On the Feasibility of Learning Human Biases for Reward Inference

Rohin Shah, Noah Gundotra, Pieter Abbeel, Anca Dragan
A conversation amongst IRL researchers
A conversation amongst IRL researchers

[Ziebart et al, 2008]

To deal with suboptimal demos, let’s model the human as *noisily rational*
A conversation amongst IRL researchers

To deal with suboptimal demos, let’s model the human as *noisily rational* [Ziebart et al, 2008]

Then you are limited to human performance, since you don’t know *how* the human made a mistake [Christiano, 2015]
A conversation amongst IRL researchers

To deal with suboptimal demos, let’s model the human as *noisily* rational

Then you are limited to human performance, since you don’t know *how* the human made a mistake

We can model human biases:
- Myopia
- Hyperbolic time discounting
- Sparse noise
- Risk sensitivity

\[
\pi(a|s) \propto e^{\beta Q(s,a;r)}
\]
A conversation amongst IRL researchers

[Christiano, 2015]

Then you are limited to human performance, since you don’t know how the human made a mistake


We can model human biases:
- Myopia
- Hyperbolic time discounting
- Sparse noise
- Risk sensitivity

[Steinhardt and Evans, 2017]

Your human model will inevitably be misspecified
A conversation amongst IRL researchers

We can model human biases:
- Myopia
- Hyperbolic time discounting
- Sparse noise
- Risk sensitivity

Your human model will inevitably be misspecified

Hmm, maybe we can learn the systematic biases from data? Then we could correct for these biases during IRL
A conversation amongst IRL researchers

[Steinhardt and Evans, 2017]
Your human model will inevitably be misspecified

 Hmm, maybe we can learn the systematic biases from data? Then we could correct for these biases during IRL

[Armstrong and Mindermann, 2017]
That’s impossible without additional assumptions
Learning a policy isn’t sufficient

Biases are a part of cognition, and are not in the policy $\pi$

We consider a **multi-task setting** so that we can learn $D$ from examples

They are in the *planning algorithm* $D$ that created the policy $\pi$
Architecture

To learn the biased planner, minimize over $\theta$

To perform IRL, minimize over $R$

World Model – $w$

Planner – $D_\theta$

Predicted Policy – $\pi'$

Reward – $r$

Gradient-based update $\min \mathcal{L}(D_\theta(W, R), \Pi)$

True Policy – $\pi$
Algorithms

Algorithm 1: Some **known rewards**
1. On tasks with known rewards, learn the planner
2. Freeze the planner and learn the reward on remaining tasks

Algorithm 2: **“Near” optimal**
1. Use Algorithm 1 to mimic a simulated optimal agent
2. Finetune planner and reward jointly on human demonstrations
Experiments

We developed five simulated human biases to test our algorithms.
(Some) Results

Our algorithms perform better on average, compared to a learned Optimal or Boltzmann model.

... But an exact model of the demonstrator does much better, hitting 98%.
Conclusion

Learning systematic biases has the potential to improve reward inference, but differentiable planners need to become significantly better before this will be feasible.