Learning to behave via Imitation

ESSAI 2024 Course

Lecture 5/5

George Vouros

University of Piraeus, Greece

July 26, 2024

Outline

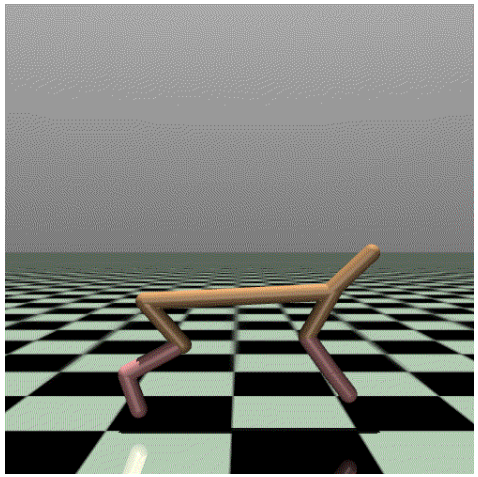
- Day 1: Motivation & Introduction to Deep Reinforcement Learning
- Day 2: Inverse Reinforcement Learning and Connections to Probabilistic Inference
- Day 3: Imitation Learning
- Day 4: Non-Markovian, Multimodal Imitation Learning
- **Day 5: Imitating in Constrained Settings** , Multiagent Imitation Learning

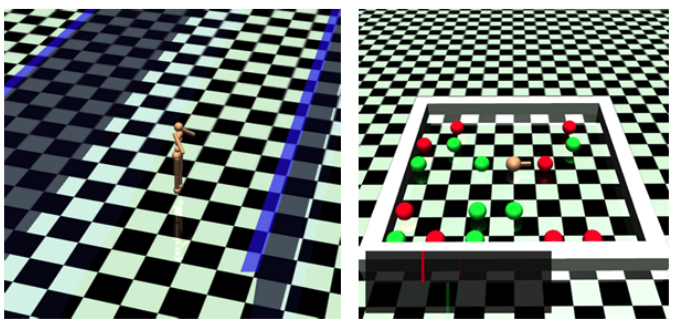
Imitation Learning

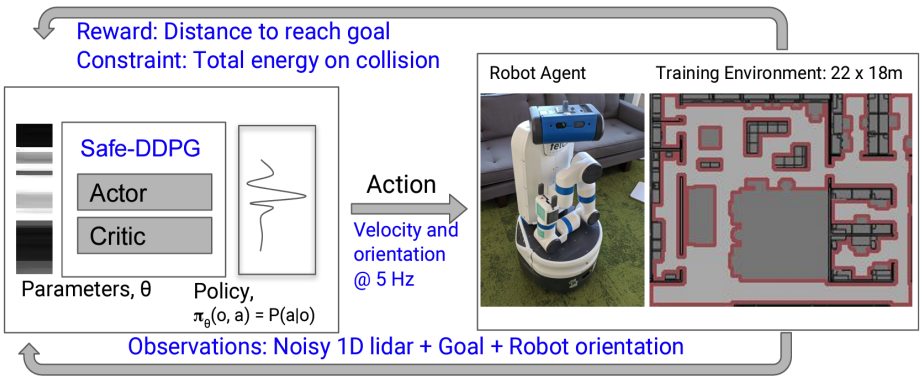
Problem (ambiguous) statement

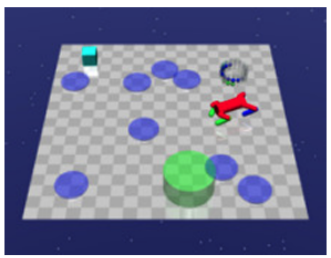
Given a set of demonstrated trajectories D generated by an unknown expert policy π_ϵ, learn a policy π that generates trajectories that are "as close as possible" to the expert trajectories.

Imitation learning in constrained settings









Up right: From W.Chow et al 2019 (not imitation learning paper but a constrained RL one!)

See also G.Liu et al.,2023, "Benchmarking Constraint inference in Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

- $\mathcal{S}$: State space
- $\mathcal{A}$: Action space
- $p(s_{t+1}|s_t, a_t)$: Transition distributions
- $r(s_t, a_t)$: Reward function
- $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+^k$: k-dimensional features **accumulated** at every time step t in a trajectory, i.e. $f(a_t, s_t)$.
- $\gamma \in [0, 1)$ the discount factor

An augmented indicator feature indicating the presence of a feature, also adding binary features (e.g. for states and actions) $\hat{f}^{\mathbb{I}} : \mathcal{S} \times \mathcal{A} \to \{0, 1\}^d, d = k + |\mathcal{S}| + |\mathcal{A}|$

Constrained Markov Decision Processes (CMDP)

The augmented indicator feature indicates the presence of a feature, adding binary features (e.g. for states and actions)

$\hat{f}^{\mathbb{I}} : \mathcal{S} \times \mathcal{A} \to \{0,1\}^d, d = k + |\mathcal{S}| + |\mathcal{A}|$

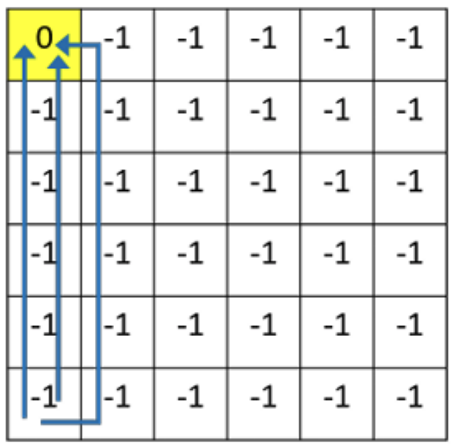
$$\hat{f}^{\mathbb{I}}_{f_i}(s,a) = \begin{cases} 1 & \text{if } f_i(s,a) > 0 \\ 0 & \text{otherwise} \end{cases}, \quad \hat{f}^{\mathbb{I}}_{f_i}(s,a) = \begin{cases} 1 & \text{if } s = s_i i \\ 0 & \text{otherwise} \end{cases}, \quad \hat{f}^{\mathbb{I}}_{f_i}(s,a) = \begin{cases} 1 & \text{if } a = a_i \\ 0 & \text{otherwise} \end{cases}$$

with sets of constraints given by

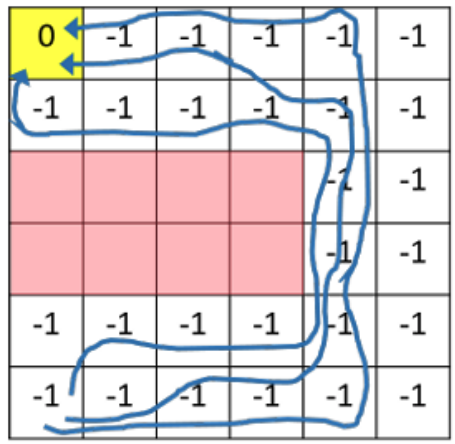
$$C_i = \{(s,a) | \hat{f}^{\mathbb{I}}_i = 1\}$$

Compound constraints: $C = \cup_i C_i$

Imitation learning in constrained settings



Nominal MDP



Constrained MDP

Images from Scobee & Sastry 2020.

