control panel, computer, and helicopter with an IMU.
HMM for Tracking the Helicopter

- **State:** \( s = (x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}) \)

- **Measurements:** [observation update]
  - 3-D coordinates from vision, 3-axis magnetometer, 3-axis gyro, 3-axis accelerometer

- **Transitions (dynamics):** [time elapse update]
  - \( s_{t+1} = f(s_t, a_t) + w_t \)  
    - \( f \): encodes helicopter dynamics, \( w \): noise
Helicopter MDP

- **State:** \( s = (x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}) \)

- **Actions (control inputs):**
  - \( a_{\text{lon}} \): Main rotor longitudinal cyclic pitch control (affects pitch rate)
  - \( a_{\text{lat}} \): Main rotor latitudinal cyclic pitch control (affects roll rate)
  - \( a_{\text{coll}} \): Main rotor collective pitch (affects main rotor thrust)
  - \( a_{\text{rud}} \): Tail rotor collective pitch (affects tail rotor thrust)

- **Transitions (dynamics):**
  \[ s_{t+1} = f(s_t, a_t) + w_t \]
  
  \([f \text{ encodes helicopter dynamics}] \]
  
  \([w \text{ is a probabilistic noise model}] \]

- **Can we solve the MDP yet?**
Problem: What’s the Reward?

- **Reward for hovering:**

\[
R(s) = -\alpha_x (x - x^*)^2 \\
-\alpha_y (y - y^*)^2 \\
-\alpha_z (z - z^*)^2 \\
-\alpha_x \dot{x}^2 \\
-\alpha_y \dot{y}^2 \\
-\alpha_z \dot{z}^2
\]
Hover

[Ng et al, 2004]
Problem: What’s the Reward?

- Rewards for “Flip”?  
  - Problem: what’s the target trajectory?  
  - Just write it down by hand?  
  - Penalize for deviation from trajectory
Flips (?)
Helicopter Apprenticeship?
Unaligned Demonstrations
Probabilistic Alignment using a Bayes’ Net

Intended trajectory

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

Expert demonstrations

\[ y_{ij} = z_{\tau_j} + \nu_j \]

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.

[Coates, Abbeel & Ng, 2008]
Aligned Demonstrations
Final Behavior

[Abbeel, Coates, Quigley, Ng, 2010]
Legged Locomotion
Quadruped

- Low-level control problem: moving a foot into a new location → search with successor function ~ moving the motors
- High-level control problem: where should we place the feet?
  - Reward function \( R(x) = w \cdot f(s) \) [25 features]

[Kolter, Abbeel & Ng, 2008]
Experimental setup

- Demonstrate path across the “training terrain”

- Run apprenticeship to learn the reward function

- Receive “testing terrain”---height map.

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

[Kolter, Abbeel & Ng, 2008]
Learning task objectives: Inverse reinforcement learning

Reinforcement learning basics:

MDP: \((S, A, T, \gamma, D, R)\)

Policy: \(\pi(s, a) \rightarrow [0, 1]\)

Value function: \(V^\pi(s_0) = \sum_{t=0}^{\infty} \gamma^t R(s_t)\)

What if we have an MDP/R?
Learning task objectives: Inverse reinforcement learning

1. Collect user demonstration \((s_0, a_0), (s_1, a_1), \ldots, (s_n, a_n)\)
   and assume it is sampled from the expert’s policy, \(\pi^E\)

2. Explain expert demos by finding \(R^*\) such that:

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

\[
E_{s_0 \sim D}[V_{\pi^E}(s_0)] \geq E_{s_0 \sim D}[V_{\pi}(s_0)] \quad \forall \pi
\]

How can search be made tractable?

[Abbeel and Ng 2004]
Learning task objectives: Inverse reinforcement learning

Define $R^*$ as a linear combination of features:

$$ R^*(s) = w^T \phi(s), \text{ where } \phi: S \rightarrow \mathbb{R}^n $$

Then,

$$ E[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) | \pi] = E[\sum_{t=0}^{\infty} \gamma^t w^T \phi(s_t) | \pi] $$

$$ = w^T E[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi] $$

$$ = w^T \mu(\pi) $$

Thus, the expected value of a policy can be expressed as a weighted sum of the expected features $\mu(\pi)$

[Abbeel and Ng 2004]
Learning task objectives: Inverse reinforcement learning

Originally - Explain expert demos by finding $R^*$ such that:

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

Use expected features:

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

Restated - find $w^*$ such that:

$$w^* \mu(\pi^E) \geq w^* \mu(\pi) \quad \forall \pi$$

[Abbeel and Ng 2004]
Learning task objectives: Inverse reinforcement learning

Goal: Find $w^*$ such that:

$$w^* \mu(\pi^E) \geq w^* \mu(\pi) \quad \forall \pi$$

1. Initialize $\pi_0$ to any policy

Iterate for $i = 1, 2, \ldots$:

2. Find $w^*$ s.t. expert maximally outperforms all previously examined policies $\pi_0 \ldots i-1$:

$$\max_{\epsilon, w^*: \|w^*\|_2 \leq 1} \epsilon \quad \text{s.t.} \quad w^* \mu(\pi^E) \geq w^* \mu(\pi_j) + \epsilon$$

3. Use RL to calc. optimal policy $\pi_i$ associated with $w^*$

4. Stop if $\epsilon \leq$ threshold

[Abbeel and Ng 2004]
Learning task objectives: Inverse reinforcement learning

Goal: Find $w^*$ such that: $w^* \mu(\pi^E) \geq w^* \mu(\pi) \ \forall \pi$

1. Initialize $\pi_0$ to any policy

Iterate for $i = 1, 2, \ldots$:

2. Find $w^*$ s.t. expert maximally outperforms all previously examined policies $\pi_0 \ldots i-1$:

$$\max_{\epsilon, w^* : \|w^*\|_2 \leq 1} \epsilon \quad \text{s.t.} \quad w^* \mu(\pi^E) \geq w^* \mu(\pi_j) + \epsilon$$

3. Use RL to calc. optimal policy $\pi_i$ associated with $w^*$

4. Stop if $\epsilon \leq$ threshold

[Abbeel and Ng 2004]
Without learning
With learned reward function
Robotic manipulation
Demonstration

[RSS 2013, IJRR 2015]
High-level task modeling

Unsegmented demonstrations of multi-step tasks

Finite-state task representation

Why?
- Superior generalization of skills
- Handle contingencies
- Adaptively sequence skills

Questions
- How many skills?
- Parameters of skills / controllers?
- How to sequence intelligently?
System overview
System overview
System overview

Joint angles → Forward kinematics → Task demos
Gripper pose → Object recognition → Stereo data
System overview
Segmenting demonstrations

Motion categories

Standard Hidden Markov Model

Observations

[IROS 2012]
Segmenting demonstrations

\[ y_{t}^{(i)} = \sum_{j=1}^{r} A_{j,z_{t}^{(i)}} y_{t-j}^{(i)} + e_{t}^{(i)} (z_{t}^{(i)}) \]
Segmenting demonstrations

\[ y_t^{(i)} = \sum_{j=1}^{r} A_{j,z_t^{(i)}} y_{t-j}^{(i)} + e_t^{(i)} (z_t^{(i)}) \]
Segmenting demonstrations

\[ y_t^{(i)} = \sum_{j=1}^{r} A_{j, z_t^{(i)}} y_{t-j}^{(i)} + e_t^{(i)} (z_t^{(i)}) \]

Motion categories

Observations

Autoregressive Hidden Markov Model

[IROS 2012]
Autoregressive Hidden Markov Model

\[ y_t^{(i)} = \sum_{j=1}^{r} A_{j, z_t^{(i)}} y_{t-j}^{(i)} + e_t^{(i)} (z_t^{(i)}) \]

Segmenting demonstrations

Motion categories

Observations

Beta Process  Autoregressive Hidden Markov Model
(Fox et al. 2011)

[IROS 2012]
System overview
System overview

Task demos

Forward kinematics

Joint angles

Gripper pose

Object recognition

Preprocessing / BP-AR-HMM segmentation

Stereo data

Segmented motions

Coordinate frame detection

System overview

Learning from demonstration

DMPs with frame-relative goals

Segmented motions

Gripper pose

Stereo data

[IROS 2012]
System overview

Joint angles
Gripper pose
Stereo data

Task demos
Forward kinematics
Object recognition

Preprocessing / BP-AR-HMM segmentation
Segmented motions
Coordinate frame detection

Frame-labeled segments
Learning from demonstration

DMPs with frame-relative goals

Realtime data from a novel task
Joint angles
Gripper pose
Object recognition
Stereo data

[IROS 2012]
System overview
Learning a task plan: Finite state automata

[RSS 2013, IJRR 2015]
Learning a task plan: Finite state automata

Controller built from motion category examples

Classifier built from robot percepts

[RSS 2013, IJRR 2015]
Interactive corrections

[RSS 2013, IJRR 2015]
Replay with corrections: missed grasp

[RSS 2013, IJRR 2015]
Replay with corrections: too far away
Replay with corrections: full run

[RSS 2013, IJRR 2015]