COPS RL reading grp

Topic: Frameworks in MARL

4th Meet

20th Feb

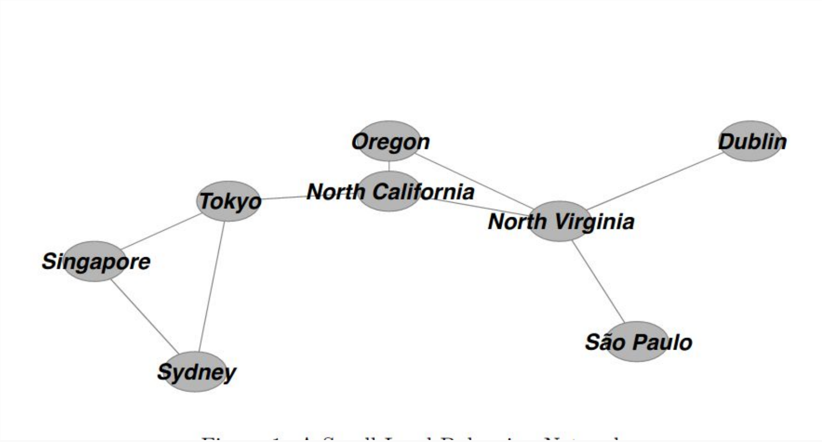
Somnath

Accelerating Multi-Agent Reinforcement Learning with Dynamic Co-Learning.

Key Problems addressed.

Consider simple MA which are formed into groups or nodes.

- What Information should be transferred between agents?
- How should we define Contextual Compatibility and use this to form sharing groups?
- How should concerns involving scalability be addressed?
- How should the architecture handle temporal heterogeneity in transition and reward models.



Forming Sharing groups for efficient experience sharing

Context Features

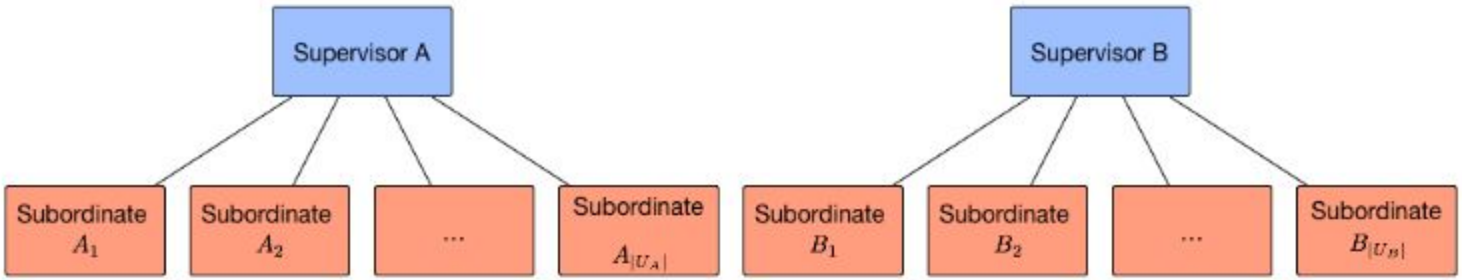


Figure 2: Example Hierarchy Involving 2 Supervisors

We Derive a set of features to help us for understanding the familiarity or the difference between two states reward model. It takes all the independent actions and conditioned states.

Context Feature Fidelity

These Context Features are then analyzed using some Distance Function (D) which measures how close the two contexts are and hence we can use it as metrics for forming Sharing groups.

Experience Sharing

Input: R_i, R_k, stochastic reward models for agents i and k, respectively.
P_i, P_k, state transition models for agents i and k, respectively.
$\lambda \in (0, 1]$, a parameter balancing the importance of accuracy in the state transition model with the reward model

Output: m, a measure of context feature fidelity

Compute similarity in reward as symmetric KL divergence for a permutation ρ of agents:

$$D_R(\rho) \leftarrow D_{SKL}\big(R_i(r_i|S_\rho, a_{\rho(1)}, a_{\rho(2)}, \ldots, a_{\rho(n)}),\\ R_k(r_k|S, a_1, a_2, \ldots, a_n)\big)$$

Compute similarity in state transition models as symmetric KL divergence:

$$D_S(\rho) \leftarrow D_{SKL}\big(P_i(s'_i|S_\rho, a_{\rho(1)}, a_{\rho(2)}, \ldots, a_{\rho(n)}),\\ P_k(s'_k|S, a_1, a_2, \ldots, a_n)\big)$$

Find the best permutation ρ according to the balancing parameter λ:

$$m \leftarrow \min_{\rho} \; \lambda\left(1 - \exp\left(-D_R(\rho)\right)\right) + \\ (1 - \lambda)\left(1 - \exp\left(-D_S(\rho)\right)\right)$$

Clustering Algorithm

Here K is a window of time between which experience is logged or shared.