Imitation Learning

Inverse Constrained Reinforcement Learning

The IRL Goal:
Recover the constraints $\hat{C}$ which are most likely to have been added to the nominal MDP given a set of demonstrations $\mathcal{D}_E$ from the agent navigating the constrained MDP.
Thus, we need to find the constraints $\hat{C}$ that maximize $P(\hat{C}|\mathcal{D}_E)$.
Assuming a uniform prior over possible constraints, from Bayes' Rule it holds that

$$P(\hat{C}|\mathcal{D}_E) \propto P(\mathcal{D}_E|\hat{C})$$

Therefore, we can solve an equivalent problem of finding which constraints maximize the likelihood of the given demonstrations.
So, let's do that: *solving the maximum likelihood constraint inference problem via solving demonstration likelihood maximization.*

Imitation learning in constrained settings

Constrained Markov Decision Processes (CMDP)

- $\mathcal{S}$: State space
- $\mathcal{A}$: Action space
- $p(s_{t+1}|s_t, a_t)$: Transition distributions
- $r(s_t, a_t)$: Reward function
- $\mathcal{C} = \{(C_i, c_i, \epsilon_i), i = 1 \dots N\}$, where C_i the (soft) constraint with associated cost function $c_i \in [0, \infty]$ and ϵ_i the associated cost bound.
- $\gamma \in [0, 1)$ the discount factor

Imitation learning in constrained settings

Constrained Markov Decision Processes (CMDP)

It is easy to specify hard constraints with a cost function

$$c_i(s, a) = \hat{f}^{\mathbb{I}}, \text{ and } \epsilon_i = 0, i = 1 \dots N$$

Imitation Learning

Constrained Markov Decision Processes (CMDP)

The RL Goal:
Find a policy π_θ that maximizes the expected discounted rewards under the set of cumulative soft constraints, with cost functions $c_i : \mathcal{S} \times \mathcal{A} \to [0, \infty]$.

$$argmax_{\pi_\theta} \mathbb{E}_{p,\pi_\theta}[\sum_{t=0}^{T} \gamma^t r_t] + \lambda \mathcal{H}(\pi)$$

s.t.

$$\mathbb{E}_{p,\pi_\theta}[\sum_{t=0}^{T} \gamma^t c_i(s_t, a_t)] \leq \epsilon_i, \forall i \in [1, N]$$

Also for $T \to \infty$, with bounds on the expectation of cumulative constraint values.

Imitation Learning

Constrained Markov Decision Processes (CMDP)

The RL Goal:
Find a policy π_θ that maximizes the expected discounted rewards under the set of cumulative soft constraints, with cost functions $c_i : \mathcal{T} \to [0, \infty]$.

$$argmax_{\pi_\theta} \mathbb{E}_{p,\pi_\theta}[\sum_{t=0}^{T} \gamma^t r_t] + \lambda \mathcal{H}(\pi)$$

s.t.

$$\mathbb{E}_{\tau \sim (p,\pi_\theta)}[c_i(\tau)] \leq \epsilon_i, \forall i \in [1, N]$$

Constrained Markov Decision Processes (CMDP)

Notice that,
considering costs on trajectories may be more restrictive than the cumulative costs.
Consider that $c(\tau) = 1 - \prod_{(s,a)\in\tau} \phi(s,a)$, where $\phi(s,a)$ **is the probability that performing** a **in** s **is safe (as demonstrated)**, and for a (s,a) with $\phi(s,a) \to 0$, then $\prod_{(s,a)\in\tau} \phi(s,a) \to 0$ and $c(\tau) \to 1$

Imitation Learning

Inverse Constrained Reinforcement Learning

Back to the IRL goal: Recover the minimum constraint set that best explains the expert data,
assuming that rewards are observable.

Therefore:

Assuming $r_\psi = r$ and **given the probability ϕ of acting safely given approximations $\hat{c}$ of cost functions over trajectories τ**, according to demonstrations $\mathcal{D}_E$.
Assuming independence of demonstrations sampled from the MaxEnt distribution, the likelihood function is
$p(\mathcal{D}_E|\phi) = \prod_{i=1}^{|\mathcal{D}_E|} p(\tau_i|\phi, r) = \prod_{i=1}^{|\mathcal{D}_E|} \frac{exp(r(\tau_i)\mathbb{I}^\phi(\tau)}{Z_\phi} =$
$\frac{1}{Z_\phi^{|\mathcal{D}_E|}} \prod_{i=1}^{|\mathcal{D}_E|} exp[r(\tau^i)]\mathbb{I}^\phi(\tau^i)$, where
(a) $Z_\phi = \int \mathbb{I}^\phi(\tau) exp(r(\tau))\, d\tau$, and
(b) $\mathbb{I}^\phi(\tau^i) = \prod_{t=1}^{T} \phi_t(s_t^i, a_t^i)$

Imitation Learning

Constrained Markov Decision Processes (CMDP)

The IRL goal: Recover the minimum constraint set that best explains the expert data, assuming that rewards are observable.

Therefore we need to maximize the demonstration probability:

$$p(\mathcal{D}_E|\phi) = \frac{1}{Z_\phi^{|\mathcal{D}_E|}} \prod_{i=1}^{|\mathcal{D}_E|} exp[r(\tau^i)]\mathbb{I}^\phi(\tau^i)$$

recovering the set of constraints $C^* = argmax_{C_\phi \in \mathcal{C}} p(\mathcal{D}_E|\phi)$

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Constrained Markov Decision Processes (CMDP)

Goal: Maximize the demonstration probability:

$$p(\mathcal{D}_E|\phi) = \frac{1}{Z_\phi^{|\mathcal{D}_E|}} \prod_{i=1}^{|\mathcal{D}_E|} exp[r(\tau^i)]\mathbb{I}^\phi(\tau^i)$$

Let

$$\mathcal{T}_\phi = \{\tau \in \mathcal{T} | \mathbb{I}^\phi(\tau) = 0\}$$

be the set of trajectories that are **made infeasible** by adding the set of constraints in the MDP.

