A Series of Unlikely Events

Learning from Sequential Behavior for Activity-Based Intelligence and Modeling Human Expertise

Introduction

Modeling patterns of sequential behavior is a task that underlies numerous difficult artificial intelligence tasks:

- How do I detect when adversaries are deviating from normal routines?
- How can I predict where a ship is going to dock?
- How can I automate the teaching of novice analysts to perform complex tasks as if they were experts?

In this work, we use a class of techniques called **Imitation Learning (IL)** to model sequential behavior to answer questions like these and others.

Methodology

Given observations of behavior:

$$\mathcal{B} = \{((s_1,a_1),(s_2,a_2),\ldots)1,\ldots,((s_1,a_1)\ldots)_n\}$$

Learn a policy $\pi: \mathcal{S} \mapsto \mathcal{A}$ that best explains the behaviors.

Two Kinds of Imitation Learning Algorithms

1. Inverse Reinforcement Learning: Learn a reward function $R: \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ that models the preference exhibited in behavior. Then, learn policy π that maximizes expected reward.
2. Behavioral Cloning: Learn π directly to mimic the actions exhibited in the behaviors.

How to use Imitation Learning for…

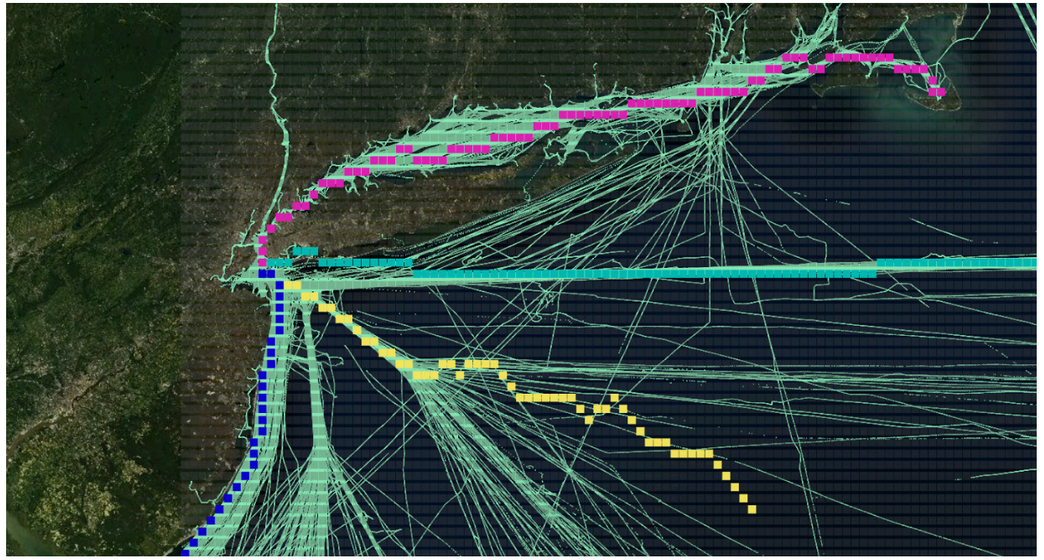
Activity-based Intelligence:

- Learn π from observed behaviors of entities of interest.
- Project future behavior by successively applying π to state.
- Detect anomalous behavior when π deems an action to be of low probability (assumes probabilistic policy).

Teaching expert behavior:

1. Learn π from expert behavior.
2. When a novice is in a state for which she doesn't know the proper action, suggest the one produced by π.

Imitation Learning techniques are an efficient and effective means to perform **activity-based intelligence** or to help automate the education of novices on how to **perform tasks like experts.**



Goals of this work

1. Investigate the practicality (assumptions made, efficiency, scalability, expressiveness) of applying IL to behavioral modeling problems.
2. Apply IL Techniques to DoD/IC relevant problems:
   - Perform efficient implementations that scale to a large number of observations.
   - Build demonstration from data ingestion to visualization tools.
3. Develop techniques that are able to explain, simulate, and demonstrate expert behavior.

Accomplishments

1. Performed technical evaluation of Generative Adversarial Imitation Learning (GAIL) (Ho and Ermon, 2016) and Disturbances for Augmenting Robot Trajectories (DART) (Lee et al., 2017) when applied to modeling nautical vessel behavior.
   - With careful engineering and domain-specific modeling assumptions, we were able to achieve a policy that was able to predict a ship's end destination state within 0.001% of their actual state (technical report forthcoming).
2. Created implementation of Maximum Causal Entropy IRL (MCEIRL) (Ziebart et al., 2010) that is 500x+ faster than academic implementation (to be publicly released).
3. Created demonstration of MCEIRL model applied to U.S. Coast Guard Nautical Vessel Data. (**https://resources.sei.cmu.edu/downloads/IRL-demo**)
4. Developed model with Stephanie Rosenthal (CMU/CSD) and Reid Simmons (CMU/RI) of expert data scientist behavior for the purpose of guiding novice data scientists through challenging tasks (technical report forthcoming).

Ho, Jonathan and Ermon, Stefano. Generative Adversarial Imitation Learning. *Advances in Neural Information Processing Systems (NIPS), 29. D. D. Lee et al. (eds]. NIPS Foundation.* 2016.
Lee, Jonathan et al. DART: Disturbances for Augmenting Robot Trajectories. *1st Conf. on Robot Learning (CORL) Project*. Nov. 2017.
Ziebart, Brian D, et al. *Maximum Causal Entropy IRL (MCEIRL)*. School of Computer Science, Carnegie Mellon University. 2010.

Eric Heim,
Jonathan Hoyle, Jay Palat, Dan DeCapria, Jake Oaks

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-0908