Inverse Reinforcement Learning in robotics

Author: Lev Kozlov

Robotics Track

l.kozlov@innopolis.university

INNOPOLIS UNIVERSITY

Table of Contents

INNOPOLIS UNIVERSITY

Why worry about learning rewards in RL?

What could be an optimal control?

$$\pi = \arg\max_{\pi} E_{\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t), \mathbf{u}_{t+1} \sim \pi(\mathbf{u}_t|\mathbf{x}_t)} [r(\mathbf{x}_t, \mathbf{u}_t)] \tag{1}$$

Better to optimize $r(\mathbf{x}_t, \mathbf{u}_t)$ to explain the data better.

Why worry about learning rewards in RL?

Imitation learning perspective:

- Simply copying actions of expert has no reasoning
- We want to infer the **intent**

RL perspective:

- Inferring reward is **underspecified** problem
- Many rewards can explain the same behaviour equally-well

IRL vs RL formally

"Forward" reinforcement learning:

- states $\mathbf{x} \in \mathbf{X}$
- controls $\mathbf{u} \in \mathbf{U}$
- (sometimes) dynamics $f(\mathbf{x}^+|\mathbf{x}, \mathbf{u})$
- reward $r(\mathbf{x}, \mathbf{u})$

Learn policy $\pi^*(\mathbf{u}|\mathbf{x})$

Inverse reinforcement learning:

- states $\mathbf{x} \in \mathbf{X}$
- controls $\mathbf{u} \in \mathbf{U}$
- (sometimes) dynamics $f(\mathbf{x}^+|\mathbf{x}, \mathbf{u})$
- samples τ_i from $\pi^*(\tau)$

Learn $r_\psi(\mathbf{x}, \mathbf{u})$ to later learn policy $\pi^*(\mathbf{u}|\mathbf{x})$

Reward function parameterization

Linear reward function:

$$r_\psi(\mathbf{x}, \mathbf{u}) = \sum_i \psi_i f_i(\mathbf{x}, \mathbf{u}) = \psi^T \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{2}$$

In more complex case the reward could be a separate neural net mapping from $\mathbf{x}$ and $\mathbf{u}$ to $r_\psi(\mathbf{x}, \mathbf{u})$

Learning the reward

Same as learning the optimality variable.

$$p(\mathcal{O}|\mathbf{x}_t, \mathbf{u}_t, \psi) = exp(r_\psi(\mathbf{x}_t, \mathbf{u}_t)) \tag{3}$$

$$p(\tau|\mathcal{O}, \psi) \propto p(\tau)\ exp(\sum_t r_\psi(\mathbf{x}_t, \mathbf{u}_t)) \tag{4}$$

Note that $p(\tau)$ is not dependent on ψ.
The whole thing becomes maximum likelihood learning:

$$\max_\psi \frac{1}{N}\sum_{i=1}^{N} log\ p(\tau_i|\mathcal{O}_{1:T}, \psi) = \max_\psi \frac{1}{N}\sum_{i=1}^{N} r_\psi(\tau_i) - log\ Z \tag{5}$$

INNOPOLIS UNIVERSITY

Partition function

Normalizer (partition) function could be defined as:

$$Z = \int p(\tau)\ exp(r_\psi(\tau))\ d\tau \tag{6}$$

Just compute gradient and optimize:

$$\nabla_\psi \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\psi\ r_\psi(\tau_i) - \frac{1}{Z}\int p(\tau)\ exp(r_\psi(\tau))\ \nabla_\psi\ r_\psi(\tau)\ d\tau \tag{7}$$

Partition function

But second term can be considered as expected value and equation becomes:

$$\nabla_\psi \mathcal{L} = E_{\tau \sim \pi^\star(\tau)} \left[\nabla_\psi r_\psi \left(\tau_i\right)\right] - E_{\tau \sim p(\tau | \mathcal{O}_{1:T}, \psi)} \left[\nabla_\psi r_\psi(\tau)\right] \tag{8}$$

- First item is estimation over expert samples
- Second item is soft optimal policy under current reward

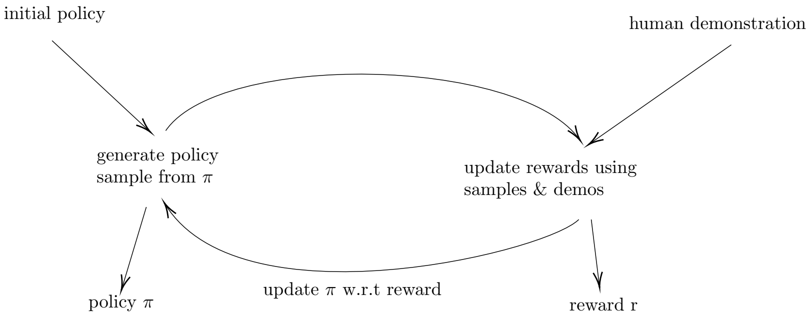
MaxEnt IRL algorithm [1]