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

The value of

$$Z_\phi = \int \mathbb{I}^\phi(\tau) exp(r(\tau))\, d\tau, \text{ with } \mathbb{I}^\phi(\tau^i) = \prod_{t=1}^{T} \phi_t(s_t^i, a_t^i)$$

is minimized when the sum of exponential rewards of infeasible trajectories is maximized. I.e., when their propability is maximized **in the unconstrained** MDP

$$\sum_{\tau \in \mathcal{T}_\phi} exp(r(\tau)) \propto \sum_{\tau \in \mathcal{T}_\phi} p(\tau|\mathcal{O}_{1:T}) = p(\mathcal{T}_\phi|\mathcal{O}_{1:T})$$

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Maximize the demonstration probability:

$$p(\mathcal{D}_E|\phi) = \frac{1}{Z_\phi^{|\mathcal{D}_E|}} \prod_{i=1}^{|\mathcal{D}_E|} exp[r(\tau^i)]\mathbb{I}^\phi(\tau^i)$$

by optimizing

$$C^* = argmax_{C \in \mathcal{C}} p(\mathcal{T}_\phi|\mathcal{O}_{1:T})$$

s.t. $\mathcal{D}_E \cap \mathcal{T}_\phi = \varnothing$

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Maximize the demonstration probability.
This entails

- Reason about the probability distribution of trajectories on the unconstrained MDP
- Find the constraints C such that $\mathcal{T}_\phi$ contains the most probability mass, while it does not contain any demonstrated trajectories.

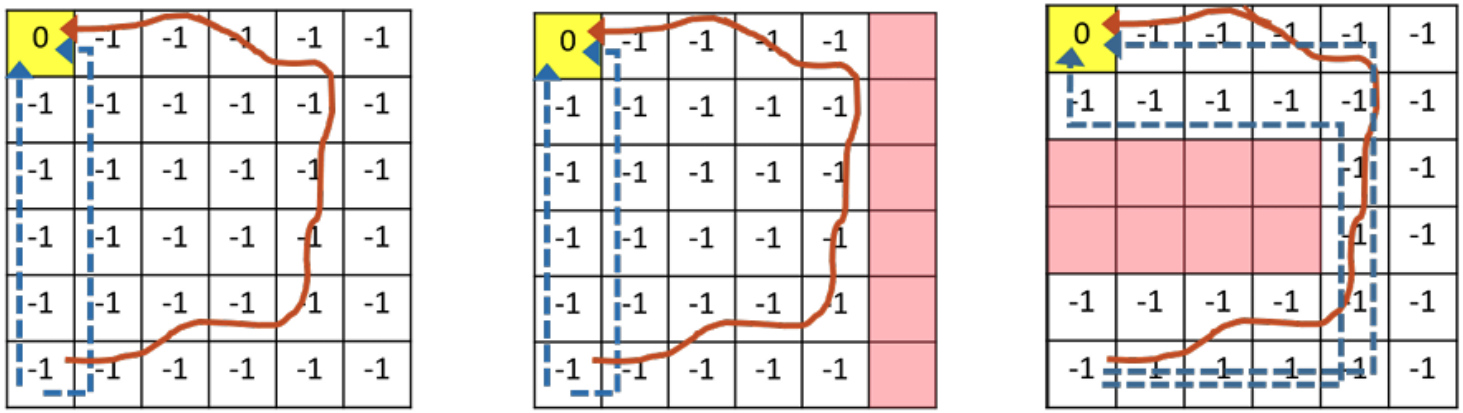
Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Maximize the demonstration probability.

Example:



(a) Nominal MDP (b) Add constraint C_1 (c) Add constraint C_2

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Recover the minimum constraint set that best explains the expert data, by maximizing the demonstration probability assuming that rewards are observable.

To infer the minimal constraints we need to penalize "heavy" ones.

However, combinations of potential minimal constraints makes the solution of this problem intractable.

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Recover the minimum constraint set that best explains the expert data, assuming that rewards are observable by maximizing the demonstration probability.

What about iteratively selecting individual (minimal) constraints

in a greedy manner?

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Recover the minimum constraint set that best explains the expert data, assuming that rewards are observable by maximizing the demonstration probability.

The "Feature Accrual History Calculation" algorithm

takes as input **the whole MDP**, a time horizon and a time varying policy that captures the expected behaviour of the demonstrator in the **nominal** MDP.

The "Feature Accrual History Calculation" algorithm is based on the Ziebart et al 2008 foward-backward algorithm for calculating expected feature counts by following a policy in the MaxEnt setting.

Details in:

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Recover the minimum constraint set that best explains the expert data, assuming that rewards are observable by maximizing the demonstration probability.

The "Feature Accrual History Calculation" algorithm

calculates the expected proportion of trajectories to have accrued any feature (corresponding to a minimal constraint) by time t: This is equal to $p(\mathcal{T}_\phi|\mathcal{O}_{1:T})$ and thus it allow as to directly select the most likely constraint according to the objective (slide 18).

Details in:

Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Recover the minimum constraint set that best explains the expert data, assuming that rewards are observable by maximizing the demonstration probability.

Remaining problem:

How to uncover the combination of constraints that covers the most probability mass.

Greedy approach: Incorporating in each iteration the constraint that covers the most currently uncovered probability mass.

Details in: Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Recover the minimum constraint set that best explains the expert data, assuming that rewards are observable by maximizing the demonstration probability.

Greedy Iterative Constraint Inference

Let $\hat{\phi}$ be the current estimated constraint set incorporated into the MDP ($M^{\hat{\phi}}$), the constraint hypothesis space $\mathcal{C}$ and the demonstrations $\mathcal{D}_E$.

1. $\hat{\phi} \leftarrow \varnothing$
2. Select the additional minimal constraint ϕ_i to fulfill the objective for the $M^{\hat{\phi}}$
3. Infer the effect of $\hat{\phi}' = \hat{\phi} \cup \phi_i$ to approximating the $\mathcal{P}_{\mathcal{D}_E}$
4. if the approximation is close enough, return $\hat{\phi}'$
5. Else go to 2

Details in: Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference

Imitation Learning

Constrained Markov Decision Processes (CMDP)

Goal: Recover the minimum constraint set that best explains the expert data, assuming that rewards are observable by maximizing the demonstration probability.

