Distance Minimization for Reward Learning from Scored Trajectories

Ben Burchfiel, Carlo Tomasi, and Ronald Parr

Learning from Demonstration (LfD)

- Learn by observing another agent
  - Natural way to guide learning

- Can mean multiple things:
  - Learn to mimic
  - Learn skills
  - Learn motivation

- May be interactive
Learning from Demonstration (LfD)

- Learn by observing another agent
  - Natural way to guide learning
- Can mean multiple things:
  - Learn to mimic
  - Learn skills
  - Learn motivation
- May be interactive

Inverse Reinforcement Learning

- Given how the world works, what should we do?
  - Infer motivation behind demonstration
- Markov Decision Problem (MDP) with no reward function
IRL as an MDP/R

- **Known:**
  - State space (S)
  - Action set (A)
  - Discount factor (\(\gamma\))
  - Transition function (\(T(s, a, s')\))
  - Initial state distribution (D)

- **Unknown:**
  - Reward function (R(s))

* Not required for DM-IRL

Learning an Expert’s Reward

- **Traditional IRL**
  - Given near-optimal expert demonstrations
  - Find R(s) inducing “close” behavior

  - Solution is ill-posed
  - Solution should generalize to novel settings
Reward Features

- Needs to generalize
  - Rewards are a function of state features
  - Rewards are linear in these features:
    \[ R(s) = w^T \Phi(s) \]

DM-IRL: Learn from Scoring

- What if you don’t have an expert?
  - High degree of skill may be required?

- Learn from an educated novice?
  - High quality feedback, not high quality demonstration
Expert Judge

- Judge provides a single score to each demonstration (trajectory)
- Learner’s performance not dependent on demonstration quality

Contributions

- Evaluate the use of DM-IRL as a general purpose IRL method
- Provably bound DM-IRL’s performance with noisy input
- Extend a hardness result on generating specific trajectories
- Empirically evaluate DM-IRL on a challenging terrain navigation problem
### Comparison of Approaches

<table>
<thead>
<tr>
<th></th>
<th>Traditional IRL</th>
<th>DM-IRL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expert Requirements</strong></td>
<td>Performs trajectories</td>
<td>Assigns trajectories a single (scalar) score</td>
</tr>
<tr>
<td><strong>Optimality Assumption</strong></td>
<td>Near-optimal demonstrator</td>
<td>Near-optimal scorer</td>
</tr>
<tr>
<td><strong>Computational Requirements</strong></td>
<td>Solve to MDP(s) to find $R(s)$, then solve to find $\pi^*$</td>
<td>Regression to find $R(s)$, then solve MDP to find to $\pi^*$</td>
</tr>
<tr>
<td><strong>Transition Functions Needed</strong></td>
<td>The Learner’s and the Expert’s</td>
<td>The Learners</td>
</tr>
<tr>
<td><strong>Learned Reward Function</strong></td>
<td>Induces Expert’s Behavior</td>
<td>The Expert’s</td>
</tr>
</tbody>
</table>

### (Linear) IRL from Scored Trajectories

Key Insight:
Rewards linear in features $\rightarrow$ learn state rewards by regressing over discounted sum of accrued features

$$M$$
Discounted sum of feature values

$M$ discounted sum of feature values

$r$ scores

$V$
Vector of Trajectory Scores

$r$ trajectories

$c$ features

$w$
Vector of Feature Weights

Unknown

Pose this as a linear regression...
Scored Trajectory IRL Pipeline

- **Trajectories**: M
- **Scores**: v
- **Feature Weights**: w
- **State Features**: Φ
- **Reward Function**: R
- **Policy / Value of States**: Planning

### Rank of M

- **Number of trajectories needed?**
- **Ideally**: \( \text{rank}(\Phi) = \text{rank}(M) \)
- **Use regularization to handle** \( \text{rank}(\Phi) > \text{rank}(M) \)

\[
\hat{w}_L = \arg\min_w \| Mw - v \|_2 + \| w \|_1
\]
Expert Noise

- Expert may not label trajectories perfectly

- Noise in trajectory labels
  - Expert isn’t consistent
  - Expert is biased
  - Expert quantizes scores into categories
    - e.g. scoring on a fixed scale

Quality of Learned Policy

Theorem:

Quality of the learned policy degrades linearly with increasing noise.

\[ V_E - V_L \leq 2(\delta + \varepsilon) \]

Expected value of optimal policy

Expected value of Learned policy

Scoring error

L₁ regression fit error

Maximum magnitude of expert’s scoring error

Under mild assumptions:
Hardness of Trajectory Targets

- Active setting
  - Generate trajectories for the expert to score

- Idea: Find a policy with specific target feature expectation

- Unfortunately...
  - Known that this is weakly NP-Complete (reduction from subset sum) for multi-criteria MDPs
  - We show that for a specific feature target, this is strongly NP-Complete via reduction from 3-SAT

Satellite Terrain Domain

1,350 discrete states

Actions: Deterministic up, down, left, right

Terrain Image

Expert Reward Function
Satellite Terrain Domain, Small Feature Pool

5 features, each one a probability of semantic terrain type: Grass, Tree, Road, Dirt, Shrub

Features trained as 2 component Gaussian mixtures from hand-segmented and labeled terrain

Terrain Example

Learned Reward (Lighter is Better)
Satellite Terrain Domain, Small Feature Pool

- 1350 discrete states
- 5 features:
  - Probability of semantic terrain type: Grass, Tree, Road, Dirt, Shrub
- Features trained as 2 component Gaussian mixtures from hand segmented and labeled terrain
- Max score perturbation ($\delta$) is $0.5 \times \eta$

Satellite Terrain Domain, Large Feature Pool

- 1350 discrete states
- ~60,000 features, Gabor filters for texture, color histograms, and 512 GIST features (holistic spatial features)
- Case where good features are not known ahead of time
- Max score perturbation ($\delta$) is $0.5 \times \eta$
Satellite Terrain Domain, Large Feature Pool, Transfer

- Learn on Demonstration Terrain
- Evaluate on Evaluation Terrain
- Large pool of ~60,000 visual features

- 3 Demonstrators, each:
  - is suboptimal in different ways
  - has different (unknown) $T$

- Involves:
  - Suboptimal demos
  - Unknown $T$
  - Noisy scores
  - Reward transfer

Investigate DM-IRL’s use as a general purpose IRL method

- Provably bound the quality of learned rewards w.r.t. scoring error

- Prove a hardness result for generating trajectories with arbitrary feature targets

- Empirically confirm our theoretical findings

- Experimentally demonstrate DM-IRL on settings not suitable for traditional IRL approaches

Questions?
Hardness of Trajectory Targets

MDP for the 3-SAT instance:

\((v_1 \lor \neg v_2 \lor \neg v_4) \land (\neg v_2 \lor \neg v_3 \lor v_4)\)

- White arrows: uniform transition probability
- Red arrows: deterministic choice