- compute probability of control given state being optimal for reward (backward message)
- compute probability of state begin optimal for reward (forward message)
- compute state-action visitation probability for pairs $(\mathbf{x}_t, \mathbf{u}_t)$
- evaluate gradient
- update

Guided cost learning algorithm [2]

As summation over policy samples is quite costly, we can use weights:

$$w_j = \frac{p(\tau)\ exp(r_\psi(\tau_j))}{\pi(\tau_j)} \tag{9}$$



INNOPOLIS UNIVERSITY

IRL and GANs

- Policy tries to fool the reward that it is a human demo
- Reward tries to distinguish between human demo and artificial one

Correspondence:

- trajectory τ
- policy $\pi \sim q(\tau)$
- reward **r**

- sample **x**
- generator **G**
- discriminator **D**

OPIRL: Sample Efficient Off-Policy Inverse Reinforcement Learning via Distribution Matching [3]

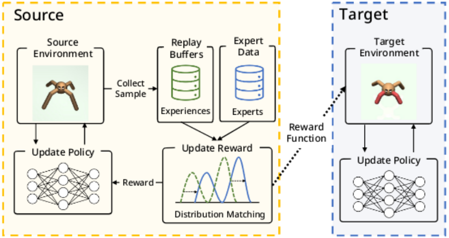


Fig. 1: **Overview of OPIRL framework:** OPIRL (1) achieves high sample efficiency and (2) recovers a robust reward function that can be transferred and generalize across different environments by applying off-policy training in distribution matching.

OPIRL: Sample Efficient Off-Policy Inverse Reinforcement Learning via Distribution Matching [3]

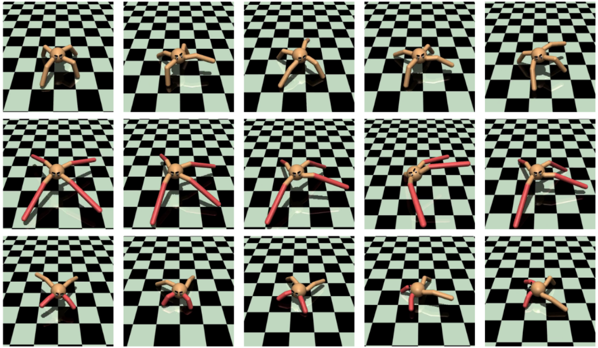


Fig. 9: **Top row:** The original quadrupedal ant moving from left to right. **Middle row:** BigAnt with successfully learning the policy with transferred reward function from the original quadrupedal ant. **Bottom row:** AmputatedAnt moving from left to right. Due to the amputated legs, it cannot move like the other environments, instead it turns backward to move in the right direction.

INNOPOLIS UNIVERSITY

DriveIRL: Drive in Real Life with Inverse Reinforcement Learning [4]

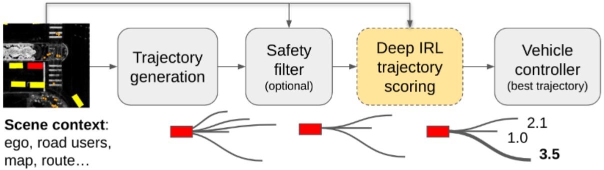
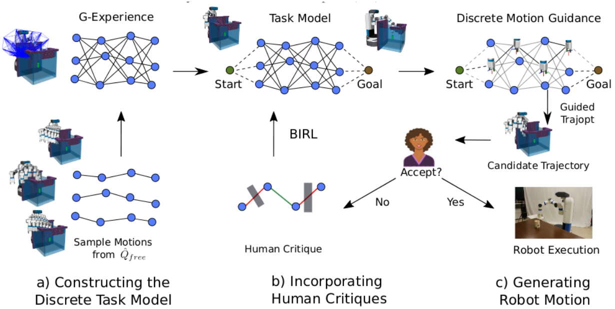
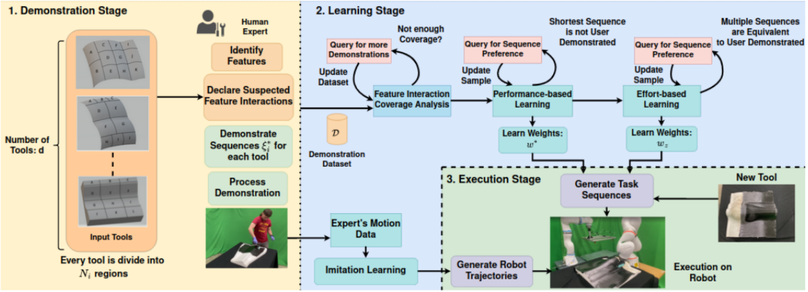


Fig. 1. DriveIRL architecture. The learned scoring component is indicated with a dotted boundary.