Greedy Iterative Constraint Inference

- In each iteration the algorithm selects the constraint set in the hypothesis space that covers the most currently uncovered probability mass (slide 18)

- The stopping criterion depends on the D_{KL} between the emprirical distribution over trajectories in $\mathcal{D}_E$ and the distribution over trajectories induced by $\hat{\phi}$.

Details in: Scobee & Sastry 2020, "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning"

Constrained Markov Decision Processes (CMDP)

The RL Goal: Given an CMDP, find a policy π_θ that maximizes the expected discounted rewards under the set of cumulative soft constraints, with cost functions $c_i \in [0, \infty]$.

Imitation Learning

Constrained Markov Decision Processes (CMDP)

The ICRL Goal: Given an MDP and set of demonstrations that follow a set of constraints, recover the CMDP with the minimum constraint set that best explains the expert data,
assuming that rewards are observable.

i.e.

$$\mathcal{C}^* \leftarrow argmax_{\mathcal{C}} p(\mathcal{D}|\phi)$$

, where p denote the probabilities considering the MDP and assuming a uniform prior to constraint sets.

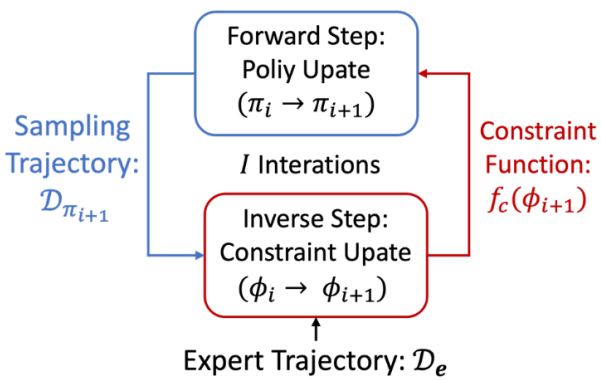


Image from G.Liu et al.,2023, "Benchmarking Constraint inference in Inverse

Imitation Learning

Inverse Constrained Reinforcement Learning[1]

$$p(\mathcal{D}_E|\phi) = \frac{1}{Z_\phi^{|\mathcal{D}_E|}} \prod_{i=1}^{|\mathcal{D}_E|} exp[r(\tau^i)]\mathbb{I}^\phi(\tau^i)$$

with (a) $Z_\phi = \int \mathbb{I}^\phi(\tau) exp(r(\tau))\, d\tau$, and (b) $\mathbb{I}^\phi(\tau^i) = \prod_{t=1}^{T} \phi_t(s_t^i, a_t^i)$

Therefore, assuming a binary classifier ϕ_ω, i.e. with parameters ω, the objective is as follows:

$$\mathcal{L}_\omega =$$

$$max_\omega \frac{1}{|\mathcal{D}_E|} \sum_{i-1}^{|\mathcal{D}_E|} [r(\tau^i) + log \prod_{t=0}^{T} \phi_\omega(s^i, a^i)] - log \int exp[r(\tau)] \prod_{t=0}^{T} \phi_\omega(s_t, a_t)\, d\tau$$

[1] S.Malik et al, Inverse Constrained Reinforcement Learning, PMLR, 2021

Imitation Learning

Inverse Constrained Reinforcement Learning[2]

Therefore,

$$\nabla_\omega \mathcal{L}_\omega =$$

$$\frac{1}{|\mathcal{D}_E|}\sum_{i-1}^{|\mathcal{D}_E|}[\sum_{t=0}^{T}\nabla_\omega log\phi_\omega(s^i,a^i)] - \mathbb{E}_{\tau\sim\pi_\phi}[\sum_{t=0}^{T}\nabla_\omega log\phi_\omega(s_t,a_t)] =$$

$$\mathbb{E}_{\tau^i\sim\pi_E}[\sum_{t=0}^{T}\nabla_\omega log\phi_\omega(s^i,a^i)] - \mathbb{E}_{\tau\sim\pi_\phi}[\sum_{t=0}^{T}\nabla_\omega log\phi_\omega(s_t,a_t)]$$

where, the maximum entropy policy, i.e. the one maximizing the rewards subject to satisfying the constraints, i.e. $\sum_t \hat{c}(s_t,a_t) < \epsilon$, where $\hat{c}(s_t,a_t) = 1 - \phi_\omega(s_t,a_t)$ is as follows:

$$\pi_\phi(\tau) = \frac{exp(r(\tau))\phi_\omega(\tau)}{\int exp(r(\bar{\tau}))\phi_\omega(\bar{\tau})\, d\bar{\tau}}$$

[2] S.Malik et al, Inverse Constrained Reinforcement Learning, PMLR, 2021

Imitation Learning

Inverse Constrained Reinforcement Learning[3]

Aiming to the least constraining constraints and avoid overfit to a small number of samples, (Malik et al 2021) incorporate a regularizer

$$\nabla_\omega \mathcal{L}_\omega = \mathbb{E}_{\tau^i \sim \pi_E}[\sum_{t=0}^{T} \nabla_\omega log \phi_\omega(s^i, a^i)] - \mathbb{E}_{\tau \sim \pi_\phi}[\sum_{t=0}^{T} \nabla_\omega log \phi_\omega(s_t, a_t)] - R(\omega)$$

where

$$R(\omega) = -\delta \sum_{\tau \sim \{\mathcal{D}_E, \pi_\phi\}} |1 - \phi_\omega(\tau)|$$

where $\delta \in [0, 1)$ a constant.

[3] S.Malik et al, Inverse Constrained Reinforcement Learning, PMLR, 2021

Imitation Learning

Inverse Constrained Reinforcement Learning[4]

$$\nabla_\omega \mathcal{L}_\phi = \mathbb{E}_{\tau^i \sim \pi_E}[\sum_{t=0}^{T} \nabla_\omega log \phi_\omega(s^i, a^i)] - \mathbb{E}_{\tau \sim \pi_\phi}[\sum_{t=0}^{T} \nabla_\omega log \phi_\omega(s_t, a_t)] - R(\omega)$$

where

$$R(\omega) = -\delta \sum_{\tau \sim \{\mathcal{D}_E, \pi_\phi\}} |1 - \phi_\omega(\tau)|$$

and $\delta \in [0, 1)$ a constant.
ICRL implements ϕ_ω as a binary classifier.

