CSE 571
Inverse Optimal Control
(Inverse Reinforcement Learning)

Many slides by Drew Bagnell
Carnegie Mellon University

Optimal Control Solution

Mode 1: Training example

Mode 1: Learned behavior
Mode 2: Learned cost map

\[
\text{Cost} = w^T F
\]

Ratliff, Bagnell, Zinkevich 2005
Ratliff, Bradley, Bagnell, Chestnutt, NIPS 2006
Silver, Bagnell, Stentz, RSS 2008

\[ (\text{High Cost}, \text{Low Cost}) \rightarrow \text{Learn F}_1 \]

Ratliff, Bagnell, Zinkevich, ICML 2006
Ratliff, Bradley, Bagnell, Chestnutt, NIPS 2006
Silver, Bagnell, Stentz, RSS 2008

\[ (\text{High Cost}, \text{Low Cost}) \rightarrow \text{Learn F}_2 \]

Ratliff, Bagnell, Zinkevich, ICML 2006
Ratliff, Bradley, Bagnell, Chestnutt, NIPS 2006
Silver, Bagnell, Stentz, RSS 2008
Learning Manipulation Preferences

- **Input**: Human demonstrations of preferred behavior (e.g., moving a cup of water upright without spilling)
- **Output**: Learned cost function that results in trajectories satisfying user preferences
Demonstration(s) → Graph → Projection

Demonstration(s) → Graph → Projection

Demonstration(s) → Graph → Projection

Demonstration(s) → Graph → Projection

Demonstration(s) → Graph

Demonstration(s) → Graph

Output trajectories → Discrete sampled paths → Learned cost

Discrete sampled paths → Learned cost
**2D obstacle avoidance task**

- **Graph generation**
  - **Goal:** Construct a graph in the robot’s configuration space providing good coverage

- **Projection**
  - **Goal:** Project the continuous demonstration onto the graph, resulting in a discrete graph path
  - Use a modified Dijkstra’s algorithm minimizing sum of:
    - Length of discrete path (Euclidean)
    - Distance to continuous demonstration

- **Learning the cost function**
  - **Goal:** Given projected demonstrations, learn the cost function
  - Learn feature weights \( \theta \) using softened value iteration on the discrete graph (MaxEnt IOC - Ziebart et al., 2008)
    - State dependent features (e.g., Distance to obstacles)
Experimental Results

Setup

- **Binary** state-dependent features (~95)
  - Histograms of distances to objects
  - Histograms of end-effector orientation
  - Object specific features (electronic vs non-electronic)
  - Approach direction w.r.t goal

- **Comparison**:
  - Human demonstrations
  - Obstacle avoidance planner (CHOMP)
  - Locally optimal IOC approach (similar to Max-Margin planning, Ratliff et. al., 2007)

Laptop task: Demonstration
(Not part of training set)

Laptop task: LTO + Discrete graph path

Laptop task: LTO + Smooth random path

Statistics for Laptop task

<table>
<thead>
<tr>
<th>Method</th>
<th>% Points in collision</th>
<th>End-Effector deviation (deg)</th>
<th>% Points above laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Demonstration</td>
<td>2.7</td>
<td>7.4</td>
<td>2.1</td>
</tr>
<tr>
<td>Obstacle avoidance planner</td>
<td>12.9</td>
<td>18.2</td>
<td>17.3</td>
</tr>
<tr>
<td>Coarse, discrete graph sample</td>
<td>12.8</td>
<td>9.9</td>
<td>11.1</td>
</tr>
<tr>
<td>Local Trajectory Optimizer + Graph samples</td>
<td>4.5</td>
<td>3.3</td>
<td>1.2</td>
</tr>
<tr>
<td>Local Trajectory Optimizer + Random path</td>
<td>4.5</td>
<td>5.5</td>
<td>3.1</td>
</tr>
</tbody>
</table>