Human-Guided Motion Planning in Partially Observable Environments [5]



Inverse Reinforcement Learning Framework for Transferring Task Sequencing Policies from Humans to Robots in Manufacturing Applications [6]



Online Prediction of Lane Change with a Hierarchical Learning-Based Approach [7]

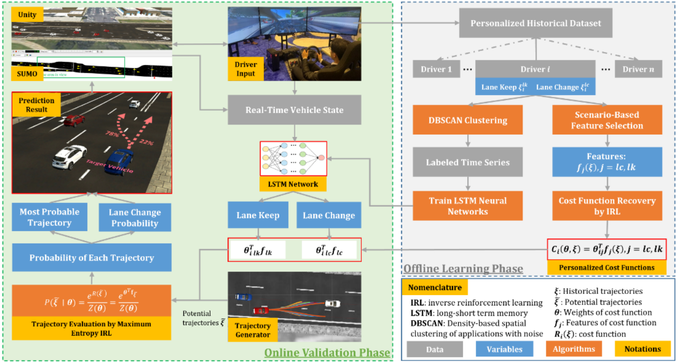


Fig. 1. System workflow of the hierarchical learning-based approach for lane-change prediction with an offline learning phase (in grey) and an online validation phase (in green).

Personalized Car Following for Autonomous Driving with Inverse Reinforcement Learning [8]

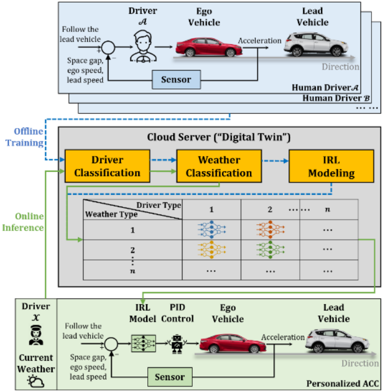


Fig. 1. System architecture of the proposed personalized adaptive cruise control (P-ACC) system.

References I

[1] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum Entropy Inverse Reinforcement Learning," en,

[2] C. Finn, S. Levine, and P. Abbeel, *Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization*, arXiv:1603.00448 [cs], May 2016. DOI: 10.48550/arXiv.1603.00448.

[3] H. Hoshino, K. Ota, A. Kanezaki, and R. Yokota, "OPIRL: Sample Efficient Off-Policy Inverse Reinforcement Learning via Distribution Matching," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 448–454. DOI: 10.1109/ICRA46639.2022.9811660.

[4] T. Phan-Minh, F. Howington, T.-S. Chu, *et al.*, "DriveIRL: Drive in Real Life with Inverse Reinforcement Learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 1544–1550. DOI: 10.1109/ICRA48891.2023.10160449.

References II

[5] C. Quintero-Peña, C. Chamzas, Z. Sun, V. Unhelkar, and L. E. Kavraki, "Human-Guided Motion Planning in Partially Observable Environments," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 7226–7232. DOI: 10.1109/ICRA46639.2022.9811893.

[6] O. M. Manyar, Z. McNulty, S. Nikolaidis, and S. K. Gupta, "Inverse Reinforcement Learning Framework for Transferring Task Sequencing Policies from Humans to Robots in Manufacturing Applications," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 849–856. DOI: 10.1109/ICRA48891.2023.10160687.

[7] X. Liao, Z. Wang, X. Zhao, *et al.*, "Online Prediction of Lane Change with a Hierarchical Learning-Based Approach," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 948–954. DOI: 10.1109/ICRA46639.2022.9812269.

INNOPOLIS UNIVERSITY

References III

[8] Z. Zhao, Z. Wang, K. Han, *et al.*, "Personalized Car Following for Autonomous Driving with Inverse Reinforcement Learning," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 2891–2897. DOI: `10.1109/ICRA46639.2022.9812446`.

INNOPOLIS UNIVERSITY