[4] S.Malik et al, Inverse Constrained Reinforcement Learning, PMLR, 2021

Imitation Learning

Inverse Constrained Reinforcement Learning[5]

The ICRL algorithm (Malik et al 2021) makes two more "tricks"

- incorporating importance sampling weights
  $w_t = w(s_t, a_t) = \frac{\phi'_\omega(s_t,a_t)}{\phi_\omega(s_t,a_t)}$.
  So the objective becomes

$$\nabla_\omega \mathcal{L}_\phi = \mathbb{E}_{\tau^i \sim \pi_E}[\sum_{t=0}^{T} \nabla_\omega log \phi_\omega(s^i, a^i)] -$$

$$\mathbb{E}_{\tau \sim \pi_\phi}[\sum_{t=0}^{T} w_t \nabla_\omega log \phi_\omega(s_t, a_t)] - R(\omega)$$

- early stopping based on KL divergence i.e. $D_{KL}(\pi'_\omega || \pi_\omega)$ and $D_{KL}(\pi_\omega || \pi'_\omega)$ bounds among policy updates in terms of importance sampling weights

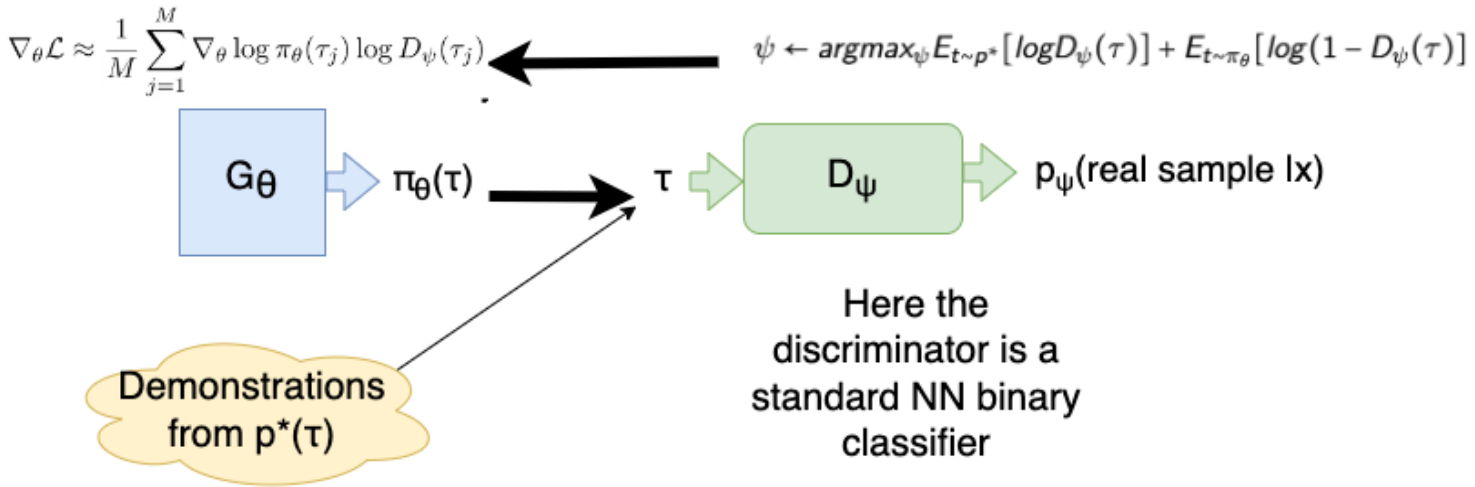
[5] S.Malik et al, Inverse Constrained Reinforcement Learning, PMLR, 2021

Imitation Learning

Multi-agent imitation learning

Imitation Learning

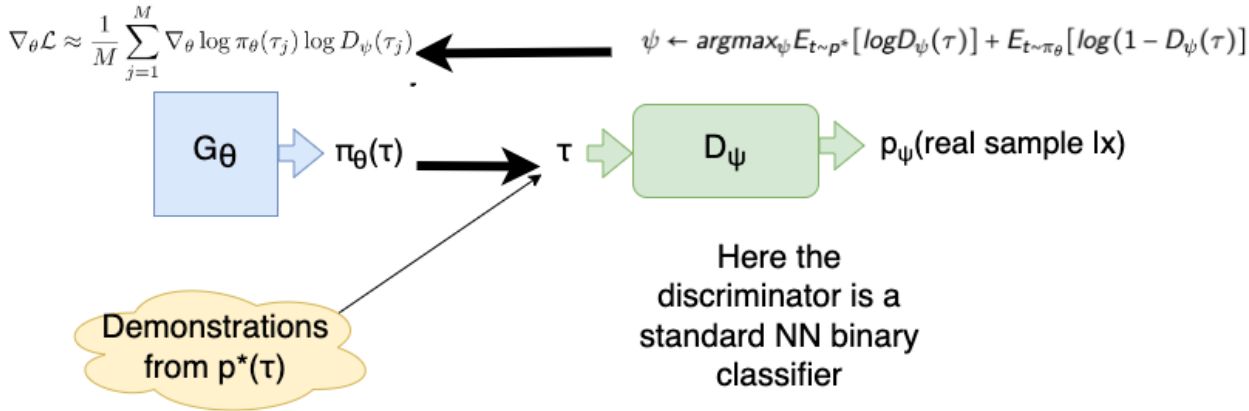
Generative Adversarial Imitation Learning



Ho and Ermon, 2016.

