Distributed Inverse Constrained Reinforcement Learning (D-ICRL) for Multi-agent Systems (MASs)

Shicheng Liu & Minghui Zhu

The Pennsylvania State University

Neural Information Processing Systems 2022

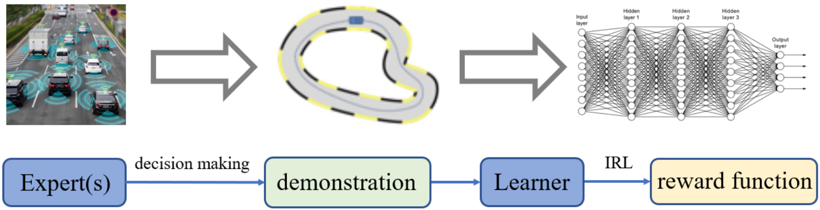
Distributed inverse constrained reinforcement learning

Reward function

Expert(s) → decision making → demonstration → Learner → IRL → reward function

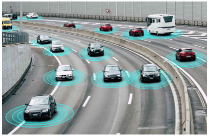
Distributed inverse constrained reinforcement learning

Reward function



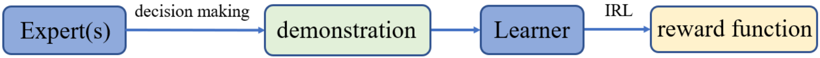
Constraints

Distributed demonstrations



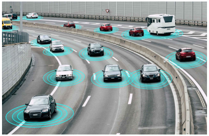
Distributed inverse constrained reinforcement learning

Reward function



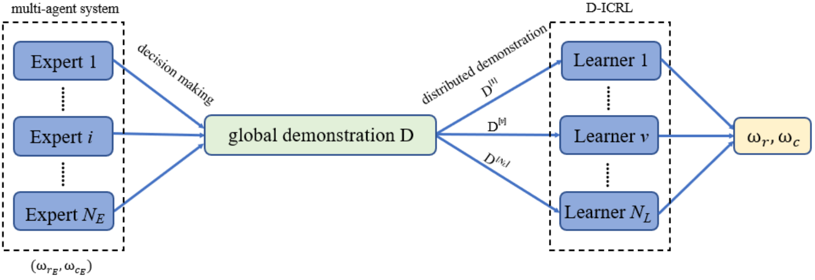
Constraints

Distributed demonstrations



- **D-ICRL**: Solve the three challenges at once.

Model: Multiple experts & multiple learners



- N_E cooperative experts: $\{r_E = \omega_{r_E}^\top \phi_r, c_E = \omega_{c_E}^\top \phi_c\} \Rightarrow \mathcal{D} = \{\mathcal{D}^{[v]}\}_{v=1}^{N_L}$
- N_L collaborative learners: $\{\mathcal{D}^{[v]} = \{\zeta^j\}_{j=1}^{m^{[v]}}, \phi_r, \phi_c\} \Rightarrow \{\omega_r, \omega_c\}$

Distributed bi-level optimization formulation

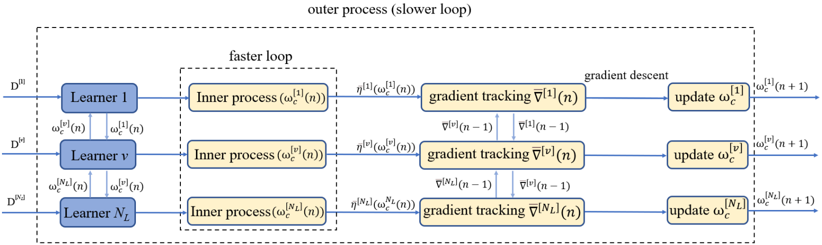
Distributed bi-level optimization formulation

$$\max_{\omega_c \in \Omega_c} \quad F(\omega_c, \eta^*(\omega_c)) = \sum_{v=1}^{N_L} F^{[v]}(\omega_c, \eta^*(\omega_c)), \qquad \text{(outer level)}$$

$$\text{s.t.} \quad \eta^*(\omega_c) = \arg\min_{\eta} \sum_{v=1}^{N_L} m^{[v]} G^{[v]}(\eta; \omega_c). \qquad \text{(inner level)}$$

- The outer level learns constraints by maximizing the log likelihood $\sum_{v=1}^{N_L} F^{[v]}$ of the demonstrations.
- Given a constraint estimate ω_c, the inner level learns the corresponding reward function and policy by minimizing the dual function $\sum_{v=1}^{N_L} m^{[v]} G^{[v]}$ of maximum causal entropy (MCE).

A perspective of double-loop learning



- Double-loop communication: sharing reward and cost function parameters
- Inner communication (faster): $W^{[vv']}(k)$ and $\mathcal{N}^{[v]}(k)$.
- Outer communication (slower): $\bar{W}^{[vv']}(n)$ and $\bar{\mathcal{N}}^{[v]}(n)$.

Inner process

- Receives $\eta^{[v']}(k)$ from neighbor $v' \in \mathcal{N}^{[v]}(k)$.
- $\eta^{[v]}(k+1) = \sum_{v'=1}^{N_L} W^{[vv']}(k)\eta^{[v']}(k) - \alpha(k)m^{[v]}\nabla_\eta G^{[v]}(\eta^{[v]}(k); \omega_c)$
- Runs K iterations

Outer process

- Difficulties
  - Local gradient $\nabla F^{[v]}(\omega_c, \eta^*(\omega_c))$ inaccessible.
  - global gradient $\nabla F(\omega_c, \eta^*(\omega_c))$ inaccessible.
  - $F(\omega_c, \eta^*(\omega_c))$ non-convex.