Input: Vector $V = \langle V_1, V_2, \ldots, V_n \rangle$ of feature vectors for agents $A = \{1, 2, \ldots, n\}$, temperature parameter $T \in \mathbb{N}$

Output: A set-valued mapping $f : A \to \mathcal{P}(A)$ describing the (possibly empty) set of agents that the input agent should share with

Cluster V into k groups $C_1, C_2, \ldots, C_k$

for $i \leftarrow \{1, 2, \ldots, k\}$ **do**

  $M \leftarrow$ Pairwise distances D^C over vectors within C_i

  $P \leftarrow \sum_{t \in 1..T} \text{BinomialPMF}\left(\frac{\exp(M)}{\sum \exp(M)}, t\right)$

  for $a \in Agents(C_i)$ **do**

    for $b \in Agents(C_i) \setminus a$ **do**

      Let p be the entry in P corresponding to entry (a, b)

      With probability p, let $f(a) \leftarrow f(a) \cup \{b\}$

    end

  end

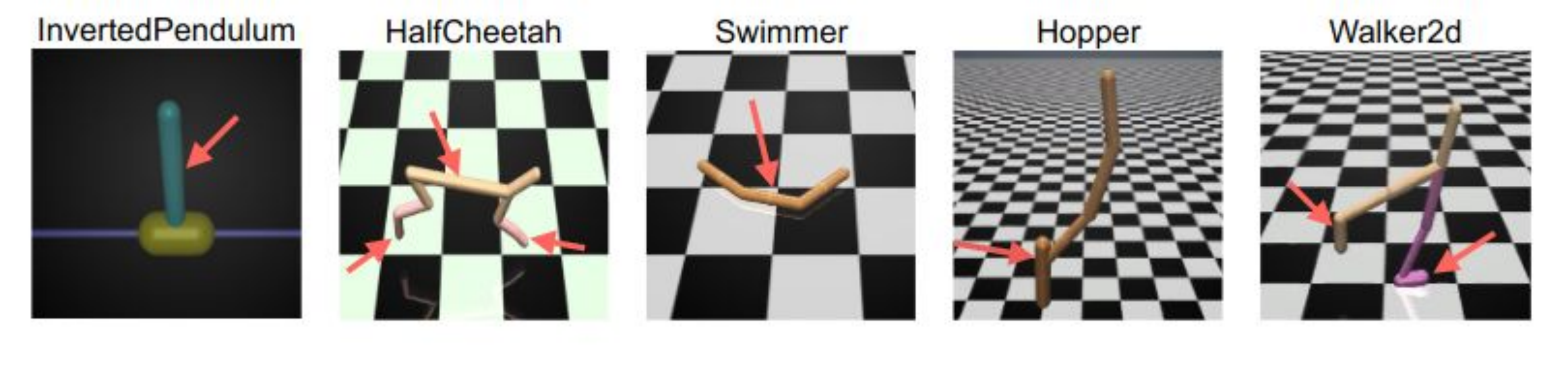
end

Algorithm 2: Selection of Sharing Partners

Ayush

Robust Adversarial Reinforcement Learning

Learning in Simulation and Challenges



RARL Algorithm

$$\rho(\mu;\theta^{\mu}) = \mathop{\mathbb{E}}_{\mathcal{P}}\left[\mathbb{E}\left[\sum_{t=0}^{T}\gamma^{t} r(s_t, a_t) | s_0, \mu, \mathcal{P}\right]\right].$$

Algorithm 1 RARL (proposed algorithm)

Input: Environment $\mathcal{E}$; Stochastic policies μ and ν

Initialize: Learnable parameters θ_0^{μ} for μ and θ_0^{ν} for ν

for i=1,2,..N_{iter} **do**

  $\theta_i^{\mu} \leftarrow \theta_{i-1}^{\mu}$

  for j=1,2,..N_{μ} **do**

    $\{(s_t^i, a_t^{1i}, a_t^{2i}, r_t^{1i}, r_t^{2i})\} \leftarrow \text{roll}(\mathcal{E}, \mu_{\theta_i^{\mu}}, \nu_{\theta_{i-1}^{\nu}}, N_{\text{traj}})$

    $\theta_i^{\mu} \leftarrow \text{policyOptimizer}(\{(s_t^i, a_t^{1i}, r_t^{1i})\}, \mu, \theta_i^{\mu})$

  end for

  $\theta_i^{\nu} \leftarrow \theta_{i-1}^{\nu}$

  for j=1,2,..N_{ν} **do**

    $\{(s_t^i, a_t^{1i}, a_t^{2i}, r_t^{1i}, r_t^{2i})\} \leftarrow \text{roll}(\mathcal{E}, \mu_{\theta_i^{\mu}}, \nu_{\theta_i^{\nu}}, N_{\text{traj}})$

    $\theta_i^{\nu} \leftarrow \text{policyOptimizer}(\{(s_t^i, a_t^{2i}, r_t^{2i})\}, \nu, \theta_i^{\nu})$

  end for

end for

Return: $\theta_{N_{\text{iter}}}^{\mu}, \theta_{N_{\text{iter}}}^{\nu}$

Performance comparison with baseline