Imitation Learning

Multi-agent Generative Adversarial Imitation Learning

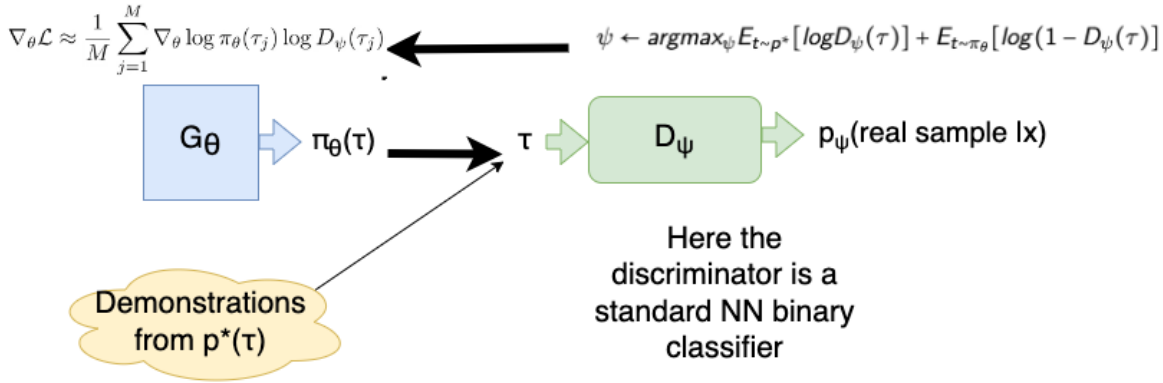


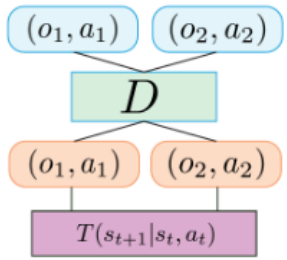
G_θ is implemented using MACK (Multi-agent Actor Critic with Kronecker-factors) following the CTDE (Centralized Training Decentralized Execution) paradigm, with an advantage function of all agents' observations and actions (no assumption of the knowledge of others' policies).

Song et al., Multi-agent Generative Adversarial Imitation Learning, 2018.

Imitation Learning

Cooperative Centralized Multi-agent Generative Adversarial Imitation Learning



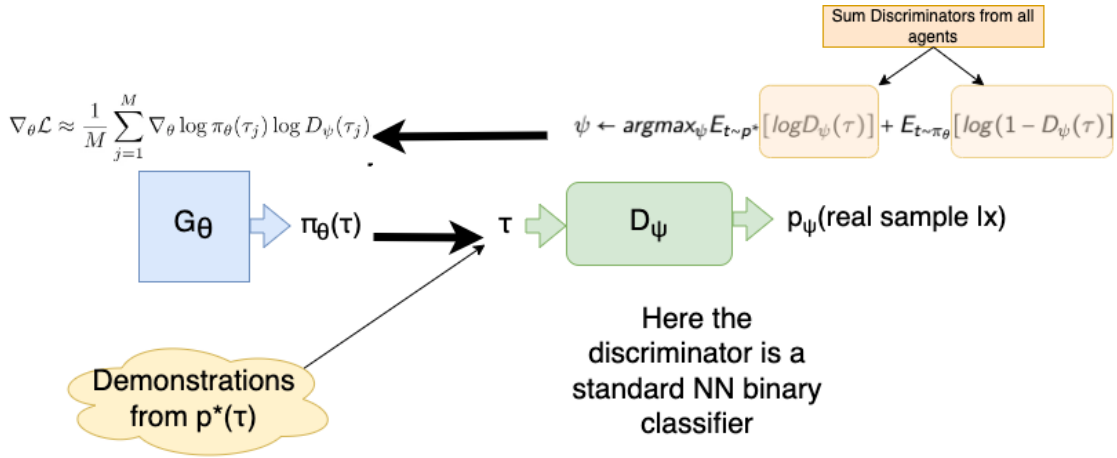


Learns the joint policy minimizing the distance between the generated state-action distribution and the experts' distribution using the J-S divergence !

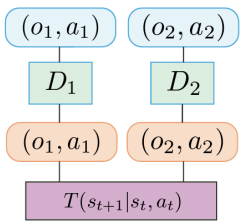
Song et al., Multi-agent Generative Adversarial Imitation Learning, 2018.

Imitation Learning

Decentralized Multi-agent Generative Adversarial Imitation Learning



Decentralized setting:

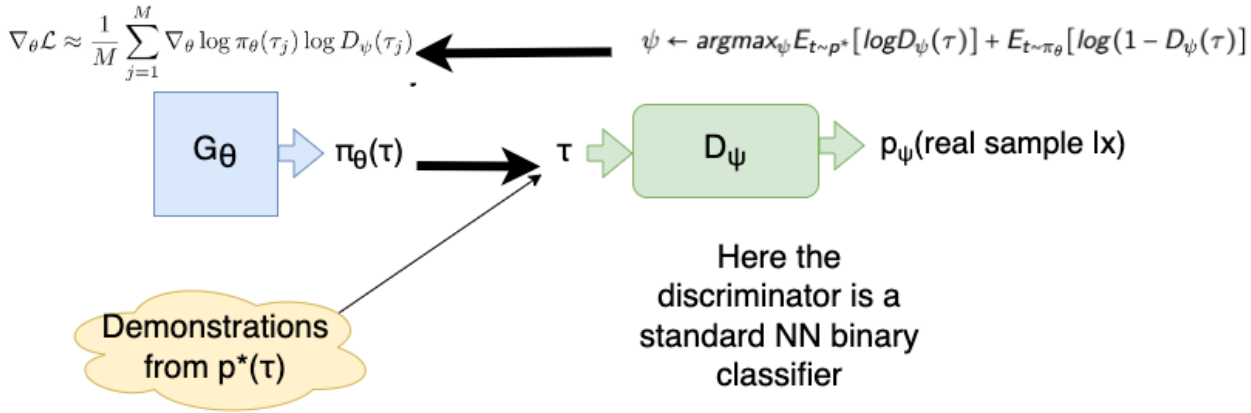


Discriminators interact indirectly via the environment !!

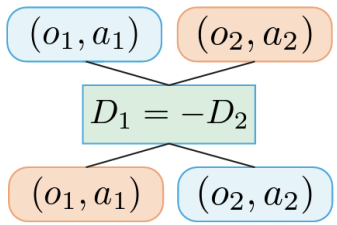
Song et al., Multi-agent Generative Adversarial Imitation Learning, 2018.