Outer process

- Difficulties
  - Local gradient $\nabla F^{[v]}(\omega_c, \eta^*(\omega_c))$ inaccessible.
  - global gradient $\nabla F(\omega_c, \eta^*(\omega_c))$ inaccessible.
  - $F(\omega_c, \eta^*(\omega_c))$ non-convex.
- Our solutions
  - Local gradient approximation $\bar{\nabla} F^{[v]}(\omega_c, \bar{\eta}^{[v]}(\omega_c))$.
  - Global gradient tracking $\bar{\nabla}^{[v]}(n) = \sum_{v=1}^{N_L} \bar{W}^{[vv']}(n)\bar{\nabla}^{[v']}(n-1)$ $+\bar{\nabla} F^{[v]}(\omega_c^{[v]}(n), \bar{\eta}^{[v]}(\omega_c^{[v]}(n))) - \bar{\nabla} F^{[v]}(\omega_c^{[v]}(n-1), \bar{\eta}^{[v]}(\omega_c^{[v]}(n-1)))$.
  - Successive convex approximation $\tilde{\omega}_c^{[v]}(n) = \text{Project}_{\Omega_c}(\omega_c^{[v]}(n) + N_L \bar{\nabla}^{[v]}(n))$.

Outer process

- Difficulties
  - Local gradient $\nabla F^{[v]}(\omega_c, \eta^*(\omega_c))$ inaccessible.
  - global gradient $\nabla F(\omega_c, \eta^*(\omega_c))$ inaccessible.
  - $F(\omega_c, \eta^*(\omega_c))$ non-convex.
- Our solutions
  - Local gradient approximation $\bar{\nabla} F^{[v]}(\omega_c, \bar{\eta}^{[v]}(\omega_c))$.
  - Global gradient tracking $\bar{\nabla}^{[v]}(n) = \sum_{v=1}^{N_L} \bar{W}^{[vv']}(n)\bar{\nabla}^{[v']}(n-1)$ $+\bar{\nabla} F^{[v]}(\omega_c^{[v]}(n), \bar{\eta}^{[v]}(\omega_c^{[v]}(n))) - \bar{\nabla} F^{[v]}(\omega_c^{[v]}(n-1), \bar{\eta}^{[v]}(\omega_c^{[v]}(n-1)))$.
  - Successive convex approximation $\tilde{\omega}_c^{[v]}(n) = \text{Project}_{\Omega_c}(\omega_c^{[v]}(n) + N_L \bar{\nabla}^{[v]}(n))$.
- Update rule: $\omega_c^{[v]}(n+1) = \sum_{v'=1}^{N_L} \left[\beta(n)\tilde{\omega}_c^{[v']}(n) + (1-\beta(n))\omega_c^{[v']}(n)\right]$

Theoretical guarantee

Convergence rate of inner problem

Suppose $\alpha(k) = \frac{\alpha}{k+1}$ where α is a positive constant, it holds for any learner v and $\omega_c \in \Omega_c$ that

$$||\bar{\eta}^{[v]}(\omega_c) - \eta^*(\omega_c)|| \leq O(\frac{1}{\sqrt{\log K}})$$

Asymptotic convergence of outer problem

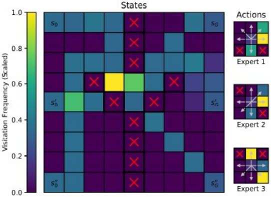
Suppose $\beta(n) \in (0, 1)$, $\sum_{n=0}^{\infty} \beta(n) = +\infty$, and $\sum_{n=0}^{\infty} \beta(n)^2 < +\infty$, it holds for any learner v that

$$\lim_{n\to\infty} \max_{v,v'} ||\omega_c^{[v]}(n) - \omega_c^{[v']}(n)|| = 0,$$

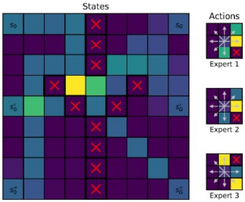
$$\limsup_{n\to\infty} (\nabla F(\omega_c^{[v]}(n), \eta^*(\omega_c^{[v]}(n))))^\top (\omega_c - \omega_c^{[v]}(n)) \leq O(\frac{1}{\sqrt{\log K}})$$

Simulations

Discrete environment

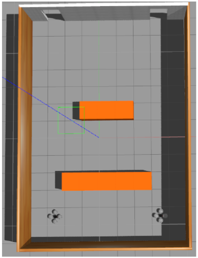


Ground truth environment

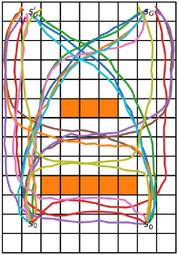


Learned environment

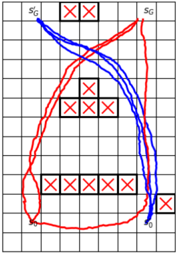
Continuous environment



Simulator environment



Demonstrated trajectories



Learned trajectories

D-ICRL can successfully imitate the experts and recover the constraints.

Conclusion

- Solve three challenges at once: Reward function, constraints, and distributed data.
- Formulate as a distributed bi-level optimization problem.
- D-ICRL: Theoretical framework effective to continuous and discrete environments empirically.