Imitation Learning

Interesting case: Competitive Multi-agent Generative Adversarial Imitation Learning



In a zero-sum setting:

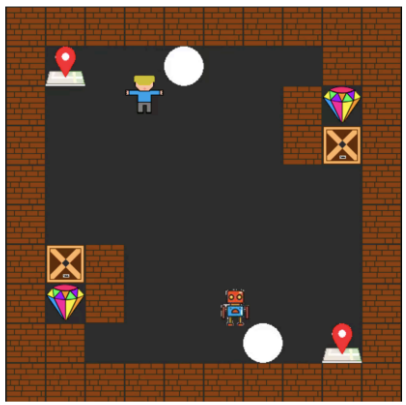
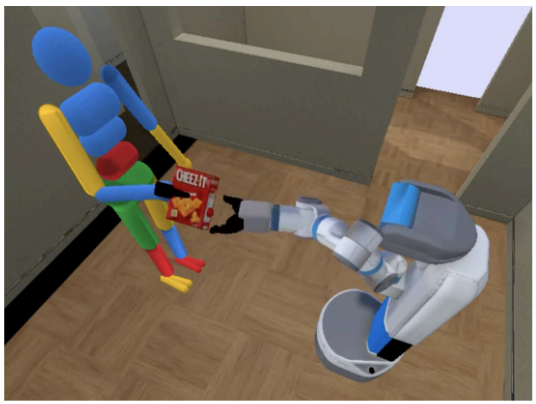


For agent 1: The discriminator tries to maximize $v(\pi_{E_1}, \pi_2) = \mathbb{E}_{\pi_{E_1}, \pi_2}(r_1(s, a))$ and minimize $v(\pi_2, \pi_{E_1})$

Song et al., Multi-agent Generative Adversarial Imitation Learning, 2018.

Imitation Learning

Collaborative GAIL for Human-Agent Collaboration


C.Wang et al., "Co-GAIL: Learning Diverse Strategies for Human-Robot Collaboration", 2023

Imitation Learning

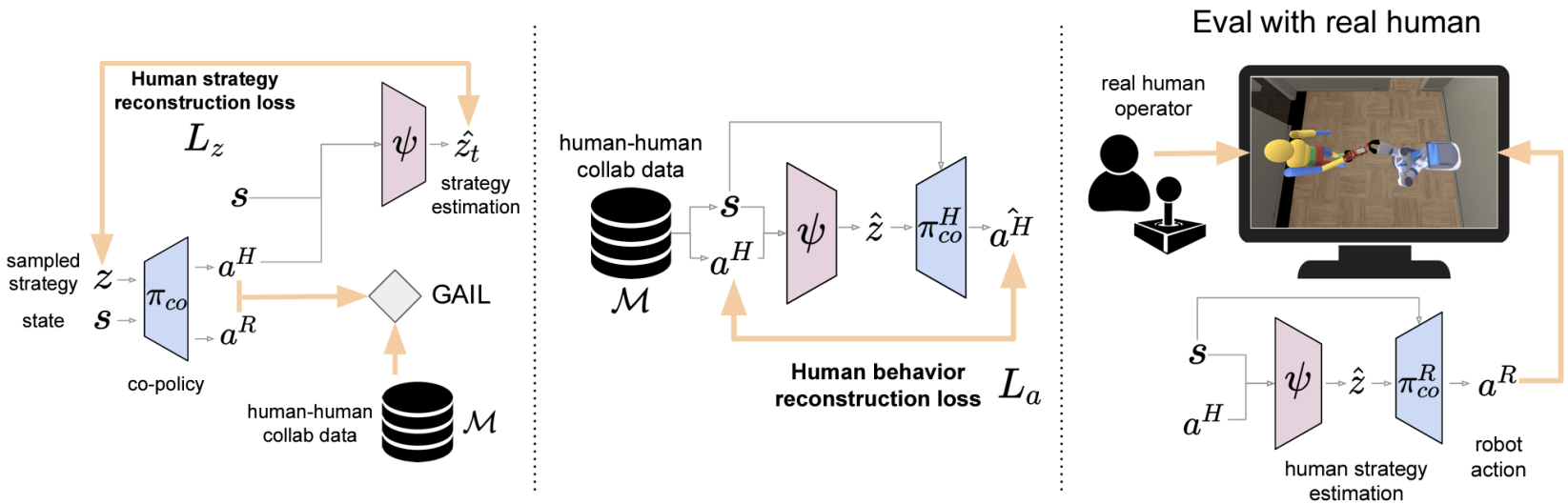
Collaborative GAIL for Human-Agent Collaboration

Goal: Learn policies that generate diverse human and robot collaboration behaviors from human-human demonstrations with an interactive co-optimization process.
How: Both polices (human and agent) co-evolve & different styles of play are represented using a latent representation.

C.Wang et al., "Co-GAIL: Learning Diverse Strategies for Human-Robot Collaboration", 2023

Imitation Learning

Collaborative GAIL for Human-Agent Collaboration

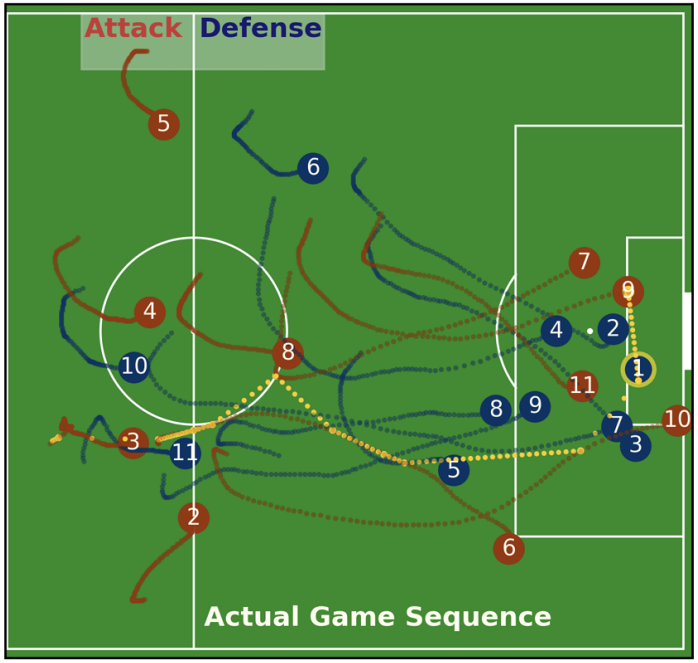


The Reward is a function of both agents' actions.
The learned co-policy maximizes the shared expected return and it is conditioned on the human styles of play (strategies) to complete the task.

C.Wang et al., "Co-GAIL: Learning Diverse Strategies for Human-Robot Collaboration", 2023

Imitation Learning

Coordinated Multi-Agent Imitation Learning

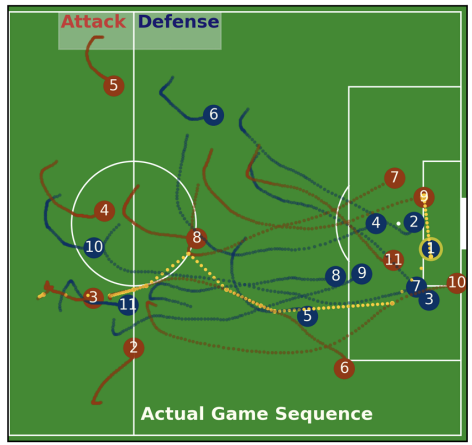


Goal: Learning a model of coordination from demonstrations.
Approach: Learn a latent coordination model (unsupervised learning) along with individual policies (conventional imitation learning)

H.M.Le et al., "Coordinated Multi-Agent Imitation Learning", 2018

Imitation Learning

Coordinated Multi-Agent Imitation Learning



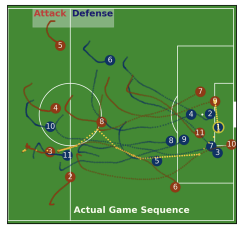
K experts (without identity) and unstructured set of synchronized demonstrations $U_k = \{u_{t,k}\}_{t=1}^{T}, k = 1 \dots K$ in associated synchronized context $C = \{c_t\}_{t=1}^{T}$.

Goal: Learn the joint policy that minimizes the imitation loss in terms of generated states' distribution.

H.M.Le et al., "Coordinated Multi-Agent Imitation Learning", 2018

Imitation Learning

Coordinated Multi-Agent Imitation Learning



Decentralized setting Goal: Learn the joint policy $\{\pi_1 \dots \pi_K\}$, where each constituent policy is tailored to a specific role, that minimizes the imitation loss in terms of generated states' distribution.

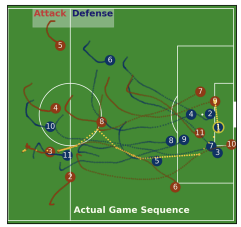
$$\mathcal{L} = \sum_{k=1}^{K} \mathbb{E}_{s \sim d_{\pi_k}} [\pi_k(s_k)]$$

Roles are undefined, unobserved, and could change dynamically within the same sequence.

H.M.Le et al., "Coordinated Multi-Agent Imitation Learning", 2018

Imitation Learning

Coordinated Multi-Agent Imitation Learning



Coordinated policy learning involves:

- Role assignment: Learn a latent variable model q for mapping the unstructured set U to a rearrangement $A(U, q)$.

$$A : \{U_1, U_2, \ldots, U_K\} \times q \rightarrow [A_1, A_2, \ldots, A_K]$$

- Learn the joint policy with the objective of maximizing the mutual information between the latent structure and demonstrations D:

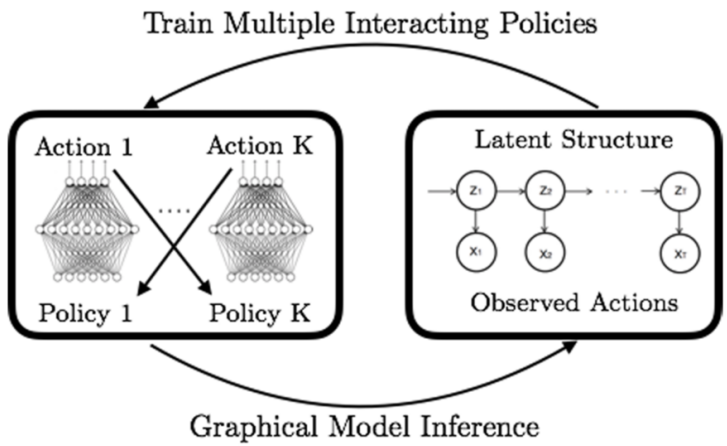
$$\mathcal{L} = min_{(\pi_i, i=1..,K),A} \sum_{k=1}^{K} \mathbb{E}_{s_k \sim d_{\pi_k}} [\pi_k(s_k)|A, D] - \lambda H(A|D)$$

Imitation Learning

Coordinated Multi-Agent Imitation Learning

Coordinated policy learning involves:

- Role assignment: A:$\{U_1, U_2, \ldots, U_K\} \times q \rightarrow [A_1, A_2, \ldots, A_K]$
- Learn the joint policy:
  $\mathcal{L} = min_{(\pi_i, i=1..,K),A} \sum_{k=1}^{K} \mathbb{E}_{s_k \sim d_{\pi_k}} [\pi_k(s_k)|A, D] - \lambda H(A|D)$



H.M.Le et al., "Coordinated Multi-Agent Imitation Learning", 2018

Imitation Learning

Coordinated Multi-Agent Imitation Learning

Coordinated policy learning involves:

- Role assignment: A:$\{U_1, U_2, \ldots, U_K\} \times q \rightarrow [A_1, A_2, \ldots, A_K]$
- Learn the joint policy:
  $\mathcal{L} = min_{(\pi_i, i=1..,K),A} \sum_{k=1}^{K} \mathbb{E}_{s_k \sim d_{\pi_k}} [\pi_k(s_k)|A, D] - \lambda H(A|D)$

The Algorithm:
Given a set of unstructured demonstrated trajectories U, and an initialized joint policy

1. Role assignment: This results into an ordered set of trajectories A_i corresponding to policy $\pi_i, i = 1, .., K$.
2. Update the joint policy: Each policy π_k is updated using the ordered set of trajectories.
3. Roll-out the joint policy to get a new set of unstructured trajectories.
4. Use the new set to update the parameters of the latent variables model.
5. Go to 1.

H.M.Le et al., "Coordinated Multi-Agent Imitation Learning", 2018

Imitation Learning

Coordinated Multi-Agent Imitation Learning

Update the joint policy: Each policy π_k is updated using the ordered set of trajectories, following a reduction-based approach (e.g. DAgger)
For increasing prediction horizons until reaching T:

1. Iteratively perform roll-out at each time step i for all K policies to obtain actions $\{\hat{a}_{i,k}\}$.
2. Each policy simultaneously updates its state $\{\hat{s}_{i,k}\}$ using the prediction from all other policies.
3. At the end of the current segment, all policies are updated using the error signal from the deviation between predicted $\{\hat{a}_{i,k}\}$ versus expert action $\{a^*_{i,k}\}$ for all i along the sub-segment.

H.M.Le et al., "Coordinated Multi-Agent Imitation Learning", 2018

Imitation Learning

Cooperative Inverse Reinforcement Learning

Hadfield-Menell et al., "Cooperative Inverse Reinforcement Learning", 2016.